SIAM J. COMPUT.
Vol. 16, No. 1, pp. 1–31

LINE SEARCH FILTER METHODS FOR NONLINEAR PROGRAMMING: MOTIVATION AND GLOBAL CONVERGENCE[*]

ANDREAS WÄCHTER[†] AND LORENZ T. BIEGLER[‡]

Abstract. Line search methods are proposed for nonlinear programming using Fletcher and Leyffer's filter method [*Math. Program.*, 91 (2002), pp. 239–269], which replaces the traditional merit function. Their global convergence properties are analyzed. The presented framework is applied to active set sequential quadratic programming (SQP) and barrier interior point algorithms. Under mild assumptions it is shown that every limit point of the sequence of iterates generated by the algorithm is feasible, and that there exists at least one limit point that is a stationary point for the problem under consideration. A new alternative filter approach employing the Lagrangian function instead of the objective function with identical global convergence properties is briefly discussed.

Key words. nonlinear programming, nonconvex constrained optimization, filter method, line search, sequential quadratic programming, interior point method, barrier method, global convergence

AMS subject classifications. 49M37, 65K05, 90C30, 90C51, 90C55

DOI. 10.1137/S1052623403426556

1. Introduction.

Recently, Fletcher and Leyffer [9] proposed filter methods, offering an alternative to merit functions, as a tool to guarantee global convergence in algorithms for nonlinear programming (NLP). The underlying concept is that trial points are accepted if they improve the objective function *or* improve the constraint violation instead of a combination of those two measures defined by a merit function. The practical results reported for the filter trust region sequential quadratic programming (SQP) method in [9] are encouraging, and subsequently global convergence results for related algorithms were established by Fletcher et al. [7] and Fletcher, Leyffer, and Toint [10]. Other researchers also proposed global convergence results for different trust region based filter methods, such as for an interior point approach (M. Ulbrich, S. Ulbrich, and Vicente [21]), a bundle method for nonsmooth optimization (Fletcher and Leyffer [8]), and a pattern search algorithm for derivative-free optimization (Audet and Dennis [1]).

In this paper we propose and analyze a filter method framework based on line search which can be applied to active set SQP methods as well as barrier interior point methods. The motivation given by Fletcher and Leyffer [9] for the development of the filter method is to avoid the necessity of determining a suitable value of the penalty parameter in the merit function. In addition, in the context of a line search method, the filter approach offers another important advantage regarding robustness. It has been known for some time that line search methods can converge to "spurious solutions," infeasible points that are not even critical points for a measure of infeasibility, if the gradients of the constraints become linearly dependent at nonfeasible points. In [19], Powell gave an example for this behavior. More recently, Wächter and Biegler [25] demonstrated another global convergence problem for many line search interior

[*] Received by the editors April 18, 2003; accepted for publication (in revised form) October 6, 2004; published electronically August 17, 2005.

http://www.siam.org/journals/sicomp/16-1/42655.html

[†] Department of Mathematical Sciences, IBM T.J. Watson Research Center, Yorktown Heights, NY 10598 (andreasw@watson.ibm.com).

[‡] Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 (lb01@andrew.cmdu.edu).

point methods on a simple well-posed example. Here, the affected methods generate search directions that point outside of the region $\mathcal{I}$ defined by the inequality constraints because they are forced to satisfy the linearization of the equality constraints. Consequently, an increasingly smaller fraction of the proposed step can be taken, and the iterates eventually converge to an infeasible point at the boundary of $\mathcal{I}$, which once again is not even a stationary point for any norm of the constraint violation (see also Marazzi and Nocedal [15] for a detailed discussion of "feasibility control"). Using a filter approach within a line search algorithm helps to overcome these problems. If the trial step size becomes too small to guarantee sufficient progress toward a solution of the problem, the proposed filter method reverts to a feasibility restoration phase, whose goal is to deliver a new acceptable iterate by decreasing the constraint violation, or to converge to a local minimizer of infeasibility if this is not possible. In this way, the filter line search procedure detects problematic cases automatically, so that global convergence problems described above cannot occur if a suitable algorithm for the restoration phase is used.

This paper is organized as follows. For easy comprehension of the derivation and analysis of the proposed line search filter method, the main part of the paper considers the particular case of solving nonlinear optimization problems without inequality constraints. At the end of the paper it is shown how the presented techniques can be applied to general NLPs using active set SQP methods and a barrier approach.

In section 2 we motivate and state the algorithm for the solution of the equality constrained problem. The method is motivated by the trust region SQP method proposed by Fletcher et al. [7]. An important difference, however, lies in the condition that determines when to switch between certain sufficient decrease criteria. The proposed rule is more general and allows us to show fast local convergence of the proposed line search filter method in the companion paper [26]. We then show in section 3 that every limit point of the sequence of iterates generated by the algorithm is feasible, and that there is at least one limit point that satisfies the first order optimality conditions for the problem.

In section 4.1 we propose an alternative measure for the filter acceptance criteria. Here, a trial point is accepted if it reduces the infeasibility or the value of the Lagrangian function (instead of the objective function). The global convergence results still hold for this modification. Having presented the line search filter framework on the simple case of problems with equality constraints only, we show in section 4.2 how it can be applied to SQP methods handling inequality constraints, preserving the same global convergence properties. Finally, section 4.3 shows how the presented line search filter method can be applied in a barrier interior point framework.

1.1. Notation. We denote the ith component of a vector $v \in \mathbb{R}^n$ by $v^{(i)}$, and the ith unit coordinate vector is called e_i in the text. Norms $\|\cdot\|$ denote a fixed vector norm and its compatible matrix norm unless otherwise noted. For brevity, we use the convention $(x, \lambda) = (x^T, \lambda^T)^T$ for vectors x, λ. For a matrix A, we denote by $\sigma_{\min}(A)$ the smallest singular value of A, and for a symmetric, positive definite matrix A we call the smallest eigenvalue $\lambda_{\min}(A)$. Given two vectors $v, w \in \mathbb{R}^n$, we define the convex segment $[v, w] := \{v + t(w - v) : t \in [0, 1]\}$. Finally, we denote by $O(t_k)$ a sequence $\{v_k\}$ satisfying $\|v_k\| \leq \beta\ t_k$ for some constant $\beta > 0$ independent of k, and by $o(t_k)$ a sequence $\{v_k\}$ satisfying $\|v_k\| \leq \beta_k t_k$ for some positive sequence $\{\beta_k\}$ with $\lim_k \beta_k = 0$.

2. A line search filter approach. For simplicity, we first describe and analyze the line search filter method for NLPs with equality constraints only; i.e., we assume that the problem to be solved is stated as

$$\min_{x\in\mathbb{R}^n} \quad f(x) \tag{1a}$$

$$\text{subject to} \quad c(x) = 0, \tag{1b}$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$ and the equality constraints $c : \mathbb{R}^n \to \mathbb{R}^m$ with $m < n$ are sufficiently smooth. We later show how this approach can be used in an active set SQP (section 4.2) and an interior point (section 4.3) framework in order to tackle general NLPs.

The Karush–Kuhn–Tucker (KKT) conditions for the NLP (1) are

$$g(x) + A(x)\lambda = 0, \tag{2a}$$

$$c(x) = 0, \tag{2b}$$

where we denote with $A(x) := \nabla c(x)$ the transpose of the Jacobian of the constraints c, and with $g(x) := \nabla f(x)$ the gradient of the objective function. The vector λ corresponds to the Lagrange multipliers for the equality constraints (1b). Under certain constraint qualifications, such as linear independence of the constraint gradients, the KKT conditions are the first order optimality conditions for (1) (see, e.g., [17]).

Given an initial estimate x_0, the line search algorithm proposed in this section generates a sequence of improved estimates x_k of the solution for the NLP (1). For this purpose in each iteration k a search direction d_k is computed from the linearization at x_k of the KKT conditions (2),

$$\begin{bmatrix} H_k & A_k \\ A_k^T & 0 \end{bmatrix} \begin{pmatrix} d_k \\ \lambda_k^+ \end{pmatrix} = -\begin{pmatrix} g_k \\ c_k \end{pmatrix}. \tag{3}$$

Here, $A_k := A(x_k)$, $g_k := g(x_k)$, and $c_k := c(x_k)$. The symmetric matrix H_k denotes the Hessian $\nabla^2_{xx}\mathcal{L}(x_k, \lambda_k)$ of the Lagrangian

$$\mathcal{L}(x, \lambda) := f(x) + c(x)^T\lambda \tag{4}$$

of the NLP (1), or an approximation to this Hessian. The vector λ_k is some estimate of the optimal multipliers corresponding to the equality constraints (1b), and λ_k^+ in (3) can be used to determine a new estimate λ_{k+1} for the next iteration. As is common for most line search methods, we assume that the projection of the Hessian approximation H_k onto the null space of the constraint Jacobian is uniformly positive definite.

After a search direction d_k has been computed, a step size $\alpha_k \in (0, 1]$ is determined in order to obtain the next iterate

$$x_{k+1} := x_k + \alpha_k d_k. \tag{5}$$

We want to guarantee that ideally the sequence $\{x_k\}$ of iterates converges to a solution of the NLP (1). In this paper we consider a backtracking line search procedure, where a decreasing sequence of step sizes $\alpha_{k,l} \in (0, 1]$ $(l = 0, 1, 2, \ldots)$ is tried until some acceptance criterion is satisfied. Traditionally, a trial step size $\alpha_{k,l}$ is accepted if the corresponding trial point

$$x_k(\alpha_{k,l}) := x_k + \alpha_{k,l} d_k \tag{6}$$

provides sufficient reduction of a *merit function*, such as the exact penalty function [14]

$$\phi_\rho(x) = f(x) + \rho\,\theta(x), \tag{7}$$

where we define the infeasibility measure $\theta(x)$ by

$$\theta(x) = \|c(x)\|.$$

Under certain regularity assumptions it can be shown that a feasible strict local minimum of the exact penalty function coincides with a local solution of the NLP (1) if the value of the *penalty parameter* $\rho > 0$ is chosen sufficiently large [14].

In order to avoid the determination of an appropriate value of the penalty parameter ρ, Fletcher and Leyffer [9] propose the concept of a *filter method* in the context of a trust region SQP algorithm. In the remainder of this section we describe how this concept can be applied to the line search framework outlined above.

The underlying idea is to interpret the NLP (1) as a biobjective optimization problem with two goals: minimizing the constraint violation $\theta(x)$ and minimizing the objective function $f(x)$. A certain emphasis is placed on the first measure, since a point has to be feasible in order to be an optimal solution of the NLP. Here, we do not require that a trial point $x_k(\alpha_{k,l})$ provides progress in a merit function such as (7), which combines these two goals as a linear combination into one single measure. Instead, following Fletcher and Leyffer's original idea, the trial point $x_k(\alpha_{k,l})$ is accepted if it improves feasibility, i.e., if $\theta(x_k(\alpha_{k,l})) < \theta(x_k)$, *or* if it improves the objective function, i.e., if $f(x_k(\alpha_{k,l})) < f(x_k)$. Note that this criterion is less demanding than the enforcement of decrease in the penalty function (7) and might in general allow larger steps.

Of course, this simple concept is not sufficient to guarantee global convergence. Several precautions have to be added, as we outline in the following; these are closely related to those proposed in [7]. The overall line search filter algorithm is formally stated in section 2.4.

2.1. Sufficient reduction. Line search methods that use a merit function ensure *sufficient* progress toward the solution. For example, they may do so by enforcing an Armijo condition for the exact penalty function (7) (see, e.g., [17]). Here, we borrow the idea from [7, 10] and replace this condition by requiring that the next iterate provides at least as much progress in one of the measures θ or f that corresponds to a small fraction of the current constraint violation, $\theta(x_k)$. More precisely, for fixed constants $\gamma_\theta, \gamma_f \in (0,1)$, we say that a trial step size $\alpha_{k,l}$ provides sufficient reduction with respect to the current iterate x_k if

$$\theta(x_k(\alpha_{k,l})) \leq (1-\gamma_\theta)\theta(x_k) \tag{8a}$$

or

$$f(x_k(\alpha_{k,l})) \leq f(x_k) - \gamma_f \theta(x_k). \tag{8b}$$

In a practical implementation, the constants γ_θ, γ_f typically are chosen to be small. However, relying solely on this criterion would allow the acceptance of a sequence $\{x_k\}$ that always provides *sufficient reduction* of the constraint violation (8a) alone, and not the objective function. This could result in convergence to a feasible but nonoptimal point. In order to prevent this, we change to a different sufficient reduction criterion whenever for the current trial step size $\alpha_{k,l}$ the *f-type switching condition*

$$m_k(\alpha_{k,l}) < 0 \qquad \text{and} \qquad [-m_k(\alpha_{k,l})]^{s_f}[\alpha_{k,l}]^{1-s_f} > \delta\,[\theta(x_k)]^{s_\theta} \tag{9}$$

holds with fixed constants $\delta > 0, s_\theta > 1, s_f \geq 1$, where

$$m_k(\alpha) := \alpha g_k^T d_k \tag{10}$$

is the linear model of the objective function f in the direction d_k. We choose to formulate the f-type switching condition (9) in terms of a general model $m_k(\alpha)$ as it allows us later, in section 4.1, to define the algorithm for an alternative measure that replaces "$f(x)$."

If the condition (9) holds, the step d_k is a descent direction for the objective function. Then, instead of insisting on (8), we require that $\alpha_{k,l}$ satisfies the Armijo-type condition

$$f(x_k(\alpha_{k,l})) \leq f(x_k) + \eta_f m_k(\alpha_{k,l}). \tag{11}$$

Here, $\eta_f \in (0, \frac{1}{2})$ is a fixed constant. It is possible that for several trial step sizes $\alpha_{k,l}$ with $l = 1, \ldots, \tilde{l}$, condition (9) but not (11) is satisfied. In this case we note that for smaller step sizes the f-type switching condition (9) may no longer be valid, so that the method reverts to the acceptance criterion (8).

The second part of the switching condition (9) deserves some discussion. It ensures that the progress for the objective function enforced by the Armijo condition (11) is sufficiently large compared to the current constraint violation. In this way, the decrease in the objective function from (11) cannot be arbitrarily small at points remote from the feasible region. Note that if we choose $s_f = 1$, condition (9) simplifies to "$-m_k(\alpha_{k,l}) > \delta[\theta(x_k)]^{s_\theta}$" and relates the progress predicted by the linear model of f for the step size $\alpha_{k,l}$ to a power of the constraint violation. This is identical to the condition used in filter trust region methods proposed in [7], except that a quadratic model is used there. However, the analysis presented below allows for larger and maybe less intuitive values of s_f. In particular, we might choose $s_f > 2s_\theta$, as required for the local convergence analysis in the companion paper [26]. This choice of s_f makes it possible to show that, close to a local solution, the condition (9) holds true only if a full step, possibly improved by a second order correction step, satisfies (11) and is accepted.

In accordance with previous publications on filter methods (e.g., [7, 10]), we call $\alpha_{k,l}$ an "f-step size" if it satisfies the f-type switching condition (9), indicating that then decrease of the objective function is required. Similarly, if an f-step size $\alpha_{k,l}$ is accepted as the final step size α_k in iteration k, we refer to k as an "f-type iteration."

2.2. Filter as taboo region. Beside requiring sufficient decrease with respect to the current iterate, the filter line search algorithm also needs to avoid cycling. For example, cycling may occur between two points that alternatingly improve one of the measures θ and f and worsen the other one. For this purpose, Fletcher and Leyffer [9] define a "taboo region" in the half-plane $\{(\theta, f) \in \mathbb{R}^2 : \theta \geq 0\}$. They maintain a list of $(\theta(x_p), f(x_p))$-pairs (called *filter*) corresponding to (some of) the previous iterates x_p and require that a point, in order to be accepted, has to improve at least one of the two measures compared to those previous iterates. In other words, a trial step $x_k(\alpha_{k,l})$ can be accepted only if

$$\theta(x_k(\alpha_{k,l})) < \theta(x_p)$$

or

$$f(x_k(\alpha_{k,l})) < f(x_p)$$

for all $(\theta(x_p), f(x_p))$ in the current filter.

In contrast to the notation in [7, 9], for the sake of a simplified notation we define the filter in this paper not as a list but as a *set* $\mathcal{F}_k \subseteq [0, \infty) \times \mathbb{R}$ containing *all* (θ, f)-pairs that are "prohibited" in iteration k. We say that a trial point $x_k(\alpha_{k,l})$ is *acceptable to the filter* if its (θ, f)-pair does not lie in the taboo region, i.e., if

$$\big(\theta(x_k(\alpha_{k,l})), f(x_k(\alpha_{k,l}))\big) \notin \mathcal{F}_k. \tag{12}$$

During the optimization we make sure that the current iterate x_k is always acceptable to the current filter $\mathcal{F}_k$.

At the beginning of the optimization, the filter is initialized to be empty: $\mathcal{F}_0 := \emptyset$ or—if one wants to impose an explicit upper bound on the constraint violation—as $\mathcal{F}_0 := \{(\theta, f) \in \mathbb{R}^2 : \theta \geq \theta_{\max}\}$ for some $\theta_{\max} > \theta(x_0)$. Throughout the optimization the filter is then augmented in some iterations after the new iterate x_{k+1} has been accepted. For this, the updating formula

$$\mathcal{F}_{k+1} := \mathcal{F}_k \cup \Big\{(\theta, f) \in \mathbb{R}^2 : \theta \geq (1-\gamma_\theta)\theta(x_k) \quad \text{and} \quad f \geq f(x_k) - \gamma_f \theta(x_k)\Big\} \tag{13}$$

is used (see also [7]). If the filter is not augmented, it remains unchanged, i.e., $\mathcal{F}_{k+1} := \mathcal{F}_k$. Note that then $\mathcal{F}_k \subseteq \mathcal{F}_{k+1}$ for all k. This ensures that all later iterates will have to provide sufficient reduction with respect to x_k as defined by criterion (8), if the filter has been augmented in iteration k. Note that for a practical implementation it is sufficient to store the "corner entries"

$$\big((1-\gamma_\theta)\theta(x_k),\ f(x_k) - \gamma_f \theta(x_k)\big). \tag{14}$$

It remains to decide which iterations should augment the filter. In order to keep the filter approach less conservative, we do not want to augment the filter in every iteration. In addition, as we see in the discussion of the next safeguard below, it is important for the proposed method that we never include feasible points in the filter. The following rule from [7] is motivated by these considerations.

We always augment the filter if the current iteration is not an f-type iteration, i.e., if for the accepted trial step size α_k the f-type switching condition (9) does not hold. Otherwise, the Armijo condition (11) must be satisfied, and the value of the objective function is strictly decreased. To see that this indeed prevents cycling let us assume for a moment that the algorithm generates a cycle of length l,

$$x_K, x_{K+1}, \ldots, x_{K+l-1}, x_{K+l} = x_K, x_{K+l+1} = x_{K+1}, \ldots . \tag{15}$$

Since a point x_k can never be reached again if the filter is augmented in iteration k, the existence of a cycle would imply that the filter is not augmented for all $k \geq K$. However, this would imply that $f(x_k)$ is a strictly decreasing sequence for $k \geq K$, giving a contradiction, so that (15) cannot be a cycle.

2.3. Feasibility restoration phase.

If the linear system (3) is consistent, d_k satisfies the linearization of the constraints and we have $\theta(x_k(\alpha_{k,l})) < \theta(x_k)$ whenever $\alpha_{k,l} > 0$ is sufficiently small. It is not guaranteed, however, that there exists a trial step size $\alpha_{k,l} > 0$ that indeed provides *sufficient* reduction as defined by criterion (8).

In this situation, where no admissible step size can be found, the method switches to a *feasibility restoration phase*, whose purpose is to find a new iterate x_{k+1} that satisfies (8) and is also acceptable to the current filter by trying to decrease the constraint violation. In this paper, we do not specify the particular procedure for this feasibility restoration phase. It could be any iterative algorithm with the goal of finding a less infeasible point, and different methods could even be used at different stages of the optimization procedure. For example, a nonlinear optimization algorithm might be applied to minimize θ, possibly ignoring the objective function. If the feasibility restoration phase terminates successfully by delivering a new admissible iterate, the filter is augmented according to (13) to avoid cycling back to the problematic point x_k.

Since a feasible iterate is never included in the filter (see Lemma 4 below), it is reasonable to assume that a suitable feasibility restoration phase algorithm is either able to find a new acceptable iterate satisfying (8) or converges to a local minimizer (or at least a stationary point) for some measure of infeasibility. The latter case may be important information for the user, as it indicates that the problem seems (at least locally) infeasible. This is, of course, no guarantee that the problem possesses no feasible point; proving infeasibility is as difficult as finding a global minimizer and beyond the capabilities of methods for finding local solutions like those discussed in this paper. However, we believe that it is a desirable practical feature of a nonlinear optimization code to return at least a local minimizer of the constraint violation if the method fails to find a solution of the optimization problem, instead of terminating at a less informative and possibly random point.

In order to detect the situation where no admissible step size can be found and the restoration phase has to be invoked, we propose the following rule. Consider the case when the current trial step size $\alpha_{k,l}$ is still large enough that the f-type switching condition (9) holds for some $\alpha \leq \alpha_{k,l}$. In this case, we do not switch to the feasibility restoration phase, since there is still the chance that a shorter step length might be accepted by the Armijo condition (11). Therefore, we can see from the f-type switching condition (9) and the definition of m_k (10) that we do not want to revert to the feasibility restoration phase if $g_k^T d_k < 0$ and

$$\alpha_{k,l} > \frac{\delta[\theta(x_k)]^{s_\theta}}{[-g_k^T d_k]^{s_f}}. \tag{16}$$

However, if the f-type switching condition (9) is not satisfied for the current trial step size $\alpha_{k,l}$ and all shorter trial step sizes, then the decision whether to switch to the feasibility restoration phase is based on the linear approximations

$$\tilde{\theta}(x_k + \alpha d_k) = \theta(x_k) - \alpha\theta(x_k), \tag{17a}$$

$$\tilde{f}(x_k + \alpha d_k) = f(x_k) + \alpha g_k^T d_k. \tag{17b}$$

(Note that indeed $\tilde{\theta}(x_k + \alpha d_k) = \theta(x_k + \alpha d_k) + O(\alpha^2)$, since $A_k^T d_k + c(x_k) = 0$ from (3)). Substituting (17a) into the sufficient decrease condition for the infeasibility measure (8a) indicates that (8a) may not be satisfied for step sizes satisfying $\alpha_{k,l} \leq \gamma_\theta$. Similarly, in case $g_k^T d_k < 0$, the sufficient decrease criterion for the objective function (8b) may not be satisfied for step sizes satisfying

$$\alpha_{k,l} \leq \frac{\gamma_f \theta(x_k)}{-g_k^T d_k}.$$

We can summarize this in the following formula for a minimal trial step size

$$\alpha_k^{\min} := \gamma_\alpha \cdot \begin{cases} \min\left\{\gamma_\theta, \frac{\gamma_f \theta(x_k)}{-g_k^T d_k}, \frac{\delta[\theta(x_k)]^{s_\theta}}{[-g_k^T d_k]^{s_f}}\right\} & \text{if } g_k^T d_k < 0, \\ \gamma_\theta & \text{otherwise} \end{cases} \tag{18}$$

and switch to the feasibility restoration phase when $\alpha_{k,l}$ becomes smaller than $\alpha_k^{\min}$. Here, $\gamma_\alpha \in (0, 1]$ is a safety factor that might be useful in a practical implementation in order to compensate for the neglected higher order terms in the linearization (17) and to avoid invoking the feasibility restoration phase unnecessarily.

It is possible, however, to employ more sophisticated rules to decide when to switch to the feasibility restoration phase while still maintaining the convergence properties. These rules could, for example, be based on higher order approximations of θ and/or f. We need only ensure that the algorithm does not switch to the feasibility restoration phase as long as (9) holds for a step size $\alpha \le \alpha_{k,l}$ where $\alpha_{k,l}$ is the current trial step size, and that the backtracking line search procedure is finite; i.e., it eventually either delivers a new iterate x_{k+1} or reverts to the feasibility restoration phase.

The proposed method also allows us to switch to the feasibility restoration phase in any iteration in which the infeasibility $\theta(x_k)$ does not become arbitrarily small. For example, this might be necessary when the Jacobian of the constraints A_k^T is (nearly) rank-deficient, so that the linear system (3) is (nearly) singular and no search direction can be computed. For the purpose of the analysis we assume that the algorithm is able to detect a situation in which the singular values of A_k become arbitrarily small and switch to the restoration phase in that case, even if the linear system can be solved numerically (see Assumption (G4) below). The search direction from (3) might still be used to generate the next iterate x_{k+1} using (5), as long as $x_{k+1} \notin \mathcal{F}_k$ and (8) can be satisfied. Even though we could consider this a non-f-type iteration, we formally treat this case as if the restoration phase is called. (Note that the iterate x_{k+1} returned from the restoration phase does not necessarily have to satisfy (8a) if (8b) holds instead.)

2.4. The algorithm.

We are now ready to formally state the overall algorithm for solving the equality constrained NLP (1).

Algorithm I

Given: Starting point x_0; constants $\theta_{\max} \in (\theta(x_0), \infty]$; $\gamma_\theta, \gamma_f \in (0, 1)$; $\delta > 0$; $\gamma_\alpha \in (0, 1]$; $s_\theta > 1$; $s_f \ge 1$; $\eta_f \in (0, \frac{1}{2})$; $0 < \tau_1 \le \tau_2 < 1$.

1. *Initialize.* Initialize the filter $\mathcal{F}_0 := \{(\theta, f) \in \mathbb{R}^2 : \theta \ge \theta_{\max}\}$ and the iteration counter $k \leftarrow 0$.
2. *Check convergence.* Stop if x_k is a stationary point of the NLP (1), i.e., if it satisfies the KKT conditions (2) for some $\lambda \in \mathbb{R}^m$.
3. *Compute search direction.* Compute the search direction d_k from the linear system (3). If this system is detected to be too ill-conditioned (see the assumptions in the next section), go to the feasibility restoration phase in step 8.
4. *Backtracking line search.*
   - 4.1. *Initialize line search.* Set $\alpha_{k,0} = 1$ and $l \leftarrow 0$.
   - 4.2. *Compute new trial point.* If the trial step size becomes too small, i.e., $\alpha_{k,l} < \alpha_k^{\min}$ with $\alpha_k^{\min}$ defined by (18), go to the feasibility restoration phase in step 8. Otherwise, compute the new trial point $x_k(\alpha_{k,l}) = x_k + \alpha_{k,l} d_k$.

4.3. *Check acceptability to the filter.* If $x_k(\alpha_{k,l}) \in \mathcal{F}_k$, reject the trial step size and go to step 4.5.

4.4. *Check sufficient decrease with respect to current iterate.*

4.4.1. *Case* I: $\alpha_{k,l}$ *is an* f*-step size (i.e.,* (9) *holds):* If the Armijo condition (11) for the objective function holds, accept the trial step and go to step 5.
Otherwise, go to step 4.5.

4.4.2. *Case* II: $\alpha_{k,l}$ *is not an* f*-step size (i.e.,* (9) *is not satisfied):* If (8) holds, accept the trial step and go to step 5.
Otherwise, go to step 4.5.

4.5. *Choose new trial step size.* Choose $\alpha_{k,l+1} \in [\tau_1 \alpha_{k,l}, \tau_2 \alpha_{k,l}]$, set $l \leftarrow l+1$, and go back to step 4.2.

5. *Accept trial point.* Set $\alpha_k := \alpha_{k,l}$ and $x_{k+1} := x_k(\alpha_k)$.

6. *Augment filter if necessary.* If k is not an f-type iteration, augment the filter using (13); otherwise leave the filter unchanged, i.e., set $\mathcal{F}_{k+1} := \mathcal{F}_k$.
(Note that steps 4.3 and 4.4.2 ensure that $(\theta(x_{k+1}), f(x_{k+1})) \notin \mathcal{F}_{k+1}$.)

7. *Continue with next iteration.* Increase the iteration counter $k \leftarrow k+1$ and go back to step 2.

8. *Feasibility restoration phase.* Compute a new iterate x_{k+1} by decreasing the infeasibility measure θ so that x_{k+1} satisfies the sufficient decrease conditions (8) and is acceptable to the filter, i.e., $(\theta(x_{k+1}), f(x_{k+1})) \notin \mathcal{F}_k$. Augment the filter using (13) (for x_k) and continue with the regular iteration in step 7.

2.5. Remarks. *Remark* 1. From step 4.5 it is clear that $\lim_l \alpha_{k,l} = 0$. In the case that $\theta(x_k) > 0$, it can be seen from (18) that $\alpha_k^{\min} > 0$. Therefore, the algorithm either accepts a new iterate in step 4.4 or switches to the feasibility restoration phase. If, on the other hand, $\theta(x_k) = 0$ and the algorithm does not stop in step 2 at a KKT point, then the positive definiteness of H_k on the null space of A_k^T implies that $g_k^T d_k < 0$ (see, e.g., Lemma 4 below). In that case, $\alpha_k^{\min} = 0$, and the Armijo condition (11) is satisfied for a sufficiently small step size $\alpha_{k,l}$; i.e., a new iterate is accepted in step 4.4.1. Overall, we see that the inner loop in step 4 always terminates after a finite number of trial steps, and the algorithm is well defined.

Remark 2. The algorithm generates an infinite sequence $\{x_k\}$ of iterates, unless it encounters a KKT point and terminates in step 2, or if the feasibility restoration phase in step 8 is not able to return a new iterate. In the latter case, the restoration phase algorithm converges to a stationary point for the constraint violation, assuming that a suitable method is used.

Remark 3. The mechanisms of the filter ensure that $(\theta(x_k), f(x_k)) \notin \mathcal{F}_k$ for all k. Furthermore, the initialization of the filter in step 1 and the update rule (13) imply that for all k the filter has the following property:

$$(\bar{\theta}, \bar{f}) \notin \mathcal{F}_k \quad \Longrightarrow \quad (\theta, f) \notin \mathcal{F}_k \quad \text{if} \quad \theta \le \bar{\theta} \quad \text{and} \quad f \le \bar{f}. \tag{19}$$

Remark 4. For practical purposes, it might not be efficient to restrict the step size by enforcing an Armijo-type decrease (11) in the objective function if the current constraint violation is not small. It is possible to change the algorithm so that the step acceptance criterion is always (8), unless the f-type switching condition (9) holds *and* $\theta(x_k) \le \theta_{\text{sml}}$ for some fixed $\theta_{\text{sml}} > 0$, in which case the Armijo condition (11) has to be satisfied. In this modified method, the filter is augmented (using (13)), whenever

(9) or (11) does not hold. The global convergence properties are not affected by this modification.

Remark 5. The proposed method shares many similarities with the trust region filter SQP method proposed and analyzed in [7]. However, we discuss a more general f-type switching rule (9) in order to be able to show fast local convergence in the companion paper [26]. Further differences result from the fact that the proposed method follows a line search approach, so that in contrast to [7] the actual step taken does not necessarily satisfy the linearization of the constraints; i.e., we might have $A_k^T(x_k - x_{k+1}) \neq c(x_k)$ in some iterations. As a related consequence, the condition when to switch to the feasibility restoration phase in step 4.2 could not be chosen to be the detection of infeasibility of the trust region QP but has to be defined by means of a minimal step size (18). Due to these differences, the global convergence analysis presented in [7] does not apply to the proposed line search filter method.

3. Global convergence.

3.1. Assumptions.

In the remainder of this paper we denote the set of indices of those iterations in which the filter has been augmented by $\mathcal{A} \subseteq \mathbb{N}$; i.e.,

$$\mathcal{F}_k \subsetneqq \mathcal{F}_{k+1} \qquad \Longleftrightarrow \qquad k \in \mathcal{A}.$$

The set $\mathcal{R} \subseteq \mathbb{N}$ is defined as the set of all iteration indices in which the feasibility restoration phase is invoked. Since step 8 makes sure that the filter is augmented in every iteration in which the restoration phase is invoked, we have $\mathcal{R} \subseteq \mathcal{A}$. We denote with $\mathcal{R}_{\text{inc}} \subseteq \mathcal{R}$ the set of those iteration counters in which the restoration phase is invoked from step 3.

Let us now state the assumptions necessary for the global convergence analysis of Algorithm I. We first state these assumptions in technical terms and discuss their practical relevance afterwards.

ASSUMPTIONS G. *Let $\{x_k\}$ be the sequence generated by Algorithm* I, *where we assume that the feasibility restoration phase in step* 8 *always terminates successfully and that the algorithm does not stop in step* 2 *at a KKT point.*

(G1) *There exists an open set $\mathcal{C} \subseteq \mathbb{R}^n$ with $[x_k, x_k + d_k] \subseteq \mathcal{C}$ for all $k \notin \mathcal{R}_{\text{inc}}$ so that f and c are differentiable on $\mathcal{C}$, and their function values, as well as their first derivatives, are bounded and Lipschitz-continuous over $\mathcal{C}$.*

(G2) *The matrices H_k approximating the Hessian of the Lagrangian in* (3) *are uniformly bounded for all $k \notin \mathcal{R}_{\text{inc}}$.*

(G3) *The Hessian approximations H_k are uniformly positive definite on the null space of the Jacobian A_k^T. In other words, there exists a constant $M_H > 0$ so that for all $k \notin \mathcal{R}_{\text{inc}}$*

$$\lambda_{\min}\left(Z_k^T H_k Z_k\right) \geq M_H, \tag{20}$$

where the columns of $Z_k \in \mathbb{R}^{n \times (n-m)}$ form an orthonormal basis matrix of the null space of A_k^T.

(G4) *There exists a constant $M_A > 0$ so that for all $k \notin \mathcal{R}_{\text{inc}}$ we have*

$$\sigma_{\min}(A_k) \geq M_A. \tag{21}$$

(G5) *The iterates for which the restoration phase is invoked from step* 3 *(for example, because* (20) *or* (21) *is violated) are not arbitrarily close to the feasible region. In other words, there exists a constant $\theta_{\text{inc}} > 0$ so that $k \notin \mathcal{R}_{\text{inc}}$ whenever $\theta(x_k) \leq \theta_{\text{inc}}$.*

Assumptions (G1) and (G2) merely establish smoothness and boundedness of the problem data. As we see later in Lemma 2, Assumption (G3) ensures a certain descent property and is similar to common assumptions on the reduced Hessian in SQP line search methods (see, e.g., [17]). To guarantee this requirement in a practical implementation, one could compute a QR-factorization of A_k to obtain matrices $Y_k \in \mathbb{R}^{n\times m}$ and $Z_k \in \mathbb{R}^{n\times(n-m)}$ so that the columns of $[Y_k\ Z_k]$ form an orthonormal basis of $\mathbb{R}^n$ and the columns of Z_k are a basis of the null space of A_k^T (see, e.g., [11]). Then, the overall search direction

$$d_k = q_k + p_k \tag{22a}$$

can be decomposed into the two orthogonal components

$$q_k := Y_k\bar{q}_k \quad \text{and} \quad p_k := Z_k\bar{p}_k, \tag{22b}$$

with

$$\bar{q}_k := -\left[A_k^T Y_k\right]^{-1} c_k, \tag{23a}$$

$$\bar{p}_k := -\left[Z_k^T H_k Z_k\right]^{-1} Z_k^T \left(g_k + H_k q_k\right) \tag{23b}$$

(see, e.g., [17]). The eigenvalues for the reduced Hessian in (23b) (the term in square brackets) could be monitored and modified if necessary. However, this procedure is prohibitive for large-scale problems, and in those cases one instead might employ heuristics to ensure at least positive definiteness of the reduced Hessian, for example, by monitoring and possibly modifying the inertia of the iteration matrix in (3) (see, e.g., [23]). Note, on the other hand, that (20) holds in the neighborhood of a local solution x_* satisfying the sufficient second order optimality conditions (see, e.g., [17]) if H_k approaches the exact Hessian of the Lagrangian of the NLP (1). Then, close to x_*, no eigenvalue correction is necessary and fast local convergence can be expected, assuming that full steps are taken close to x_*. See the companion paper [26] for a local convergence analysis of the presented method.

In the description of the algorithm in section 2.4 we did not specify precisely when the method switches in step 3 to the feasibility restoration phase, since there might be several practical implementations compatible with Assumptions G. For completeness, one possible option is outlined next. By monitoring and possibly modifying the eigenvalues of the reduced Hessian it is possible to make sure that (20) is valid in *every* iteration. Similarly, we can guarantee that the *entire* sequence $\{H_k\}$ is uniformly bounded. Let us further make the assumption (on the problem statement) that the gradients of the constraints are uniformly linearly independent for all iterates x_k close to the feasible region; i.e., there exist constants $b_1, b_2 > 0$ so that

$$\theta(x_k) \le b_1 \qquad \Longrightarrow \qquad \sigma_{\min}(A_k) \ge b_2.$$

Then, if we decide in step 3 to invoke the feasibility restoration phase whenever $\sigma_{\min}(A_k) \le b_3\theta(x_k)$ for some fixed constant $b_3 > 0$, then Assumptions G hold (with $M_A = \min\{b_2, b_1 b_3\}$ and $\theta_{\text{inc}} = \frac{M_A}{2b_3}$).

3.2. Preliminary results. Similar to the analysis in [7], we make use of a *first order criticality measure* $\chi(x_k) \in [0,\infty]$ with the property that if a subsequence $\{x_{k_i}\}$ of iterates with $\chi(x_{k_i}) \to 0$ converges to a feasible limit point x_*, then x_* corresponds to a KKT solution. In the case of Algorithm I, this means that there exist λ_* so that the KKT conditions (2) are satisfied for (x_*, λ_*).

For the convergence analysis of the filter method we define the criticality measure for iterations $k \notin \mathcal{R}_{\rm inc}$ as

$$\chi(x_k) := \|\bar{p}_k\|_2 \,, \tag{24}$$

with $\bar{p}_k$ from (23b). Note that this definition is unique, since p_k in (22a) is unique due to the orthogonality of Y_k and Z_k, and since $\|\bar{p}_k\|_2 = \|p_k\|_2$ due to the orthonormality of Z_k. For completeness, we define $\chi(x_k) := \infty$ for $k \in \mathcal{R}_{\rm inc}$.

In order to see that $\chi(x_k)$ defined in this way is indeed a criticality measure under Assumptions G, let us consider a subsequence of iterates $\{x_{k_i}\}$ with $\lim_i \chi(x_{k_i}) = 0$ and $\lim_i x_{k_i} = x_*$ for some feasible limit point x_*. Since $\chi(x_{k_i}) = \infty$ if $k_i \in \mathcal{R}_{\rm inc}$, we then have $k_i \notin \mathcal{R}_{\rm inc}$ for i sufficiently large. Furthermore, from Assumption (G4) and (23a) we have $\lim_i \bar{q}_{k_i} = 0$, and then from $\lim_i \chi(x_{k_i}) = 0$, (24), (23b), and Assumption (G3) we have that $\lim_{i\to\infty} \|Z_{k_i}^T g_{k_i}\| = 0$, which is a well-known optimality measure (see, e.g., [17]).

Before we begin the global convergence analysis, let us state some preliminary results.

Lemma 1. *Suppose Assumptions* G *hold. Then there exist constants M_d, M_λ, $M_m > 0$, such that*

$$\|d_k\| \le M_d, \qquad \|\lambda_k^+\| \le M_\lambda, \qquad |m_k(\alpha)| \le M_m\alpha \tag{25}$$

for all $k \notin \mathcal{R}_{\rm inc}$ and $\alpha \in (0,1]$.

Proof. From (G1) we have that the right-hand side of (3) is uniformly bounded. Additionally, Assumptions (G2), (G3), and (G4) guarantee that the inverse of the matrix in (3) exists and is uniformly bounded for all $k \notin \mathcal{R}_{\rm inc}$. Consequently, the solution of (3), (d_k, λ_k^+), is uniformly bounded, and therefore also $m_k(\alpha)/\alpha = g_k^T d_k$. □

The following result shows that the search direction is a direction of sufficient descent for the objective function at points that are sufficiently close to feasible and nonoptimal.

Lemma 2. *Suppose Assumptions* G *hold. If $\{x_{k_i}\}$ is a subsequence of iterates for which $\chi(x_{k_i}) \ge \epsilon$ with a constant $\epsilon > 0$ independent of i, then there exist constants $\epsilon_1, \epsilon_2 > 0$, such that*

$$\theta(x_{k_i}) \le \epsilon_1 \qquad \Longrightarrow \qquad m_{k_i}(\alpha) \le -\epsilon_2\alpha$$

for all i and $\alpha \in (0,1]$.

Proof. Consider a subset $\{x_{k_i}\}$ of iterates with $\chi(x_{k_i}) = \|\bar{p}_{k_i}\|_2 \ge \epsilon$. Then, by Assumption (G5), for all x_{k_i} with $\theta(x_{k_i}) \le \theta_{\rm inc}$ we have $k_i \notin \mathcal{R}_{\rm inc}$. Furthermore, with $q_{k_i} = O(\|c(x_{k_i})\|)$ (from (23a) and Assumption (G4)) it follows that for $k_i \notin \mathcal{R}_{\rm inc}$

$$\begin{aligned}
m_{k_i}(\alpha)/\alpha &= g_{k_i}^T d_{k_i} \overset{(22)}{=} g_{k_i}^T Z_{k_i}\bar{p}_{k_i} + g_{k_i}^T q_{k_i} && \text{(26a)}\\
&\overset{(23b)}{=} -\bar{p}_{k_i}^T \left[Z_{k_i}^T H_{k_i} Z_{k_i}\right]\bar{p}_{k_i} - \bar{p}_{k_i}^T Z_{k_i}^T H_{k_i} q_{k_i} + g_{k_i}^T q_{k_i} && \text{(26b)}\\
&\overset{(G2),(G3)}{\le} -c_1\,\|\bar{p}_{k_i}\|_2^2 + c_2\,\|\bar{p}_{k_i}\|_2\,\|c_{k_i}\| + c_3\|c_{k_i}\| && \text{(26c)}\\
&\le \chi(x_{k_i})\left(-\epsilon\, c_1 + c_2\theta(x_{k_i}) + \frac{c_3}{\epsilon}\theta(x_{k_i})\right) && \text{(26d)}
\end{aligned}$$

for some constants $c_1, c_2, c_3 > 0$, where we used $\chi(x_{k_i}) \ge \epsilon$ in the last inequality. If we now define

$$\epsilon_1 := \min\left\{\theta_{\rm inc}, \frac{\epsilon^2\, c_1}{2(c_2\,\epsilon + c_3)}\right\},$$

it follows for all x_{k_i} with $\theta(x_{k_i}) \le \epsilon_1$ that

$$m_{k_i}(\alpha) \le -\alpha \frac{\epsilon\, c_1}{2} \chi(x_{k_i}) \le -\alpha \frac{\epsilon^2\, c_1}{2}.$$

The claim follows after defining $\epsilon_2 := \frac{\epsilon^2\, c_1}{2}$. □

LEMMA 3. *Suppose Assumption* (G1) *holds. Then there exist constants* $C_\theta, C_f > 0$ *so that for all* $k \notin \mathcal{R}_{\rm inc}$ *and* $\alpha \le 1$

$$|\theta(x_k + \alpha d_k) - (1-\alpha)\theta(x_k)| \le C_\theta \alpha^2 \|d_k\|^2, \tag{27a}$$

$$|f(x_k + \alpha d_k) - f(x_k) - m_k(\alpha)| \le C_f \alpha^2 \|d_k\|^2. \tag{27b}$$

These inequalities follow directly from second order Taylor expansions and (3).

Finally, we show that step 8 (feasibility restoration phase) of Algorithm I is well defined. Unless the feasibility restoration phase terminates at a stationary point of the constraint violation it is essential that reducing the infeasibility measure $\theta(x)$ eventually leads to a point that is acceptable to the filter. This is guaranteed by the following lemma which shows that no (θ, f)-pair corresponding to a feasible point is ever included in the filter.

LEMMA 4. *Suppose Assumptions* G *hold. Then*

$$\theta(x_k) = 0 \quad \Longrightarrow \quad m_k(\alpha) < 0 \tag{28}$$

and

$$\Theta_k := \min\{\theta : (\theta, f) \in \mathcal{F}_k\} > 0 \tag{29}$$

for all k *and* $\alpha \in (0,1]$.

Proof. If $\theta(x_k) = 0$, we have from Assumption (G5) that $k \notin \mathcal{R}_{\rm inc}$. In addition, $\chi(x_k) > 0$ then follows because Algorithm I would have terminated otherwise in step 2, in contrast to Assumptions G. Considering the decomposition (22), it follows, as in (26), that

$$m_k(\alpha)/\alpha = g_k^T d_k \le -c_1 \chi(x_k)^2 < 0;$$

i.e., (28) holds.

The proof of (29) is by induction. It is clear from step 1 of Algorithm I that the claim is valid for $k = 0$ since $\theta_{\max} > 0$. Suppose the claim is true for k. Then, if $\theta(x_k) > 0$ and the filter is augmented in iteration k, it is clear from the update rule (13) that $\Theta_{k+1} > 0$, since $\gamma_\theta \in (0,1)$. If, on the other hand, $\theta(x_k) = 0$, we have from (28) that $m_k(\alpha) < 0$ for all $\alpha \in (0,1]$ so that the f-type switching condition (9) is true for all trial step sizes. Therefore, step 4.4 always considers Case I, and the reason for α_k having been accepted must have been that α_k satisfies (11). Consequently, the filter is not augmented in step 6. Hence, $\Theta_{k+1} = \Theta_k > 0$. □

3.3. Feasibility. In this section we show that under Assumptions G the sequence $\theta(x_k)$ converges to zero; i.e., all limit points of $\{x_k\}$ are feasible.

LEMMA 5. *Suppose that Assumptions* G *hold and that the filter is augmented only a finite number of times, i.e.,* $|\mathcal{A}| < \infty$. *Then*

$$\lim_{k\to\infty} \theta(x_k) = 0. \tag{30}$$

Proof. Choose K so that for all iterations $k \geq K$ the filter is not augmented in iteration k; in particular, $k \notin \mathcal{R}_{\text{inc}} \subseteq \mathcal{A}$ for $k \geq K$. From step 6 in Algorithm I we then have that for all $k \geq K$ both conditions (9) and (11) are satisfied for α_k. We now distinguish two cases, where $k \notin \mathcal{A}$.

Case 1 ($s_f > 1$). From (9) it follows with M_m from Lemma 1 that

$$\delta[\theta(x_k)]^{s_\theta} < [-m_k(\alpha_k)]^{s_f}[\alpha_k]^{1-s_f} \leq M_m^{s_f}\alpha_k$$

and hence (since $1 - 1/s_f > 0$)

$$c_4[\theta(x_k)]^{s_\theta - \frac{s_\theta}{s_f}} < [\alpha_k]^{1-\frac{1}{s_f}} \qquad \text{with} \qquad c_4 := \left(\frac{\delta}{M_m^{s_f}}\right)^{1-\frac{1}{s_f}}.$$

This implies

$$\begin{aligned} f(x_{k+1}) - f(x_k) &\overset{(11)}{\leq} \eta_f m_k(\alpha_k) \\ &\overset{(9)}{<} -\eta_f \delta^{\frac{1}{s_f}}[\alpha_k]^{1-\frac{1}{s_f}}[\theta(x_k)]^{\frac{s_\theta}{s_f}} \\ &< -\eta_f \delta^{\frac{1}{s_f}} c_4 [\theta(x_k)]^{s_\theta}. \end{aligned}$$

Case 2 ($s_f = 1$). From (9) we have $\delta[\theta(x_k)]^{s_\theta} < -m_k(\alpha_k)$ so that from (11) we immediately obtain $f(x_{k+1}) - f(x_k) < -\eta_f \delta[\theta(x_k)]^{s_\theta}$.

In either case, we have for all $k \notin \mathcal{A}$ that

$$f(x_{k+1}) - f(x_k) < -\tilde{c}_4[\theta(x_k)]^{s_\theta} \tag{31}$$

for some $\tilde{c}_4 > 0$. Hence, for all $i = 1, 2, \dots$,

$$\begin{aligned} f(x_{K+i}) &= f(x_K) + \sum_{k=K}^{K+i-1}(f(x_{k+1}) - f(x_k)) \\ &< f(x_K) - \tilde{c}_4 \sum_{k=K}^{K+i-1}[\theta(x_k)]^{s_\theta}. \end{aligned}$$

Since $f(x_{K+i})$ is bounded below as $i \to \infty$, the series on the right-hand side in the last line is bounded, which in turn implies (30). □

Note that this result could be obtained with a simpler proof if the model $m_k(\alpha)$ has the particular form (10), but the above version also holds for the model (56) in section 4.1.

The following lemma considers a subsequence $\{x_{k_i}\}$ with $k_i \in \mathcal{A}$ for all i.

LEMMA 6. *Let $\{x_{k_i}\}$ be a subsequence of iterates generated by Algorithm* I *so that the filter is augmented in iteration k_i; i.e., $k_i \in \mathcal{A}$ for all i. Furthermore, assume that there exist constants $c_f \in \mathbb{R}$ and $C_\theta > 0$ so that*

$$f(x_{k_i}) \geq c_f \qquad \textit{and} \qquad \theta(x_{k_i}) \leq C_\theta$$

for all i (for example, if Assumption (G1) *holds). It then follows that*

$$\lim_{i\to\infty} \theta(x_{k_i}) = 0.$$

The proof of this lemma can be found in [7, Lemma 3.3]. There the proof is stated for slightly different circumstances, but it is easy to verify that it is still valid in our context.

The previous two lemmas prepare the proof of the following theorem.

THEOREM 1. *Suppose Assumptions* G *hold. Then*

$$\lim_{k\to\infty} \theta(x_k) = 0.$$

Proof. In the case that the filter is augmented only a finite number of times, Lemma 5 implies the claim. If in the other extreme there exists some $K \in \mathbb{N}$, so that the filter is updated by (13) in *all* iterations $k \geq K$, then the claim follows from Lemma 6. It remains to consider the case where for all $K \in \mathbb{N}$ there exist $k_1, k_2 \geq K$ with $k_1 \in \mathcal{A}$ and $k_2 \notin \mathcal{A}$.

The proof is by contradiction. Suppose $\limsup_k \theta(x_k) = M > 0$. Now construct two subsequences $\{x_{k_i}\}$ and $\{x_{l_i}\}$ of $\{x_k\}$ in the following way:

1. Set $i \leftarrow 0$ and $k_{-1} = -1$.
2. Pick $k_i > k_{i-1}$ with

$$\theta(x_{k_i}) \geq M/2 \tag{32}$$

and $k_i \notin \mathcal{A}$. (Note that Lemma 6 ensures the existence of $k_i \notin \mathcal{A}$ since otherwise $\theta(x_{k_i}) \to 0$.)

3. Choose $l_i := \min\{l \in \mathcal{A} : l > k_i\}$; i.e., l_i is the first iteration after k_i in which the filter is augmented.
4. Set $i \leftarrow i+1$ and go back to step 2.

Thus, every x_{k_i} satisfies (32), and for each x_{k_i} the iterate x_{l_i} is the first iterate after x_{k_i} for which $(\theta(x_{l_i}), f(x_{l_i}))$ is included in the filter.

Since (31) holds for all $k = k_i, \ldots, l_i - 1 \notin \mathcal{A}$, we obtain for all i

$$f(x_{l_i}) \leq f(x_{(k_i+1)}) < f(x_{k_i}) - \tilde{c}_4[M/2]^{s_\theta}. \tag{33}$$

This ensures that for all $K \in \mathbb{N}$ there exists some $i \geq K$ with $f(x_{k_{(i+1)}}) \geq f(x_{l_i})$ because otherwise (33) would imply

$$f(x_{k_{(i+1)}}) < f(x_{l_i}) < f(x_{k_i}) - \tilde{c}_4[M/2]^{s_\theta}$$

for all i and consequently $\lim_i f(x_{k_i}) = -\infty$ in contradiction to the fact that $\{f(x_k)\}$ is bounded below. Thus, there exists a subsequence $\{i_j\}$ of $\{i\}$ so that

$$f(x_{k_{(i_j+1)}}) \geq f(x_{l_{i_j}}). \tag{34}$$

Since $x_{k_{(i_j+1)}} \notin \mathcal{F}_{k_{(i_j+1)}} \supseteq \mathcal{F}_{l_{i_j}}$ and $l_{i_j} \in \mathcal{A}$, it follows from (34) and the filter update rule (13) that

$$\theta(x_{k_{(i_j+1)}}) \leq (1-\gamma_\theta)\theta(x_{l_{i_j}}). \tag{35}$$

Since $l_{i_j} \in \mathcal{A}$ for all j, Lemma 6 yields $\lim_j \theta(x_{l_{i_j}}) = 0$ so that from (35) we obtain $\lim_j \theta(x_{k_{i_j}}) = 0$ in contradiction to (32). □

Remark 6. As one can easily verify, if $s_f = 1$ is chosen in the f-type switching rule (9), then the proof of the previous theorem does not actually require the assumption that $\{f(x_k)\}$ and $\{\|\nabla f(x_k)\|\}$ are bounded above (see Assumption (G1)). This is important for the discussion of the interior point method in section 4.3.

3.4. Optimality. In this section we show that Assumptions G guarantee that the optimality measure $\chi(x_k)$ is not bounded away from zero; i.e., if $\{x_k\}$ is bounded, at least one limit point is a first order optimal point for the NLP (1).

The first lemma shows conditions under which it can be guaranteed that there exists a step length bounded away from zero so that the Armijo condition (11) for the objective function is satisfied.

Lemma 7. *Suppose Assumptions* G *hold. Let $\{x_{k_i}\}$ be a subsequence with $k_i \notin \mathcal{R}_{\text{inc}}$ and $m_{k_i}(\alpha) \le -\alpha\epsilon_2$ for a constant $\epsilon_2 > 0$ independent of k_i and for all $\alpha \in (0,1]$. Then there exists some constant $\bar\alpha > 0$ so that for all k_i and $\alpha \le \bar\alpha$*

$$f(x_{k_i} + \alpha d_{k_i}) - f(x_{k_i}) \le \eta_f m_{k_i}(\alpha). \tag{36}$$

Proof. Let M_d and C_f be the constants from Lemmas 1 and 3. It then follows for all $\alpha \le \bar\alpha$ with $\bar\alpha := \frac{(1-\eta_f)\epsilon_2}{C_f M_d^2}$ that

$$\begin{aligned} & f(x_{k_i} + \alpha d_{k_i}) - f(x_{k_i}) - m_{k_i}(\alpha) \\ \overset{(27b)}{\le} \; & C_f \alpha^2 \|d_{k_i}\|^2 \le \alpha(1-\eta_f)\epsilon_2 \\ \le \; & -(1-\eta_f) m_{k_i}(\alpha), \end{aligned}$$

which implies (36). □

Let us again first consider the "easy" case, in which the filter is augmented only a finite number of times.

Lemma 8. *Suppose that Assumptions* G *hold and that the filter is augmented only a finite number of times, i.e., $|\mathcal{A}| < \infty$. Then*

$$\lim_{k\to\infty} \chi(x_k) = 0.$$

Proof. Since $|\mathcal{A}| < \infty$, there exists $K \in \mathbb{N}$ so that $k \notin \mathcal{A}$ for all $k \ge K$. Suppose the claim is not true; i.e., there exist a subsequence $\{x_{k_i}\}$ and a constant $\epsilon > 0$ so that $\chi(x_{k_i}) \ge \epsilon$ for all i. From (30) and Lemma 2 there exist $\epsilon_1, \epsilon_2 > 0$ and $\tilde K \ge K$ so that for all $k_i \ge \tilde K$ we have $\theta(x_{k_i}) \le \epsilon_1$ and

$$m_{k_i}(\alpha) \le -\alpha\epsilon_2 \quad \text{for all} \quad \alpha \in (0,1]. \tag{37}$$

It then follows from (11) that for $k_i \ge \tilde K$

$$f(x_{k_i+1}) - f(x_{k_i}) \le \eta_f m_{k_i}(\alpha_{k_i}) \le -\alpha_{k_i}\eta_f\epsilon_2.$$

Reasoning as in the proof of Lemma 5, one can conclude that $\lim_i \alpha_{k_i} = 0$, since $f(x_{k_i})$ is bounded below and since $f(x_k)$ is monotonically decreasing (from (31)) for all $k \ge \tilde K$. We can now assume without loss of generality that $\tilde K$ is sufficiently large so that $\alpha_{k_i} < 1$. This means that for $k_i \ge \tilde K$ the first trial step $\alpha_{k,0} = 1$ has not been accepted. The last rejected trial step size

$$\alpha_{k_i,l_i} \in [\alpha_{k_i}/\tau_2, \alpha_{k_i}/\tau_1] \tag{38}$$

during the backtracking line search procedure then satisfies (9) since $k_i \notin \mathcal{A}$ and $\alpha_{k_i,l_i} > \alpha_{k_i}$. Thus, it must have been rejected because it violates (11); i.e., it satisfies

$$f(x_{k_i} + \alpha_{k_i,l_i} d_{k_i}) - f(x_{k_i}) > \eta_f m_{k_i}(\alpha_{k_i,l_i}), \tag{39}$$

or it has been rejected because it is not acceptable to the current filter, i.e.,

$$(\theta(x_{k_i} + \alpha_{k_i,l_i} d_{k_i}), f(x_{k_i} + \alpha_{k_i,l_i} d_{k_i})) \in \mathcal{F}_{k_i} = \mathcal{F}_K. \tag{40}$$

We conclude the proof by showing that neither (39) nor (40) can be true for sufficiently large k_i.

Consider (39): Since $\lim_i \alpha_{k_i} = 0$, we also have $\lim_i \alpha_{k_i,l_i} = 0$ (see (38)). In particular, for sufficiently large k_i we have $\alpha_{k_i,l_i} \leq \bar{\alpha}$ with $\bar{\alpha}$ from Lemma 7; i.e., (39) cannot be satisfied for those k_i.

Consider (40): Let $\Theta_K := \min\{\theta : (\theta, f) \in \mathcal{F}_K\}$. From Lemma 4 we have $\Theta_K > 0$. Using Lemmas 1 and 3, we then see that

$$\theta(x_{k_i} + \alpha_{k_i,l_i} d_{k_i}) \leq (1 - \alpha_{k_i,l_i})\theta(x_{k_i}) + C_\theta M_d^2 [\alpha_{k_i,l_i}]^2.$$

Since $\lim_i \alpha_{k_i,l_i} = 0$ and from Theorem 1 also $\lim_i \theta(x_{k_i}) = 0$, it follows that for k_i sufficiently large we have $\theta(x_{k_i} + \alpha_{k_i,l_i} d_{k_i}) < \Theta_K$, which contradicts (40). □

The next lemma establishes conditions under which a step size can be found that is acceptable to the current filter (see (12)).

LEMMA 9. *Suppose Assumptions* G *hold. Let* $\{x_{k_i}\}$ *be a subsequence with* $k_i \notin \mathcal{R}_{\text{inc}}$ *and* $m_{k_i}(\alpha) \leq -\alpha\epsilon_2$ *for a constant* $\epsilon_2 > 0$ *independent of* k_i *and for all* $\alpha \in (0, 1]$. *Then there exist constants* $c_5, c_6 > 0$ *so that*

$$(\theta(x_{k_i} + \alpha d_{k_i}), f(x_{k_i} + \alpha d_{k_i})) \notin \mathcal{F}_{k_i}$$

for all k_i *and* $\alpha \leq \min\{c_5, c_6\theta(x_{k_i})\}$.

Proof. Let M_d, C_θ, and C_f be the constants from Lemmas 1 and 3. Define $c_5 := \min\{1, \epsilon_2/(M_d^2\, C_f)\}$ and $c_6 := 1/(M_d^2\, C_\theta)$.

Now choose an iterate x_{k_i}. The mechanisms of Algorithm I ensure (see comment in step 6) that

$$(\theta(x_{k_i}), f(x_{k_i})) \notin \mathcal{F}_{k_i}. \tag{41}$$

For $\alpha \leq c_5$ we have $\alpha^2 \leq \frac{\alpha\epsilon_2}{M_d^2\, C_f} \leq \frac{-m_{k_i}(\alpha)}{C_f \|d_{k_i}\|^2}$ or, equivalently,

$$m_{k_i}(\alpha) + C_f \alpha^2 \|d_{k_i}\|^2 \leq 0,$$

and it follows with (27b) that

$$f(x_{k_i} + \alpha d_{k_i}) \leq f(x_{k_i}). \tag{42}$$

Similarly, for $\alpha \leq c_6\theta(x_{k_i}) \leq \frac{\theta(x_{k_i})}{\|d_{k_i}\|^2\, C_\theta}$, we have $-\alpha\theta(x_{k_i}) + C_\theta \alpha^2 \|d_{k_i}\|^2 \leq 0$ and thus from (27a)

$$\theta(x_{k_i} + \alpha d_{k_i}) \leq \theta(x_{k_i}). \tag{43}$$

The claim then follows from (41), (42), and (43) using (19). □

The last lemma in this section shows that in iterations corresponding to a subsequence with only nonoptimal limit points the filter is eventually not augmented. This result is used in the proof of the main global convergence theorem to yield a contradiction.

LEMMA 10. *Suppose Assumptions* G *hold. Let* $\{x_{k_i}\}$ *be a subsequence with* $\chi(x_{k_i}) \geq \epsilon$ *for a constant* $\epsilon > 0$ *independent of* k_i. *Then there exists* $K \in \mathbb{N}$ *so that for all* $k_i \geq K$ *the filter is not augmented in iteration* k_i*; i.e.,* $k_i \notin \mathcal{A}$.

Proof. Since by Theorem 1 we have $\lim_i \theta(x_{k_i}) = 0$, it follows from Lemma 2 that there exist constants $\epsilon_1, \epsilon_2 > 0$ so that

$$\theta(x_{k_i}) \leq \epsilon_1 \quad \text{and} \quad m_{k_i}(\alpha) \leq -\alpha\epsilon_2 \tag{44}$$

for k_i sufficiently large and $\alpha \in (0, 1]$; without loss of generality we can assume that (44) is valid for all k_i. We can now apply Lemmas 7 and 9 to obtain the constants $\bar{\alpha}, c_5, c_6 > 0$. Choose $K \in \mathbb{N}$ so that for all $k_i \geq K$

$$\theta(x_{k_i}) < \min\left\{\theta_{\text{inc}}, \frac{\bar{\alpha}}{c_6}, \frac{c_5}{c_6}, \left[\frac{\tau_1 c_6 \epsilon_2^{s_f}}{\delta}\right]^{\frac{1}{s_\theta - 1}}\right\} \tag{45}$$

with τ_1 from step 4.5. For all $k_i \geq K$ with $\theta(x_{k_i}) = 0$ we can argue as in the proof of Lemma 4 that both (9) and (11) hold in iteration k_i so that $k_i \notin \mathcal{A}$.

For the remaining iterations $k_i \geq K$ with $\theta(x_{k_i}) > 0$ we note that (45) implies that $k_i \notin \mathcal{R}_{\text{inc}}$,

$$\frac{\delta\,[\theta(x_{k_i})]^{s_\theta}}{\epsilon_2^{s_f}} < \tau_1 c_6 \theta(x_{k_i}) \tag{46}$$

(since $s_\theta > 1$), as well as

$$c_6\theta(x_{k_i}) < \min\{\bar{\alpha}, c_5\}. \tag{47}$$

Now choose an arbitrary $k_i \geq K$ with $\theta(x_{k_i}) > 0$ and define

$$\beta_{k_i} := c_6\theta(x_{k_i}) \overset{(47)}{=} \min\{\bar{\alpha}, c_5, c_6\theta(x_{k_i})\}. \tag{48}$$

Lemmas 7 and 9 then imply that a trial step size $\alpha_{k_i,l} \leq \beta_{k_i}$ satisfies both

$$f(x_{k_i}(\alpha_{k_i,l})) \leq f(x_{k_i}) + \eta_f m_{k_i}(\alpha_{k_i,l}) \tag{49}$$

and

$$\big(\theta(x_{k_i}(\alpha_{k_i,l})), f(x_{k_i}(\alpha_{k_i,l}))\big) \notin \mathcal{F}_{k_i}. \tag{50}$$

If we now denote with $\alpha_{k_i,L}$ the first trial step size satisfying both (49) and (50), the backtracking line search procedure in step 4.5 then implies that for $\alpha \geq \alpha_{k_i,L}$

$$\alpha \geq \tau_1\beta_{k_i} \overset{(48)}{=} \tau_1 c_6\theta(x_{k_i}) \overset{(46)}{>} \frac{\delta[\theta(x_{k_i})]^{s_\theta}}{\epsilon_2^{s_f}}$$

and therefore for $\alpha \geq \alpha_{k_i,L}$

$$\delta[\theta(x_{k_i})]^{s_\theta} < \alpha\epsilon_2^{s_f} = [\alpha]^{1-s_f}\,(\alpha\epsilon_2)^{s_f} \overset{(44)}{\leq} [\alpha]^{1-s_f}\,[-m_{k_i}(\alpha)]^{s_f}\,.$$

This means that $\alpha_{k_i,L}$ and all previous trial step sizes are f-step sizes. Consequently, for all trial step sizes $\alpha_{k_i,l} \geq \alpha_{k_i,L}$, Case I is considered in step 4.4, and by definition we have $\alpha_{k_i,L} \geq \alpha_{k_i}^{\min}$ (see discussion around (16)). Hence, the method does not switch to the feasibility restoration phase in step 4.2 for those trial step sizes. Therefore, $\alpha_{k_i,L}$ is indeed the accepted step size α_{k_i}. Since it satisfies both (9) and (49), the filter is not augmented in iteration k_i. $\square$

We are now ready to prove the main global convergence result.

THEOREM 2. *Suppose Assumptions* G *hold. Then*

$$\lim_{k\to\infty} \theta(x_k) = 0 \tag{51a}$$

and

$$\liminf_{k\to\infty} \chi(x_k) = 0. \tag{51b}$$

In other words, all limit points are feasible, and if $\{x_k\}$ is bounded, then there exists a limit point x_* of $\{x_k\}$ which is a first order optimal point for the equality constrained NLP (1).

Proof. Equation (51a) follows from Theorem 1. In order to show (51b), we consider two cases:

(i) The filter is augmented only a finite number of times. Then Lemma 8 proves the claim.

(ii) There exists a subsequence $\{x_{k_i}\}$ so that $k_i \in \mathcal{A}$ for all i. Now suppose that $\limsup_i \chi(x_{k_i}) > 0$. Then there exist a subsequence $\{x_{k_{i_j}}\}$ of $\{x_{k_i}\}$ and a constant $\epsilon > 0$ so that $\lim_j \theta(x_{k_{i_j}}) = 0$ and $\chi(x_{k_{i_j}}) > \epsilon$ for all k_{i_j}. Applying Lemma 10 to $\{x_{k_{i_j}}\}$, we see that there is an iteration k_{i_j}, in which the filter is not augmented; i.e., $k_{i_j} \notin \mathcal{A}$. This contradicts the choice of $\{x_{k_i}\}$ so that $\lim_i \chi(x_{k_i}) = 0$, which proves (51b). $\square$

Remark 7. We do not think that it is possible to obtain a stronger result in Theorem 2 under Assumptions G, such as "$\lim_k \chi(x_k) = 0$." The reason for this is that arbitrarily close to a strict local solution the restoration phase might be invoked even though the search direction is very good. This can happen if the current filter contains information corresponding to previous iterates that lie in a different region of $\mathbb{R}^n$ but has values for θ and f similar to those for the current iterate. For example, if for the current iterate the pair $(\theta(x_k), f(x_k))$ is very close to the current filter (e.g., there exist filter pairs $(\bar{\theta}, \bar{f}) \in \mathcal{F}_k$ with $\bar{\theta} < \theta(x_k)$ and $\bar{f} \approx f(x_k)$) and the objective function f has to be increased in order to approach the optimal solution, then the trial step sizes can be repeatedly rejected in step 4.3. In this case, $\alpha_{k,l}$ finally becomes smaller than $\alpha_k^{\min}$ and the restoration phase is triggered. Without making additional assumptions on the restoration phase we know only that the next iterate x_{k+1} returned from the restoration phase is acceptable to the augmented filter but possibly far away from any KKT point. We believe that it is not possible under the current assumptions to exclude the chance that this situation occurs repeatedly, in which case "$\lim_k \chi(x_k) = 0$" would not be valid.

Remark 8. It is possible to strengthen the convergence result under stronger assumptions. In addition to Assumptions G, suppose that x_* is a local solution of the NLP (1) satisfying the second order sufficient optimality conditions [17] with optimal multipliers λ_*. Let us further assume that the line search filter method generates multiplier iterates λ_k based on the linearization (3) by choosing $\lambda_{k+1} = \lambda_k + \alpha_k d_k^\lambda$ with

$$d_k^\lambda := \lambda_k^+ - \lambda_k \tag{52}$$

in each iteration $k \notin \mathcal{R}_{\text{inc}}$. Also, assume that close to x_* the algorithm uses exact second derivatives; i.e., $H_k = \nabla^2 f(x_k) + \sum_i \lambda_k^{(i)} \nabla^2 c(x_k)$. Finally, suppose that, in the neighborhood of (x_*, λ_*), also the algorithm used for the restoration phase is taking steps (d_k, d_k^λ) generated from (3) and (52), where H_k is bounded and satisfies (20). Then, once the iterates (x_k, λ_k) are sufficiently close to (x_*, λ_*), the overall algorithm always takes fractions of steps (d_k, d_k^λ). The assumptions ensure that the KKT error (the norm of the left-hand side of (2)) is monotonically decreased and that the iterates are attracted to (x_*, λ_*). As a second order sufficient optimal solution, (x_*, λ_*) is the only root of (2) in a sufficiently small neighborhood. Therefore we obtain together with Theorem 2 that the entire sequence converges to the solution, once the iterates are sufficiently close.

One way to construct a restoration phase that satisfies the condition necessary for this result is as follows. Suppose that we have a "rigorous" Algorithm R for the restoration phase, which either converges to a stationary point of the constraint violation or produces an acceptable new iterate for the filter method. If the restoration phase is now invoked at a point where the KKT error is small, then, instead of directly using Algorithm R, we first compute a search direction (d_k, d_k^λ) from (3) and (52). If the new (intermediate) iterate obtained by taking the full step does not reduce the KKT error by a fixed fraction $\kappa_R \in (0,1)$, we switch to Algorithm R. Otherwise we continue taking steps from (3) and (52) (still formally within the restoration phase in step 8), until finally either a new acceptable iterate x_{k+1} is obtained, or the method reverts to Algorithm R.

4. Alternative algorithms.

4.1. Measures based on the augmented Lagrangian function. The two measures $f(x)$ and $\theta(x)$ can be considered as the two components of the exact penalty function (7). Another popular choice for a merit function is the *augmented Lagrangian function* (see, e.g., [2, 5, 18])

$$\ell_\rho(x, \lambda) := f(x) + \lambda^T c(x) + \frac{\rho}{2} c(x)^T c(x), \tag{53}$$

where λ are multiplier estimates corresponding to the equality constraints (1b). If λ_* is the vector of multipliers corresponding to a strict local solution x_* of the NLP (1), then there exists a penalty parameter $\rho > 0$ so that x_* is a strict local minimizer of $\ell_\rho(x, \lambda_*)$.

In the line search filter method described in section 2 we can alternatively follow an approach based on the two components $\mathcal{L}(x, \lambda)$ (defined in (4)) and $\theta(x)$ (or, equivalently, $\theta(x)^2$) of the augmented Lagrangian function rather than on the components of the exact penalty function. (Recently, S. Ulbrich [22] proposed a related approach using the Lagrangian function in a trust region filter method, including both global and local convergence results.) In Algorithm I we then replace all occurrences of the measure "$f(x)$" by "$\mathcal{L}(x, \lambda)$." In addition to the iterates x_k we now also keep iterates λ_k as estimates of the equality constraint multipliers, and compute in each iteration k a search direction d_k^λ for those variables. This search direction can be obtained with no additional computational cost from (52) with λ_k^+ from (3). Defining

$$\lambda_k(\alpha_{k,l}) := \lambda_k + \alpha_{k,l} d_k^\lambda, \tag{54}$$

the sufficient reduction criteria (8b) and (11) are then replaced by

$$\mathcal{L}(x_k(\alpha_{k,l}), \lambda_k(\alpha_{k,l})) \le \mathcal{L}(x_k, \lambda_k) - \gamma_f \theta(x_k) \tag{55a}$$

and

$$\mathcal{L}(x_k(\alpha_{k,l}), \lambda_k(\alpha_{k,l})) \le \mathcal{L}(x_k, \lambda_k) + \eta_f m_k(\alpha_{k,l}), \tag{55b}$$

respectively, where the model m_k for $\mathcal{L}$ is now defined as

$$m_k(\alpha) := \alpha g_k^T d_k - \alpha \lambda_k^T c_k + \alpha(1-\alpha) c_k^T d_k^\lambda, \tag{56}$$

which is obtained by Taylor expansions of $f(x)$ and $c(x)$ around (x_k, λ_k) into direction (d_k, d_k^λ) and the use of (3).

The f-type switching condition (9) remains unchanged, but the definition of the minimum step size (18) has to be modified to accommodate (55) and (56). The only requirements for this change are again that it is guaranteed that the method does not switch to the feasibility restoration phase in step 4.2 as long as the f-type switching condition (9) is satisfied for a trial step size $\alpha \le \alpha_{k,l}$, and that the backtracking line search in step 4 is finite. We also require that the multipliers λ_{k+1} that are used after the restoration phase has been called are uniformly bounded (e.g., by choosing $\lambda_{k+1} = \lambda_k$ for $k \in \mathcal{R}$).

In order to see that the global convergence analysis in section 3 still holds, let us briefly revisit the individual results. The first two bounds in Lemma 1 remain valid, so that with

$$\lambda_{k+1} \overset{(54)}{=} \lambda_k + \alpha_k d_k^\lambda \overset{(52)}{=} (1-\alpha_k)\lambda_k + \alpha_k \lambda_k^+$$

we obtain by induction that λ_k, and therefore also d_k^λ are uniformly bounded for all $k \notin \mathcal{R}_{\mathrm{inc}}$. With this, also the last bound in (25) holds, as can be seen from (56). Since λ_k is bounded for all k, we further see that the sequence $\{\mathcal{L}(x_k, \lambda_k)\}$ is bounded below, a property used at several points in the analysis. It is then easy to verify that Lemmas 2 and 4 are still valid for the model definition (56), since the first equality in (26a) then becomes

$$m_{k_i}(\alpha)/\alpha = g_{k_i}^T d_{k_i} + O(\|c_{k_i}\|),$$

and thus only the constant c_3 in the proof may change. Furthermore, Lemma 3 still holds for the model definition (56) and with the measure "f" replaced by "$\mathcal{L}$," because

$$\begin{aligned}
& \mathcal{L}(x_k + \alpha d_k, \lambda_k + \alpha d_k^\lambda) - \mathcal{L}(x_k, \lambda_k) \\
= \ & f(x_k + \alpha d_k) - f(x_k) + (\lambda_k + \alpha d_k^\lambda)^T c(x_k + \alpha d_k) - \lambda_k^T c(x_k) \\
= \ & \alpha g_k^T d_k + O(\alpha^2 \|d_k\|^2) + (\lambda_k + \alpha d_k^\lambda)^T (c(x_k) + \alpha A_k^T d_k + O(\alpha^2 \|d_k\|^2)) - \lambda_k^T c(x_k) \\
\overset{(3)}{=} \ & \alpha g_k^T d_k + (\lambda_k + \alpha d_k^\lambda)^T (1-\alpha) c(x_k) - \lambda_k^T c(x_k) + O(\alpha^2 \|d_k\|^2) \\
\overset{(56)}{=} \ & m_k(\alpha) + O(\alpha^2 \|d_k\|).
\end{aligned}$$

Finally, the analysis in sections 3.3 and 3.4 then holds with replacing "f" by "$\mathcal{L}$" where appropriate. The only point that deserves special attention is the proof of Lemma 8. Here, it is essential that the last rejected trial step size (38) satisfies the f-type switching condition (9), at least for k_i sufficiently large. To see that this is also true for the model definition (56), which is no longer linear in α, let us define the function

$$h_{k_i}(\alpha) := [-m_{k_i}(\alpha)]^{s_f} \ \alpha^{1-s_f} - \delta[\theta(x_{k_i})]^{s_\theta}.$$

This function is well defined for the considered k_i due to (37), and we have $h_{k_i}(\alpha_{k,l}) > 0$ if and only if (9) holds. Since we assume $\lim_i \theta(x_{k_i}) = 0$ and $\chi(x_{k_i}) \geq \epsilon$ in the proof, it can then be shown (using arguments similar to those in the proof of Lemma 2) that $h'_{k_i}(0) \geq \epsilon_3$ for some $\epsilon_3 > 0$ and k_i sufficiently large and that $h''_{k_i}(0)$ is uniformly bounded. Since $\alpha_{k_i} \to 0$ and $h_{k_i}(\alpha_{k_i}) > 0$, it then follows that the f-type switching condition (9) holds for $\alpha_{k_i,l_i} \in [\alpha_{k_i}/\tau_2, \alpha_{k_i}/\tau_1]$ when k_i is sufficiently large.

4.2. Line search filter SQP methods. In this section we show how Algorithm I can be applied to line search SQP methods for the solution of nonlinear optimization problems of the form

$$\min_{x\in\mathbb{R}^n} \quad f(x) \tag{57a}$$

$$\text{subject to} \quad c(x) = 0, \tag{57b}$$

$$x \geq 0. \tag{57c}$$

We choose to consider only bounds of the form (57c) to simplify the presentation, but our discussion can easily be adapted to general bound constraints (such as "$x_L \leq x \leq x_U$").

At an iterate x_k, a line search SQP method obtains the search direction d_k as a solution of the quadratic program (QP)

$$\min_{d\in\mathbb{R}^n} \quad g_k^T d + \frac{1}{2} d^T H_k d \tag{58a}$$

$$\text{subject to} \quad A_k^T d + c(x_k) = 0, \tag{58b}$$

$$x_k + d \geq 0. \tag{58c}$$

As before, $g_k := \nabla f(x_k)$, $A_k := \nabla c(x_k)$. Furthermore, now H_k denotes a symmetric matrix approximating the Hessian of the Lagrangian

$$\mathcal{L}(x, \lambda, z) := f(x) + \lambda^T c(x) - v^T x \tag{59}$$

of the NLP (57). The vector $v \geq 0$ stands for the Lagrangian multipliers corresponding to the bound constraints (57c). We denote by λ_k^+ and $v_k^+ \geq 0$ some (not necessarily unique) multipliers corresponding to the QP solution d_k.

In the following analysis, we assume that the particular method for solving (58) is able to ensure that the QP Hessian H_k is positive definite in a certain space (see Assumption (G3*) below), possibly by modifying the matrix H_k. Then a (finite) solution of the QP exists, and the generated search direction d_k is a direction of sufficient decrease for the objective function if the constraint violation is small.

Starting from an initial point $x_0 \geq 0$, Algorithm I can then be used with the following modification:

- The computation of the search direction in step 3 is replaced by the solution of the QP (58).
- The restoration phase is invoked from step 3 if the QP (58) is unbounded, infeasible, or "not sufficiently consistent" (see Assumption (G4*) below). As before, $\mathcal{R}_{\text{inc}}$ denotes the set of all iteration counters in which the restoration phase is invoked from step 3.
- The feasibility restoration phase in step 8 has to return an iterate x_{k+1} that satisfies the bound constraints (57c).

In order to state the assumptions necessary for a global convergence analysis let us define for each $k \notin \mathcal{R}_{\text{inc}}$ the set of coordinates that are active at the current point x_k *and* at $x_k + d_k$,

$$\mathcal{S}_k := \left\{ i \in \{1, \ldots, n\} : x_k^{(i)} = 0 \quad \text{and} \quad d_k^{(i)} = 0 \right\}.$$

For the purpose of the analysis we again consider a decomposition of the search direction

$$d_k = q_k + p_k, \tag{60}$$

where q_k is now defined as the solution of the QP

$$\begin{align} & \min_{q \in \mathbb{R}^n} \quad q^T q & & \tag{61a} \\ & \text{subject to} \quad A_k^T q + c_k = 0, & & \tag{61b} \\ & \qquad q^{(i)} = 0 & & \text{for } i \in \mathcal{S}_k, \tag{61c} \\ & \qquad x_k^{(i)} + q^{(i)} \geq 0 & & \text{for } i \notin \mathcal{S}_k. \tag{61d} \end{align}$$

Therefore, q_k is the shortest vector that satisfies the constraints in the QP (58) and stays at those bounds that are active for all trial points (6). We further choose Z_k as an orthonormal null space matrix for the matrix

$$\left[\begin{array}{cccc} A_k & e_{j_1} & \cdots & e_{j_{l_k}} \end{array} \right]^T, \qquad \text{where} \qquad \mathcal{S}_k = \{j_1, \ldots, j_{l_k}\};$$

i.e., Z_k is a basis of the null space for the gradients of all equality constraints and bounds that are active at x_k and $x_k + d_k$. With this, we can compute $p_k = Z_k \bar{p}_k$ with $\bar{p}_k$ as the solution of the reduced QP (see, e.g., [17])

$$\begin{align} & \min_{\bar{p} \in \mathbb{R}^{n-m-l_k}} \quad \left(Z_k^T g_k + Z_k^T H_k q_k \right)^T \bar{p} + \frac{1}{2} \bar{p}^T Z_k^T H_k Z_k \bar{p} \tag{62a} \\ & \text{subject to} \quad x_k + q_k + Z_k \bar{p} \geq 0. \tag{62b} \end{align}$$

Note that the set $\mathcal{S}_k$ is not known before the QP (58) has been solved. The QPs (61) and (62) are defined only to state the assumptions below and are not a possible procedure to obtain the search direction d_k.

For the global convergence analysis of the filter line search SQP method we replace Assumptions (G3) and (G4) by the following:

(G3*) *There exists a constant* $M_H > 0$ *so that for all* $k \notin \mathcal{R}_{\text{inc}}$ *we have*

$$\lambda_{\min} \left(Z_k^T H_k Z_k \right) \geq M_H, \tag{63}$$

where Z_k *has been defined above.*

(G4*) *There exist constants* $M_q, M_\lambda, M_v > 0$ *so that for all* $k \notin \mathcal{R}_{\text{inc}}$ *we have*

$$\|q_k\| \leq M_q \theta(x_k), \qquad \|\lambda_k^+\| \leq M_\lambda, \qquad \|v_k^+\| \leq M_v.$$

As Assumption (G3) for the original analysis, Assumption (G3*) is necessary to ensure descent in the objective function at points with small infeasibility. In order to ensure this condition, the algorithm could monitor the eigenvalues of the projection of H_k onto the null space of the gradients for all constraints active at x_k and $x_k + d_k$, and perform modifications of H_k if necessary.[1] Note that Assumption (G3*) is natural in the sense that if the method converges to a local solution x_* of the NLP (57) satisfying the strong second order optimality conditions, then the active set $\mathcal{S}_k$ finally becomes unchanged and Z_k is a null space matrix for the constraints active at x_*. Hence, no correction of the reduced Hessian is necessary close to x_* if exact second derivatives are used and if λ_k converges to λ_*.

Assumption (G4*) is similar in spirit to the assumption expressed by (2.10) for the trust region filter SQP method in [7]. Essentially, it implies that eventually the restoration phase is triggered from step 3 if the constraints of the QP (58) are becoming increasingly degenerate close to feasible points.

It is straightforward to verify that the proofs in section 3 still hold under the modified Assumptions G. Only the proof of Lemma 2 requires special attention. Let us first state the KKT conditions of the reduced QP (62), which have to be satisfied by the solution d_k and the corresponding multipliers,

$$Z_k^T H_k Z_k \bar{p}_k + (Z_k^T g_k + Z_k^T H_k q_k) - Z_k^T v_k^+ = 0, \tag{64a}$$

$$x_k + q_k + Z_k \bar{p}_k \geq 0, \tag{64b}$$

$$(x_k + q_k + Z_k \bar{p}_k)^T v_k^+ = 0, \tag{64c}$$

$$v_k^+ \geq 0. \tag{64d}$$

For $k \notin \mathcal{R}_{\text{inc}}$ we then have

$$\begin{aligned} Z_k^T g_k &\overset{(64a)}{=} Z_k^T v_k^+ - Z_k^T H_k Z_k \bar{p}_k - Z_k^T H_k q_k, \\ (x_k + q_k)^T v_k^+ &\overset{(64c)}{=} -(v_k^+)^T Z_k \bar{p}_k \end{aligned}$$

and therefore

$$\begin{aligned} g_k^T Z_k \bar{p}_k &= -(x_k + q_k)^T v_k^+ - \bar{p}_k^T Z_k^T H_k Z_k \bar{p}_k - \bar{p}_k^T Z_k^T H_k q_k \\ &\overset{(61c),(61d),(64d)}{\leq} -\bar{p}_k^T Z_k^T H_k Z_k \bar{p}_k - \bar{p}_k^T Z_k^T H_k q_k. \end{aligned}$$

This gives, together with the modified Assumptions G,

$$\begin{aligned} m_k(\alpha)/\alpha \overset{(10)}{=} g_k^T d_k &\overset{(60)}{=} g_k^T Z_k \bar{p}_k + g_k^T q_k \\ &\leq -\bar{p}_k^T Z_k^T H_k Z_k \bar{p}_k - \bar{p}_k^T Z_k^T H_k q_k + g_k^T q_k \\ &\leq -M_H [\chi(x_k)]^2 + O\left(\chi(x_k)\theta(x_k)\right) + O(\theta(x_k)), \end{aligned}$$

which corresponds to (26c). We can conclude the proof of Lemma 2 as before.

4.3. Line search filter interior point methods. An alternative to active set methods for handling inequality constraints is offered by *interior point* or *barrier methods*. In this section we demonstrate how the line search filter method presented in section 2 can be used within an interior point framework. The presented algorithm can be changed in an obvious way if (57c) is generalized to lower and upper bound constraints on all or only some variables.

[1] The solution d_k is not known before the QP (58) is solved. One possible way to find a suitable modification of H_k is to solve (58) repeatedly with $H_k = \nabla^2_{xx}\mathcal{L}(x_k, \lambda_k) + \xi I$ for an increasing sequence of modifications $\xi \geq 0$, until the QP is not unbounded and H_k has the required convexity properties expressed in (63).

The barrier method presented here can be of the primal or primal-dual type, and differs from the interior point filter algorithm proposed by M. Ulbrich, S. Ulbrich, and Vicente [21] in that the barrier parameter is kept constant for several iterations. This enables us to base the acceptance of trial steps directly on the (barrier) objective function instead of only on the norm of the optimality conditions. Therefore the presented method can be expected to be less likely to converge to saddle points or maxima than the algorithm proposed in [21].

A barrier method solves a sequence of *barrier problems*

$$\min_{x\in\mathbb{R}^n} \quad \varphi_\mu(x) := f(x) - \mu \sum_{i=1}^{n} \ln(x^{(i)}) \tag{65a}$$

$$\text{subject to} \quad c(x) = 0 \tag{65b}$$

for a decreasing sequence μ_l of *barrier parameters* with $\lim_l \mu_l = 0$. Local convergence of barrier methods as $\mu \to 0$ has been discussed in detail by other authors, in particular by Nash and Sofer [16] for primal methods, and by Byrd, Liu, and Nocedal [4] and Gould et al. [12, 13] for primal-dual methods. In those approaches, the barrier problem (65) is solved to a certain tolerance $\epsilon > 0$ for a fixed value of the barrier parameter μ. The parameter μ is then decreased, and the tolerance ϵ is tightened for the next barrier problem. For example, in [12] it is shown that if the parameters μ and ϵ are updated in a particular fashion, the new starting point (enhanced by an extrapolation step with the cost of one regular iteration that tries to follow the path defined by the optimality conditions for (65) as μ changes) eventually solves the next barrier problem well enough to satisfy the new tolerance. Then the barrier parameter μ is decreased again immediately (without taking an additional step), leading to a superlinear convergence rate of the overall interior point algorithm for solving the original NLP (57).

Consequently, the step acceptance criterion in the solution procedure for a fixed barrier parameter μ becomes irrelevant as soon as the (extrapolated) starting points are immediately accepted. Until then, we can consider the (approximate) solution of the individual barrier problems as independent procedures, similar to the approach taken by Byrd et al. in [3]. The focus of this paper is the properties of the line search filter approach, and we therefore address only the convergence properties of an algorithm for solving the barrier problem (65) for a *fixed* value of the barrier parameter μ. Some additional comments on the overall interior point method are given in Remark 9 at the end of this section.

The main idea is to apply the technique developed and analyzed in sections 2 and 3 to solve the barrier problem (65); i.e., we replace all occurrences of "f" by "φ_μ." However, there are two issues that we have to consider:

1. The barrier objective function (65a) is defined only as long as all components of x are strictly within bounds, i.e., $x > 0$;
2. The barrier objective function and its derivatives become unbounded if any of the components of x approaches its bound.

In order to handle the first item, we enforce that all iterates x_k are strictly positive. For this purpose, we assume that the starting point satisfies $x_0 > 0$ and that an iterate returned from the restoration phase satisfies $x_{k+1} > 0$. We further define a maximal step size $\alpha_k^{\max} \in (0, 1]$ using the *fraction-to-the-boundary rule*

$$\alpha_k^{\max} := \max\left\{\alpha \in (0,1] : x_k + \alpha d_k \geq (1-\tau)x_k\right\} \tag{66}$$

for a fixed parameter $\tau \in (0,1)$, usually chosen close to 1. With this, we start the backtracking line search in step 4.1 of Algorithm I from $\alpha_{k,0} = \alpha_k^{\max}$. Then all trial points (6) lie strictly within bounds.

Addressing the second item, we show below that under additional assumptions the iterates generated by the modified Algorithm I (using (66)) are bounded away from the bounds (see Theorem 3), so that in turn the appropriate quantities in analysis of section 3 are bounded. Here it is necessary to assume that the parameter s_f in the f-type switching condition (9) is chosen to be 1 (see Remark 6).

For later reference, let us restate the linear system (3) in the notation of this section. Recalling that "f" is replaced by "φ_μ," this system can be written as

$$\begin{bmatrix} W_k + \mu X_k^{-2} & A_k \\ A_k^T & 0 \end{bmatrix} \begin{pmatrix} d_k \\ \lambda_k^+ \end{pmatrix} = -\begin{pmatrix} \nabla f(x_k) - \mu X_k^{-1} e \\ c(x_k) \end{pmatrix}, \tag{67}$$

where $X_k := \mathrm{diag}(x_k)$, e is a vector of appropriate dimension of 1's, and W_k is (an approximation of) the Hessian of the Lagrangian (59) for the *original* NLP (57). Note that the Hessian H_k in (3) is equal to $W_k + \mu X_k^{-2}$. It is easy to verify that the following arguments also hold if the primal Hessian "μX_k^{-2}" of the log-barrier terms is replaced by the primal-dual Hessian "$\Sigma_k = X_k^{-1} V_k$" (with variables $v_k > 0$), as long as there exists $m_\Sigma > 1$ such that

$$\frac{1}{m_\Sigma}\mu \le v_k^{(i)} x_k^{(i)} \le m_\Sigma \mu$$

for all i and k.

Next we state the assumptions necessary to show global convergence for the barrier line search filter algorithm.

Assumptions B. *Given a starting point $x_0 > 0$, let $\{x_k\}$ be the sequence generated by Algorithm* I *(adapted to the solution of the barrier problem and with $s_f = 1$ in* (9)*), where we assume that the feasibility restoration phase in step* 8 *always terminates successfully with $x_{k+1} > 0$ and that the algorithm does not stop in step* 2 *at a KKT point.*

(B1) *There exists an open set $\mathcal{C} \subseteq \mathbb{R}^n$ with $[x_k, x_k + \alpha_k^{\max} d_k] \subseteq \mathcal{C}$ for all $k \notin \mathcal{R}_{\mathrm{inc}}$ so that f and c are differentiable on $\mathcal{C}$, and their function values, as well as their first derivatives, are bounded and Lipschitz-continuous over $\mathcal{C}$.*

(B2) *The matrices W_k approximating the Hessian of the Lagrangian of the* original *NLP* (57) *used in* (67) *are uniformly bounded for all $k \notin \mathcal{R}_{\mathrm{inc}}$.*

(B3) *The matrices $H_k = W_k + \mu X_k^{-2}$ are uniformly positive definite on the null space of the Jacobian A_k^T. In other words, there exists a constant $M_H > 0$ so that for all $k \notin \mathcal{R}_{\mathrm{inc}}$*

$$\lambda_{\min}\left(Z_k^T (W_k + \mu X_k^{-2}) Z_k\right) \ge M_H, \tag{68}$$

where the columns of $Z_k \in \mathbb{R}^{n \times (n-m)}$ form an orthonormal basis matrix of the null space of A_k^T.

(B4) *There exists a constant $M_A > 0$ so that for all $k \notin \mathcal{R}_{\mathrm{inc}}$ we have*

$$\sigma_{\min}(A_k) \ge M_A. \tag{69}$$

(B5) *The iterates for which the restoration phase is invoked from step* 3 *(for example, because* (68) *or* (69) *is violated) are not arbitrarily close to the feasible region. In other words, there exists a constant $\theta_{\mathrm{inc}} > 0$ so that $k \notin \mathcal{R}_{\mathrm{inc}}$ whenever $\theta(x_k) \le \theta_{\mathrm{inc}}$.*

(B6) *The iterates* $\{x_k\}$ *are bounded.*

(B7) *At all* feasible *limit points* $\bar{x}$ *of* $\{x_k\}$*, the gradients of the active constraints,*

$$\nabla c_1(\bar{x}), \ldots, \nabla c_m(\bar{x}), \qquad \textit{and} \qquad e_i \textit{ for } i \in \{j : \bar{x}^{(j)} = 0\}, \tag{70}$$

are linearly independent.

(B8) *There exist constants* $\tilde{\delta}_\theta, \tilde{\delta}_x > 0$ *so that whenever the restoration phase is called in step* 8 *in an iteration* $k \in \mathcal{R}$ *with* $\theta(x_k) \leq \tilde{\delta}_\theta$*, it returns a new iterate with* $x_{k+1}^{(i)} \geq x_k^{(i)}$ *for all components satisfying* $x_k^{(i)} \leq \tilde{\delta}_x$*.*

Assumptions (B1)–(B5) are essentially identical to the original Assumptions (G1)–(G5). Note, however, that the boundedness assumptions in (B1) and (B2) pertain to the original functions and not to those from the barrier problem (65), to which the line search filter algorithm is applied. Boundedness of the barrier function φ_μ cannot be assumed, as pointed out above. On the other hand, the lower bound (68) refers to the Hessian used for the step computation (67).

Assumption (B6) is necessary in order to guarantee that the barrier objective function $\varphi_\mu(x)$ is bounded below. Assumption (B7) implies Assumptions (B4) and (B5) if the strategy described at the end of section 3.1 is used to define when $k \in \mathcal{R}_{\text{inc}}$.

Assumption (B7) is considerably less restrictive than the regularity assumptions made for the global convergence analysis of the interior point methods proposed by El-Bakry et al. [6], Yamashita [28], and Yamashita, Yabe, and Tanabe [29]. For those algorithms, it is essentially required that the gradients of all equality constraints and active inequality constraints (70) are linearly independent at *all* limit points, and not only at all *feasible* limit points. If those methods are applied to the example presented by Wächter and Biegler in [25] (which satisfies Assumption (B7)), they converge to a spurious solution that is neither feasible nor a stationary point for any norm of the constraint violation; for details, see [25]. In contrast to this, the proposed algorithm is at least guaranteed to converge to a stationary point for the infeasibility (assuming that a reasonable restoration phase algorithm is used), and in practice converges to the solution [24]. We note here that the method presented by Tits et al. in [20] has a similar convergence guarantee as the proposed method, in the sense that the regularity assumption for the constraints in [20] excludes only infeasible limit points, at which there is no feasible descent direction for the constraint violation measured in the ℓ_1 norm.

To see that Assumption (B8) is reasonable, suppose that the gradients of the active constraints (70) are uniformly linearly independent at all feasible points $\bar{x}$ (this is similar to Assumption (B7)). By proof of contradiction one can then show that there exist constants $\tilde{\delta}_\theta, \tilde{\delta}_x > 0$ so that whenever $\theta(x_k) \leq \tilde{\delta}_\theta$ for $k \in \mathcal{R}$, then there exists a *feasible* point $\bar{x}_k$ with $\theta(\bar{x}_k) = 0$, $\bar{x}_k > 0$, and $\bar{x}_k^{(i)} \geq x_k^{(i)}$ for all i with $x_k^{(i)} \leq \tilde{\delta}_x$. The point $\bar{x}_k$ is a candidate for the point x_{k+1} returned from the restoration phase algorithm satisfying the condition in Assumption (B8).

To find an approximation to such a point $\bar{x}_k$, we may apply some algorithm for bound constraint optimization to the problem

$$\min \quad \|c(x)\|_2^2 + \rho\|x - x_k\|_2^2 \tag{71a}$$

$$\text{subject to} \quad x^{(i)} \geq \min\{\epsilon, x_k^{(i)}\} \quad \text{for } i = 1, \ldots, n. \tag{71b}$$

Here, the regularization term weighted by $\rho > 0$ aims to keep the solution of this problem in the neighborhood of x_k, and $\epsilon > 0$ is some small number that we introduce to make sure that the (approximate) solution for this problem is not arbitrarily close to the boundary. In order to find suitable values of ρ and ϵ one might start with some initial choices, and whenever the optimal solution of (71) does not reduce $\theta(x)$ sufficiently to be accepted in step 8 as x_{k+1}, problem (71) could be resolved with smaller values of ρ or ϵ; M. Ulbrich, S. Ulbrich, and Vicente [21] outline a related restoration phase procedure. From the discussion in the previous paragraph, it is clear that a careful implementation of such a procedure should eventually produce (approximate) solutions $\bar{x}_k$ of (71) for $k \in \mathcal{R}$ that satisfy Assumption (B8).[2]

While a detailed discussion of the restoration phase algorithm is beyond the scope of this paper, we propose in Wächter and Biegler [27] a procedure for the restoration phase which applies the interior point filter algorithm recursively to a problem formulation similar to (71) and seems to perform well in practice.

Finally we remark that Assumption (B3) is weaker than the one made in an earlier version of our analysis [24].

The remainder of this section deals with the proof of the following theorem.

Theorem 3. *Suppose Assumptions* B *hold. Then there exists a constant $\epsilon_x > 0$ so that $x_k \geq \epsilon_x e$ for all k.*

This means that the iterates generated by Algorithm I (for the barrier algorithm) are bounded away from the boundary of the region defined by the bound constraints (57c). Once this is established one can verify that then Assumptions B imply Assumptions G, and therefore the global convergence results from section 3 hold. We only point out that Theorem 3 and Lemma 1 together with (66) establish that the starting step size in the backtracking line search $\alpha_k^{\max}$ is uniformly bounded away from zero, a property necessary in the proofs of Lemmas 7, 8, and 9 (for details, see also [24]).

In order to prove Theorem 3 we make use of the following lemma.

Lemma 11. *Suppose Assumptions* B *hold. Then, for a given subset of indices $\mathcal{S} \subseteq \{1, \dots, n\}$ and a constant $\delta_l > 0$, there exist $\delta_s, \delta_\theta > 0$ so that $d_k^{(i)} > 0$ for $i \in \mathcal{S}$ whenever $k \notin \mathcal{R}$ and*

$$x_k \in L := \left\{ x \geq 0 : x^{(i)} \leq \delta_s \text{ for } i \in \mathcal{S}, x^{(i)} \geq \delta_l \text{ for } i \notin \mathcal{S}, \theta(x) \leq \delta_\theta \right\};$$

i.e., at sufficiently feasible points, the search direction points away from almost active bounds.

Proof. Let us denote with x_k^s the components of x_k in $\mathcal{S}$ and with x_k^l the remaining ones. Without loss of generality we assume $x_k = [(x_k^s)\ (x_k^l)]$; similarly, we define A_k^s, A_k^l, etc. First, we rewrite the linear system (67) by scaling the first rows and columns by X_k^s:

$$\text{(72)} \quad \begin{bmatrix} X_k^s W_k^{ss} X_k^s + \mu I & X_k^s W_k^{sl} & X_k^s A_k^s \\ W_k^{ls} X_k^s & W_k^{ll} + \mu (X_k^l)^{-2} & A_k^l \\ (A_k^s)^T X_k^s & (A_k^l)^T & 0 \end{bmatrix} \begin{pmatrix} \tilde{d}_k^s \\ d_k^l \\ \lambda_k^+ \end{pmatrix} = - \begin{pmatrix} X_k^s g_k^s - \mu e \\ g_k^l - \mu (X_k^l)^{-1} e \\ c(x_k) \end{pmatrix},$$

where we defined $\tilde{d}_k^s := (X_k^s)^{-1} d_k^s$.

For some initial choice of $\delta_s, \delta_\theta > 0$, let $\bar{x} \in L$ be a feasible point with $\bar{x}^s = 0$. We then have from Assumption (B7) that the columns of the matrix

$$\begin{bmatrix} [\nabla c(\bar{x})]^s & I \\ [\nabla c(\bar{x})]^l & 0 \end{bmatrix},$$

[2] Recall that we assume here that the restoration phase always terminates successfully. Otherwise, this procedure should produce a limit point that is a local minimizer, or at least a stationary point, for the constraint violation within the bound constraints $x \geq 0$.

and therefore also the columns of $[\nabla c(\bar{x})]^l$, are linearly independent. Using a compactness argument and Assumption (B6), we can find a constant $m_\sigma > 0$ so that $\sigma_{\min}([\nabla c(\bar{x})]^l) \geq m_\sigma$ for all feasible limit points $\bar{x} \in L$ of $\{x_k\}$ with $\bar{x}^s = 0$. Therefore, we have from Assumption (B1) that

$$\sigma_{\min}(A_k^l) \geq \frac{m_\sigma}{2}$$

for all $x_k \in L$ if δ_θ and δ_s are chosen sufficiently small.

In addition, possibly after further decreasing δ_θ, it follows from Assumptions (B3) and (B5) that for all $x_k \in L$ the projection of $W_k^{ll} + \mu(X_k^l)^{-2}$ into the null space of $(A_k^l)^T$ is uniformly positive definite.

Together with the boundedness assumptions (B1) and (B2), we then see that (72) satisfies

$$\left(\begin{bmatrix} \mu I & 0 & 0 \\ 0 & W_k^{ll} + \mu(X_k^l)^{-2} & A_k^l \\ 0 & (A_k^l)^T & 0 \end{bmatrix} + O(\delta_s) \right) \begin{pmatrix} \tilde{d}_k^s \\ d_k^l \\ \lambda_k^+ \end{pmatrix} = - \begin{pmatrix} -\mu e \\ g_k^l - \mu(X_k^l)^{-1} e \\ c(x_k) \end{pmatrix} + O(\delta_s),$$

for $x_k \in L$, where the inverse of the matrix in the square brackets, as well as the right-hand side, are uniformly bounded for δ_s sufficiently small. Therefore, for $x_k \in L$, we have that $\tilde{d}_k^s = e + O(\delta_s)$, and consequently $\tilde{d}_k^s > 0$, after possibly reducing δ_s even more. The claim then follows from $d_k^s = X_k^s \tilde{d}_k^s$. □

We finish with the proof of Theorem 3.

Proof of Theorem 3. We first show by contradiction that there exist constants $\delta_x, \delta_\theta > 0$ so that $d_k^{(i)} > 0$ for all indices i with $x_k^{(i)} \leq \delta_x$ whenever $\theta(x_k) \leq \delta_\theta$ and $k \notin \mathcal{R}$.

Suppose this claim is not true. Then, there exists a subsequence $\{x_{k_j}\}$ of iterates with $k_j \notin \mathcal{R}$, $\lim_j \theta(x_{k_j}) = 0$ and $\lim_j x_{k_j}^{(s)} = 0$ for some index s, as well as $d_{k_j}^{(s)} \leq 0$ for all j. Let $\bar{x}$ be a limit point of $\{x_{k_j}\}$, and define $\mathcal{S} := \{i : \bar{x}^{(i)} = 0\}$ and $\delta_l := \min\{\bar{x}^{(i)}/2 : i \notin \mathcal{S}\} > 0$. Applying Lemma 11, we can conclude that $d_{k_j}^{(s)} > 0$ (since $s \in \mathcal{S}$) for j sufficiently large, in contradiction to the definition of the subsequence.

Since the filter mechanisms ensure $\lim_k \theta(x_k) = 0$ (even if the barrier objective function is unbounded above; see Remark 6), we can find K so that $\theta(x_k) \leq \min\{\delta_\theta, \tilde{\delta}_\theta\}$ for $k \geq K$ (recall the definition of $\tilde{\delta}_\theta$ and $\tilde{\delta}_x$ in Assumption (B8)). Define

$$\epsilon_x := \min\left\{ (1-\tau)\min\{\delta_x, \tilde{\delta}_x\}, \min_i\{x_k^{(i)} : k \leq K\} \right\} > 0.$$

By definition it is clear that $x_k \geq \epsilon_x e$ for $k \leq K$, which can be used as the anchor for a proof by induction. Now suppose that $x_k \geq \epsilon_x e$ for some $k \geq K$. Since $d_k^{(i)} > 0$ for $x_k^{(i)} \leq \delta_x$ for $k \notin \mathcal{R}$, as well as from Assumption (B8), we see that we can have $x_{k+1}^{(i)} < x_k^{(i)}$ for an index i only if $x_k^{(i)} \geq \min\{\delta_x, \tilde{\delta}_x\}$. From (66) we then obtain $x_{k+1}^{(i)} \geq (1-\tau)x_k^{(i)} \geq (1-\tau)\min\{\delta_x, \tilde{\delta}_x\}$ so that overall $x_{k+1} \geq \epsilon_x e$. □

Remark 9. For the overall barrier method as the barrier parameter μ is driven to zero, we may simply restart Algorithm I by deleting the current filter whenever the barrier parameter changes. Alternatively, we may choose to store the values of the two terms $f(x_l)$ and $\sum_i \ln(x_l^{(i)})$ in the barrier function $\varphi_\mu(x_l)$ separately for each corner entry (14) in the filter, which would allow one to initialize the filter for the new barrier problem under consideration of already known information. Details on such a procedure are beyond the scope of this paper.

5. Conclusions. A framework for line search filter methods that can be applied to barrier methods and active set SQP methods has been presented. Global convergence has been shown under mild assumptions, which are, in particular, less restrictive than those made previously for some line search interior point methods. The method also possesses favorable local convergence behavior, as we discuss in the companion paper [26]. We further proposed an alternative measure for the filter, using the Lagrangian function instead of the objective function, for which the global convergence properties still hold.

In a recent report [27] we present practical experience with the line search filter barrier method proposed in this paper. The numerical results on a large set of test problems show that the algorithm exhibits very good practical performance in terms of efficiency and robustness and that it is competitive with other current NLP codes.

Acknowledgments. Funding from the National Science Foundation (DMS-9706950) and by an unrestricted grant from the ExxonMobil Corporation is gratefully acknowledged. We are very indebted to Jorge Nocedal for intriguing discussions, helpful comments, and encouragement regarding this paper. We thank Caroline Sainvitu and Philippe Toint for their careful study of the manuscript and for catching several typos. Last but not least, we are also very grateful to Associate Editor Sven Leyffer and to several referees for their insightful comments and suggestions, which greatly helped to improve the paper.

REFERENCES

[1] C. Audet and J. E. Dennis, Jr., *A pattern search filter method for nonlinear programming without derivatives*, SIAM J. Optim., 14 (2004), pp. 980–1010.

[2] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.

[3] R. H. Byrd, J. C. Gilbert, and J. Nocedal, *A trust region method based on interior point techniques for nonlinear programming*, Math. Program., 89 (2000), pp. 149–185.

[4] R. H. Byrd, G. Liu, and J. Nocedal, *On the local behaviour of an interior point method for nonlinear programming*, in Numerical Analysis 1997, Pitman Res. Notes Math. Ser. 380, D. F. Griffiths and D. J. Higham, eds., Longman, Harlow, UK, 1998, pp. 37–56.

[5] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds*, SIAM J. Numer. Anal., 28 (1991), pp. 545–572.

[6] A. S. El-Bakry, R. A. Tapia, T. Tsuchiya, and Y. Zhang, *On the formulation and theory of the Newton interior-point method for nonlinear programming*, J. Optim. Theory Appl., 89 (1996), pp. 507–541.

[7] R. Fletcher, N. I. M. Gould, S. Leyffer, Ph. L. Toint, and A. Wächter, *Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming*, SIAM J. Optim., 13 (2002), pp. 635–659.

[8] R. Fletcher and S. Leyffer, *A Bundle Filter Method for Nonsmooth Nonlinear Optimization*, Tech. report NA/195, Department of Mathematics, University of Dundee, Nethergate, Dundee, Scotland, 1999.

[9] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Math. Program., 91 (2002), pp. 239–269.

[10] R. Fletcher, S. Leyffer, and Ph. L. Toint, *On the global convergence of a filter-SQP algorithm*, SIAM J. Optim., 13 (2002), pp. 44–59.

[11] D. M. Gay, M. L. Overton, and M. H. Wright, *A primal-dual interior method for nonconvex nonlinear programming*, in Advances in Nonlinear Programming, Y. Yuan, ed., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998, pp. 31–56.

[12] N. I. M. Gould, D. Orban, A. Sartenaer, and Ph. L. Toint, *Superlinear convergence of primal-dual interior point algorithms for nonlinear programming*, SIAM J. Optim., 11 (2001), pp. 974–1002.

[13] N. I. M. Gould, D. Orban, A. Sartenaer, and Ph. L. Toint, *Componentwise fast convergence in the solution of full-rank systems of nonlinear equations*, Math. Program., 92 (2002), pp. 481–508.

[14] S.-P. Han, *A globally convergent method for nonlinear programming*, J. Optim. Theory Appl., 22 (1977), pp. 297–309.

[15] M. Marazzi and J. Nocedal, *Feasibility control in nonlinear optimization*, in Foundations of Computational Mathematics, London Math. Soc. Lecture Note Ser. 284, A. DeVore, A. Iserles, and E. Suli, eds., Cambridge University Press, Cambridge, UK, 2001, pp. 125–154.

[16] S. G. Nash and A. Sofer, *Why Extrapolation Helps Barrier Methods*, Tech. report, Operations Research and Engineering Department, George Mason University, Fairfax, VA 22030, 1998.

[17] J. Nocedal and S. Wright, *Numerical Optimization*, Springer-Verlag, New York, NY, 1999.

[18] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, in Optimization, R. Fletcher, ed., Academic Press, New York, 1969, pp. 283–298.

[19] M. J. D. Powell, *A hybrid method for nonlinear equations*, in Numerical Methods for Nonlinear Algebraic Equations, P. Rabinowitz, ed., Gordon and Breach, London, 1970, pp. 87–114.

[20] A. L. Tits, A. Wächter, S. Bakhtiari, T. J. Urban, and C. T. Lawrence, *A primal-dual interior-point method for nonlinear programming with strong global and local convergence properties*, SIAM J. Optim., 14 (2003), pp. 173–199.

[21] M. Ulbrich, S. Ulbrich, and L. N. Vicente, *A globally convergent primal-dual interior-point filter method for nonlinear programming*, Math. Program., 100 (2004), pp. 379–410.

[22] S. Ulbrich, *On the superlinear local convergence of a filter-SQP method*, Math. Program., 100 (2004), pp. 217–245.

[23] R. J. Vanderbei and D. F. Shanno, *An interior-point algorithm for nonconvex nonlinear programming*, Comput. Optim. Appl., 13 (1999), pp. 231–252.

[24] A. Wächter, *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, 2002.

[25] A. Wächter and L. T. Biegler, *Failure of global convergence for a class of interior point methods for nonlinear programming*, Math. Program., 88 (2000), pp. 565–574.

[26] A. Wächter and L. T. Biegler, *Line search filter methods for nonlinear programming: Local convergence*, SIAM J. Optim., 6 (2005), pp. 32–48.

[27] A. Wächter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Math. Program., to appear.

[28] H. Yamashita, *A globally convergent primal-dual interior-point method for constrained optimization*, Optim. Methods Softw., 10 (1998), pp. 443–469.

[29] H. Yamashita, H. Yabe, and T. Tanabe, *A Globally and Superlinearly Convergent Primal-Dual Interior Point Trust Region Method for Large Scale Constrained Optimization*, Tech. report, Mathematical System Institute, Inc., Tokyo, Japan, 1997 (revised July 1998).

SIAM J. OPTIM.
Vol. 16, No. 1, pp. 32–48

LINE SEARCH FILTER METHODS FOR NONLINEAR PROGRAMMING: LOCAL CONVERGENCE[*]

ANDREAS WÄCHTER[†] AND LORENZ T. BIEGLER[‡]

Abstract. A line search method is proposed for nonlinear programming using Fletcher and Leyffer's filter method, which replaces the traditional merit function. A simple modification of the method proposed in a companion paper [*SIAM J. Optim.*, 16 (2005), pp. 1–31] introducing second order correction steps is presented. It is shown that the proposed method does not suffer from the Maratos effect, so that fast local convergence to second order sufficient local solutions is achieved.

Key words. nonlinear programming, nonconvex constrained optimization, filter method, line search, local convergence, Maratos effect, second order correction

AMS subject classifications. 49M37, 65K05, 90C30, 90C55

DOI. 10.1137/S1052623403426544

1. Introduction.

Recently, Fletcher and Leyffer [7] proposed filter trust region methods, offering an alternative to merit functions, as a tool to guarantee global convergence in algorithms for nonlinear programming. The underlying concept is that trial points are accepted if they improve the objective function *or* improve the constraint violation instead of a merit function. In a companion paper [14] we propose and analyze a filter line search method which can be applied to equality constrained nonlinear programs (NLPs), as well as problems with nonlinear equality and bound constraints using active set SQP methods and barrier interior point methods.

In this paper we discuss the local convergence properties of the filter line search algorithm proposed in [14]. As mentioned by Fletcher and Leyffer [7], the filter approach can suffer from the so-called Maratos effect [10]. The Maratos effect occurs if, arbitrarily close to a strict local solution of the NLP (1), a full Newton step increases *both* the objective function and the constraint violation, and is therefore rejected by the line search, even though it could be a very good step toward the solution. This can result in poor local convergence behavior. As a remedy, Fletcher and Leyffer propose to improve the search direction, if the full step is rejected, by means of a second order correction which aims to further reduce infeasibility. In this paper we show that this modification is indeed able to prevent the Maratos effect.

Ulbrich [13] has recently presented a trust region filter method using the Lagrangian function (instead of the objective function) as one of the measures in the filter (similar to what we propose in our companion paper [14]). In [13], Ulbrich shows fast local convergence without second order correction steps.

The paper is organized as follows. In order to keep the analysis simple, we focus first only on the easiest case of equality constrained optimization problems. In section 2 we revisit the filter line search procedure from the companion paper [14].

[*] Received by the editors April 18, 2003; accepted for publication (in revised form) October 6, 2004; published electronically August 17, 2005. This research was supported by the National Science Foundation (DMS-9706950) and an unrestricted grant from the ExxonMobil Corporation.

http://www.siam.org/journals/siopt/16-1/42654.html

[†] Department of Mathematical Sciences, IBM T.J. Watson Research Center, Yorktown Heights, NY 10598 (andreasw@watson.ibm.com).

[‡] Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 (lb01@andrew.cmu.edu).

Section 3 states the modified filter line search algorithm including second order correction steps. The local convergence analysis is presented in section 4. In section 5 we briefly discuss how this approach can be applied to a line search and a trust region filter SQP method to handle inequality constrained problems.

Notation. We denote the ith component of a vector $v \in \mathbb{R}^n$ by $v^{(i)}$. Norms $\|\cdot\|$ denote a fixed vector norm and its compatible matrix norm. We denote by $O(t_k)$ a sequence $\{v_k\}$ satisfying $\|v_k\| \leq \beta\, t_k$ for some constant $\beta > 0$ independent of k, and by $o(t_k)$ a sequence $\{v_k\}$ satisfying $\|v_k\| \leq \beta_k t_k$ for some positive sequence $\{\beta_k\}$ with $\lim_k \beta_k = 0$.

2. A line search filter method.

The proposed algorithm is a filter line search algorithm for solving nonlinear optimization problems of the form

$$\min_{x\in\mathbb{R}^n} \quad f(x) \tag{1a}$$

$$\text{subject to} \quad c(x) = 0, \tag{1b}$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$ and the equality constraints $c : \mathbb{R}^n \to \mathbb{R}^m$ with $m < n$ are twice continuously differentiable. The Karush–Kuhn–Tucker (KKT) conditions for this problem are given by

$$g(x) + A(x)\lambda = 0, \tag{2a}$$

$$c(x) = 0 \tag{2b}$$

with the Lagrangian multipliers λ, where $g(x) := \nabla f(x)$ and $A(x) := \nabla c(x)$. Under suitable *constraint qualifications*, such as linear independence of the constraint gradients $\nabla c(x)$, these are the first order optimality conditions for (1) (see, e.g., [12]).

Given a starting point x_0, the proposed line search algorithm generates a sequence of improved estimates x_k of the solution for the NLP (1). For this purpose in each iteration k a search direction d_k is computed from the linearization of the KKT conditions (2) at x_k,

$$\begin{bmatrix} H_k & A_k \\ A_k^T & 0 \end{bmatrix} \begin{pmatrix} d_k \\ \lambda_k^+ \end{pmatrix} = - \begin{pmatrix} g_k \\ c_k \end{pmatrix}. \tag{3}$$

Here, $A_k := A(x_k)$, $g_k := g(x_k)$, and $c_k := c(x_k)$. The symmetric matrix H_k denotes the Hessian $\nabla^2_{xx}\mathcal{L}(x_k, \lambda_k)$ of the Lagrangian

$$\mathcal{L}(x, \lambda) := f(x) + c(x)^T\lambda \tag{4}$$

of the NLP (1), or an approximation to this Hessian. The vector λ_k is some estimate of the optimal multipliers corresponding to the equality constraints (1b), and λ_k^+ in (3) can be used to determine a new estimate λ_{k+1} for the next iteration. In the context of this paper the particular choice of λ_k is not important. As is common for many line search methods, we assume that the projection of the Hessian approximation H_k onto the null space of the constraint Jacobian is uniformly positive definite to ensure certain descent properties.

After a search direction d_k has been computed, a step size $\alpha_k \in (0, 1]$ is determined in order to obtain the next iterate

$$x_{k+1} := x_k + \alpha_k d_k. \tag{5}$$

In the companion paper [14] we propose a backtracking line search procedure, where a decreasing sequence of step sizes $\alpha_{k,l} \in (0,1]$ $(l = 0,1,2,\dots)$ with $\lim_l \alpha_{k,l} = 0$ is tried until an acceptance criterion is satisfied. The procedure that decides which trial step size is accepted is a "filter method." In the remainder of this section we only briefly revisit this approach; the detailed motivation can be found in [14]. The formal statement of the algorithm is presented in section 3.

Filter methods were originally proposed by Fletcher and Leyffer [7]. The basic idea behind this approach is to interpret the optimization problem (1) as a biobjective optimization problem with the two goals of minimizing the objective function $f(x)$ and the constraint violation $\theta(x) := \|c(x)\|$ (with a certain emphasis on the latter quantity). Following this paradigm, we might consider a trial point $x_k(\alpha_{k,l}) := x_k + \alpha_{k,l} d_k$ during the line search to be acceptable if it leads to sufficient progress toward either goal compared to the current iterate, i.e., if

$$\theta(x_k(\alpha_{k,l})) \leq (1-\gamma_\theta)\theta(x_k) \tag{6a}$$

$$\text{or} \qquad f(x_k(\alpha_{k,l})) \leq f(x_k) - \gamma_f \theta(x_k) \tag{6b}$$

holds for fixed constants $\gamma_\theta, \gamma_f \in (0,1)$. However, the above criterion is replaced by requiring sufficient progress in the objective function, whenever the "switching condition"

$$g_k^T d_k < 0 \qquad \text{and} \qquad \alpha_{k,l}[-g_k^T d_k]^{s_f} > \delta\,[\theta(x_k)]^{s_\theta} \tag{7}$$

with constants $\delta > 0, s_\theta > 1, s_f > 2s_\theta$ holds.[1] If (7) is true for the current step size $\alpha_{k,l}$, the trial point has to satisfy the Armijo condition

$$f(x_k(\alpha_{k,l})) \leq f(x_k) + \eta_f \alpha_{k,l} g_k^T d_k, \tag{8}$$

instead of (6), in order to be acceptable. Here, $\eta_f \in (0, \frac{1}{2})$ is a constant. Since the projection of the matrix H_k in (3) onto the null space of A_k^T is uniformly positive definite, it can be shown that condition (7) becomes true if a feasible, but nonoptimal, point is approached. Enforcing decrease in the objective function by (8) then prevents the method from converging to such a point. In accordance with previous publications on filter methods (e.g., [6, 8]) we may call a trial step size $\alpha_{k,l}$ for which (7) holds, an "f-step size."

In order to prevent the method from cycling, the algorithm maintains a "filter" $\mathcal{F}_k \subseteq \{(\theta, f) \in \mathbb{R}^2 : \theta \geq 0\}$, a set of (θ, f)-pairs that are "prohibited" for a trial point in iteration k. During the line search, a trial point $x_k(\alpha_{k,l})$ is rejected if it is not acceptable to the current filter, i.e., if $(\theta(x_k(\alpha_{k,l})), f(x_k(\alpha_{k,l}))) \in \mathcal{F}_k$. At the beginning of the optimization, the filter is initialized to

$$\mathcal{F}_0 := \{(\theta, f) \in \mathbb{R}^2 : \theta \geq \theta^{\max}\}. \tag{9}$$

Later, the filter is augmented for a new iteration using the update formula

$$\mathcal{F}_{k+1} := \mathcal{F}_k \cup \Big\{(\theta, f) \in \mathbb{R}^2 : \theta \geq (1-\gamma_\theta)\theta(x_k) \quad \text{and} \quad f \geq f(x_k) - \gamma_f \theta(x_k)\Big\} \tag{10}$$

if the accepted trial step size does not satisfy the switching condition (7). In this way, the iterates cannot return back into the neighborhood of x_k. On the other hand, if (7)

[1] For the global convergence analysis in [14] it is sufficient if the constant s_f satisfies $s_f \geq 1$. However, for the proofs in this paper it has to satisfy a tighter condition, so that the relationship (26) below holds.

(and therefore also (8)) holds for the accepted step size, the filter remains unchanged. Because such an iteration guarantees progress in the objective function, we may call it an "f-type iteration."

Finally, in some cases it is not possible to find a trial step size $\alpha_{k,l}$ that satisfies the above criteria. Using linear models of the involved functions, we assume to be in this situation if $\alpha_{k,l}$ becomes smaller than

$$\alpha_k^{\min} := \gamma_\alpha \cdot \begin{cases} \min\left\{\gamma_\theta, \frac{\gamma_f \theta(x_k)}{-g_k^T d_k}, \frac{\delta[\theta(x_k)]^{s_\theta}}{[-g_k^T d_k]^{s_f}}\right\} & \text{if } g_k^T d_k < 0, \\ \gamma_\theta & \text{otherwise}, \end{cases} \tag{11}$$

with a "safety factor" $\gamma_\alpha \in (0,1]$. If the backtracking line search encounters a trial step size with $\alpha_{k,l} \leq \alpha_k^{\min}$, the algorithm reverts to a *feasibility restoration phase*. Here, the algorithm tries to find a new iterate x_{k+1} that is acceptable to the current filter and for which (6) holds, by reducing the constraint violation with some iterative method. Note that a suitable restoration phase algorithm might not be able to produce a new iterate for the filter line search method and instead converges to a local minimizer of the constraint violation, indicating to the user that the problem seems (at least locally) infeasible.

3. Second order correction steps.

It has been noted by Fletcher and Leyffer [7] that the filter approach, similar to a penalty function approach, can suffer from the Maratos effect. Here, a full Newton (or Newton-type) step increases *both* the objective function and the constraint violation, even arbitrarily close to a local solution of the NLP (1). As a consequence, the filter line search procedure rejects the full Newton step and accepts only small fractions of the step. This can result in poor local convergence behavior. As a remedy, Fletcher and Leyffer propose to improve the search direction by means of a second order correction.

A second order correction step $d_k^{\rm soc}$ aims to reduce infeasibility by applying an additional Newton-type step for the constraints at the point $x_k + d_k$. There is a wide range of options to compute such a step. Here, we assume that it is obtained from the solution of the linear system

$$\begin{bmatrix} H_k^{\rm soc} & A_k^{\rm soc} \\ (A_k^{\rm soc})^T & 0 \end{bmatrix} \begin{pmatrix} d_k^{\rm soc} \\ \lambda_k^{\rm soc} \end{pmatrix} = -\begin{pmatrix} g_k^{\rm soc} \\ c(x_k + d_k) + c_k^{\rm soc} \end{pmatrix}, \tag{12}$$

where $H_k^{\rm soc}$ is a symmetric $n \times n$ matrix, $A_k^{\rm soc} \in \mathbb{R}^{n\times m}$, $g_k^{\rm soc} \in \mathbb{R}^n$, and $c_k^{\rm soc} \in \mathbb{R}^m$. Second order correction steps of the form (12) are discussed by Conn, Gould, and Toint in [3, section 15.3.2.3]. We assume that $H_k^{\rm soc}$ is uniformly positive definite on the null space of $(A_k^{\rm soc})^T$, and that in a neighborhood of a second order sufficient solution we have

$$g_k^{\rm soc} = o(\|d_k\|), \qquad A_k - A_k^{\rm soc} = O(\|d_k\|), \qquad c_k^{\rm soc} = o(\|d_k\|^2). \tag{13}$$

In [3], the analysis is made for the particular choices $c_k^{\rm soc} = 0$, $A_k^{\rm soc} = A(x_k + p_k)$ for some $p_k = O(\|d_k\|)$, and $H_k = \nabla_{xx}^2 \mathcal{L}_\mu(x_k, \lambda_k)$ in (3) for multiplier estimates λ_k. However, the careful reader will be able to verify that the cited results from [3] still hold as long as

$$(W_k - H_k)d_k = o(\|d_k\|), \tag{14}$$

if x_k converges to a second order sufficient solution x_* of the NLP with corresponding multipliers λ_* (see Assumption (L2) below), where

$$W_k = \nabla^2_{xx}\mathcal{L}(x_k, \lambda_*) \overset{(4)}{=} \nabla^2 f(x_k) + \sum_{i=1}^{m} (\lambda_*)^{(i)} \nabla^2 c^{(i)}(x_k). \tag{15}$$

We note that if we choose $H_k := \nabla^2_{xx}\mathcal{L}_\mu(x_k, \lambda_k)$, where the sequence of multiplier estimates $\{\lambda_k\}$ is generated using λ_k^+ from (3) (e.g., by setting $\lambda_{k+1} := \lambda_k^+$), then (14) holds if x_k converges to a second order sufficient local solution x_* satisfying Assumption (L2) below.

Possible choices for the quantities in the computation of the second order correction step in (12) that satisfy (13) are the following.

(SOC-1) $H_k^{\text{soc}} = I$, $g_k^{\text{soc}} = 0$, $c_k^{\text{soc}} = 0$, and $A_k^{\text{soc}} = A_k$ or $A_k^{\text{soc}} = A(x_k + d_k)$; this corresponds to a least-squares step for the constraints.

(SOC-2) $H_k^{\text{soc}} = H_k$, $g_k^{\text{soc}} = 0$, $c_k^{\text{soc}} = 0$, and $A_k^{\text{soc}} = A_k$; this option is inexpensive since it allows us to reuse the factorization of the linear system (3).

(SOC-3) H_k^{soc} is the Hessian approximation corresponding to $x_k + d_k$, $g_k^{\text{soc}} = g(x_k + d_k) + A(x_k + d_k)\lambda_k^+$, $c_k^{\text{soc}} = 0$, and $A_k^{\text{soc}} = A(x_k + d_k)$; this step corresponds to the step in the next iteration, supposing that $x_k + d_k$ has been accepted. In this sense, this choice has the flavor of the watchdog technique [2].

(SOC-4) If d_k^{soc} is a second order correction step, and $\bar{d}_k^{\text{soc}}$ is an additional second order correction step (i.e., with "$c(x_k + d_k)$" replaced by "$c(x_k + d_k + d_k^{\text{soc}})$" in (12)), then $d_k^{\text{soc}} + \bar{d}_k^{\text{soc}}$ can be understood as a single second order correction step for d_k (in that case with $c_k^{\text{soc}} \neq 0$). Similarly, several consecutive correction steps can be considered as a single one.

It is easy to show that for the combined step $d_k + d_k^{\text{soc}}$ we have $c(x_k + d_k + d_k^{\text{soc}}) = o(\|d_k\|^2)$ (see (21b) below). As a consequence, the combined step has a better chance of being accepted by the filter method than the original step d_k if x_k is close to a local solution. In order to overcome the Maratos effect, we modify the filter line search procedure outlined in section 2, so that a second order correction step is tried whenever the full step has not been accepted. As we will see in section 4, this indeed enables the algorithm to accept full steps close to a second order sufficient solution of (1), so that fast local convergence is achieved.

We now formally state the line search filter algorithm from [14] with the modification to include second order correction steps.

Algorithm I.

Given: Starting point x_0; constants $\theta_{\max} \in (\theta(x_0), \infty]$; $\gamma_\theta, \gamma_f \in (0,1)$; $\delta > 0$; $\gamma_\alpha \in (0,1]$; $s_\theta > 1$; $s_f > 2s_\theta$; $\eta_f \in (0, \frac{1}{2})$; $0 < \tau_1 \leq \tau_2 < 1$.

1. *Initialize.* Initialize the filter (using (9)) and the iteration counter $k \leftarrow 0$.
2. *Check convergence.* Stop, if x_k is a local solution (or at least stationary point) of the NLP (1), i.e., if it satisfies the KKT conditions (2) for some $\lambda \in \mathbb{R}^m$.
3. *Compute search direction.* Compute the search direction d_k from the linear system (3). If this system is detected to be too ill-conditioned or singular, go to feasibility restoration phase in Step 8.
4. *Backtracking line search.*
   4.1. *Initialize line search.* Set $\alpha_{k,0} = 1$ and $l \leftarrow 0$.
   4.2. *Compute new trial point.* If the trial step size becomes too small, i.e., $\alpha_{k,l} < \alpha_k^{\min}$ with $\alpha_k^{\min}$ defined by (11), go to the feasibility restoration phase in Step 8. Otherwise, compute the new trial point $x_k(\alpha_{k,l}) := x_k + \alpha_{k,l} d_k$.

4.3. *Check acceptability to the filter.* If $(\theta(x_k(\alpha_{k,l})), f(x_k(\alpha_{k,l}))) \in \mathcal{F}_k$, reject the trial step size and go to Step 4.5.

4.4. *Check sufficient decrease with respect to current iterate.*

4.4.1. *Case* I: $\alpha_{k,l}$ *is an f-step size (i.e.,* (7) *holds):* If the Armijo condition (8) for the objective function holds, accept the trial step $x_{k+1} := x_k(\alpha_{k,l})$ and go to Step 5. Otherwise, go to Step 4.5.

4.4.2. *Case* II: $\alpha_{k,l}$ *is not an f-step size (i.e.,* (7) *is not satisfied):* If (6) holds, accept the trial step $x_{k+1} := x_k(\alpha_{k,l})$ and go to Step 5. Otherwise, go to Step 4.5.

4.5. *Compute second order correction step.* If $l \neq 0$, go to Step 4.8. Otherwise, solve the linear system (12) to obtain the second order correction step d_k^{soc} and define

$$\bar{x}_{k+1} := x_k + d_k + d_k^{\text{soc}}.$$

4.6. *Check acceptability to the filter.* If $\bar{x}_{k+1} \in \mathcal{F}_k$, reject the second order correction step and go to Step 4.8.

4.7. *Check sufficient decrease with respect to current iterate.*

4.7.1. *Case* I: *The switching condition* (7) *holds (for* $\alpha_{k,0}$ *and* d_k*):* If the Armijo condition for the objective function,

$$f(\bar{x}_{k+1}) \leq f(x_k) + \eta_f \, g_k^T d_k, \tag{16}$$

holds, accept $x_{k+1} := \bar{x}_{k+1}$ and go to Step 5. Otherwise, go to Step 4.8.

4.7.2. *Case* II: *The switching condition* (7) *is not satisfied:* If

$$\theta(\bar{x}_{k+1}) \leq (1 - \gamma_\theta)\theta(x_k) \tag{17a}$$

$$\text{or} \qquad f(\bar{x}_{k+1}) \leq f(x_k) - \gamma_f \theta(x_k) \tag{17b}$$

holds, accept $x_{k+1} := \bar{x}_{k+1}$ and go to Step 5. Otherwise, go to Step 4.8.

4.8. *Choose new trial step size.* Choose $\alpha_{k,l+1} \in [\tau_1 \alpha_{k,l}, \tau_2 \alpha_{k,l}]$, set $l \leftarrow l + 1$, and go back to Step 4.2.

5. *Accept trial point.* Set $\alpha_k := \alpha_{k,l}$.

6. *Augment filter if necessary.* If k is not an f-type iteration (i.e., (7) does not hold for α_k), augment the filter using (10); otherwise leave the filter unchanged, i.e., set $\mathcal{F}_{k+1} := \mathcal{F}_k$.

7. *Continue with next iteration.* Increase the iteration counter $k \leftarrow k + 1$ and go back to Step 2.

8. *Feasibility restoration phase.* Compute a new iterate x_{k+1} by decreasing the infeasibility measure θ, so that x_{k+1} satisfies the sufficient decrease conditions (6) and is acceptable to the filter, i.e., $(\theta(x_{k+1}), f(x_{k+1})) \notin \mathcal{F}_k$. Augment the filter using (10) (for x_k) and continue with the regular iteration in Step 7.

It can be verified easily that this modification of Algorithm I in the companion paper [14] does not affect the global convergence properties proved in [14].

4. Local convergence analysis.

We start the analysis by stating the necessary assumptions.

Assumptions L. Assume that the algorithm generates an infinite sequence $\{x_k\}$ of iterates that converges to a local solution x_* of the NLP (1), and that the following hold.

(L1) The functions f and c are twice continuously differentiable in a neighborhood of x_*.

(L2) x_* satisfies the following sufficient second order optimality conditions:

- there exists $\lambda_* \in \mathbb{R}^m$ so that the KKT conditions (2) are satisfied for (x_*, λ_*);
- the constraint Jacobian $A(x_*)^T$ has full rank; and
- the Hessian of the Lagrangian $W_* = \nabla^2_{xx}\mathcal{L}(x_*, \lambda_*)$ is positive definite on the null space of $A(x_*)^T$.

(L3) In (3), H_k is uniformly positive definite on the null space of $(A_k)^T$, as well as bounded.

(L4) In (12), H_k^{soc} is uniformly positive definite on the null space of $(A_k^{\mathrm{soc}})^T$, and (13) holds.

(L5) The matrices H_k in (3) satisfy (14).

(L6) There exists a constant $\theta_{\mathrm{inc}} > 0$, so that the algorithm does not switch in Step 3 to the restoration phase if $\theta(x_k) \leq \theta_{\mathrm{inc}}$.

The assumption "$\lim_k x_k = x_*$" is discussed in Remark 3.15 in the companion paper [14]. It is shown that if a particular restoration phase algorithm (based on Newton steps for the KKT conditions) is used in the neighborhood of a solution x_* satisfying (L2), then the iterates of the overall filter line search algorithm are attracted to x_* so that $x_k \to x_*$ follows. Assumption (L5) is reminiscent of the Dennis–Moré characterization of superlinear convergence [4], but it is stronger than the one necessary for superlinear convergence [1] which requires only that $Z_k^T(W_k - H_k)d_k = o(\|d_k\|)$, where Z_k is a null space matrix for A_k^T. However, if multiplier estimates λ_k based on λ_k^+ from (3) and exact second derivatives are used to obtain H_k close to x_*, i.e., if

$$H_k = \nabla^2_{xx}\mathcal{L}(x_k, \lambda_k), \tag{18}$$

then Assumptions (L3) and (L5) are satisfied, since $H_k \to W_*$ in that case. Finally, the algorithm allows us to revert to the restoration phase in Step 3. This option exists so that the overall globally convergent line search method can handle infeasible points at which the constraint gradients are linearly dependent (see [14] for details). Therefore, Assumption (L6) is introduced as a formality to guarantee that the algorithm does not switch in arbitrary iterations to the restoration phase close to feasible points. In light of Assumption (L2), it is easy to see that the iteration matrix in (3) is nonsingular close to x_*, if H_k is chosen to be close to W_* (e.g., by (18)), so that there is no need to revert to the restoration phase in Step 3 close to x_*, and Assumption (L6) is satisfied.

The above assumptions imply Assumptions G in the companion paper [14] in a neighborhood of the solution. Therefore, Lemma 1 from [14] remains valid close to x_*, which states that d_k and λ_k^+ from (3) are uniformly bounded. Furthermore, as can be verified easily, the proof of Lemma 4 in [14] holds using Assumptions (L3) and (L6), so that

$$\theta(x_k) = 0 \quad \Longrightarrow \quad g_k^T d_k < 0 \qquad \text{and} \tag{19}$$

$$\Theta_k := \min\{\theta : (\theta, f) \in \mathcal{F}_k\} > 0 \tag{20}$$

for all k.

First we summarize some preliminary results.

LEMMA 4.1. *Suppose Assumptions* L *hold. Then there exists a neighborhood U_1 of x_*, so that for all $x_k \in U_1$ we have*

$$d_k^{\mathrm{soc}} = o(\|d_k\|), \tag{21a}$$

$$c(x_k + d_k + d_k^{\mathrm{soc}}) = o(\|d_k\|^2). \tag{21b}$$

Proof. From continuity, condition (13), and full rank of A_*^T, as well as Assumption (L4), we have that the matrix in (12) has a uniformly bounded inverse in the neighborhood of x_*. Hence, since the right-hand side is $o(\|d_k\|)$, claim (21a) follows. Furthermore, from

$$\begin{aligned} c(x_k + d_k + d_k^{\text{soc}}) &= c(x_k + d_k) + A(x_k + d_k)^T d_k^{\text{soc}} + O(\|d_k^{\text{soc}}\|^2) \\ &\overset{(12)}{=} -c_k^{\text{soc}} - (A_k^{\text{soc}})^T d_k^{\text{soc}} + (A_k + O(\|d_k\|))^T d_k^{\text{soc}} \\ &\qquad + O(\|d_k^{\text{soc}}\|^2) \\ &\overset{(13)}{=} o(\|d_k\|^2) + O(\|d_k\|\|d_k^{\text{soc}}\|) + O(\|d_k^{\text{soc}}\|^2) \\ &\overset{(21a)}{=} o(\|d_k\|^2) \end{aligned}$$

for x_k close to x_*, the claim (21b) follows. □

In order to prove our local convergence result we make use of two results established in [3] regarding the effect of second order correction steps on the exact penalty function

$$\phi_\rho(x) = f(x) + \rho\, \theta(x). \tag{22}$$

Note that we employ the exact penalty function only as a technical device, but the algorithm never refers to it. We also use the following model of the penalty function:

$$q_\rho(x_k, d) = f(x_k) + g_k^T d + \frac{1}{2} d^T H_k d + \rho \left\| A_k^T d + c_k \right\|. \tag{23}$$

The first result follows from Theorem 15.3.7 in [3].

LEMMA 4.2. *Suppose Assumptions* L *hold. Let* ϕ_ρ *be the exact penalty function* (22) *and* q_ρ *defined by* (23) *with* $\rho > \|\lambda_*\|_D$, *where* $\|\cdot\|_D$ *is the dual norm to* $\|\cdot\|$. *Then,*

$$\lim_{k\to\infty} \frac{\phi_\rho(x_k) - \phi_\rho(x_k + d_k + d_k^{\text{soc}})}{q_\rho(x_k, 0) - q_\rho(x_k, d_k)} = 1. \tag{24}$$

The next result follows from Theorem 15.3.2 in [3].

LEMMA 4.3. *Suppose Assumptions* L *hold. Let* (d_k, λ_k^+) *be a solution of the linear system* (3), *and let* $\rho > \|\lambda_k^+\|_D$. *Then,*

$$q_\rho(x_k, 0) - q_\rho(x_k, d_k) \geq 0. \tag{25}$$

The next lemma shows that in a neighborhood of x_*, Step 4.7.1 of Algorithm I is successful if the combined step $d_k + d_k^{\text{soc}}$ is an f-type step.

LEMMA 4.4. *Suppose Assumptions* L *hold. Then there exists a neighborhood* $U_2 \subseteq U_1$ *of* x_* *so that whenever* (7) *holds for* $\alpha_{k,l} = 1$, *the Armijo condition* (16) *is satisfied.*

Proof. Choose U_1 to be the neighborhood from Lemma 4.1. It then follows that for $x_k \in U_1$ satisfying (7),

$$\theta(x_k) < \delta^{-\frac{1}{s_\theta}} [-g_k^T d_k]^{\frac{s_f}{s_\theta}} = O(\|d_k\|^{\frac{s_f}{s_\theta}}) = o(\|d_k\|^2), \tag{26}$$

since $\frac{s_f}{s_\theta} > 2$ and g_k is uniformly bounded in U_1.

Since $\eta_f < \frac{1}{2}$, Lemma 4.2 and (25) imply that there exists $K \in \mathbb{N}$ such that for all $k \geq K$ we have for some constant $\rho > 0$ with $\rho > \|\lambda_k^+\|_D$ independent of k that

$$\phi_\rho(x_k) - \phi_\rho(x_k + d_k + d_k^{\text{soc}}) \geq \left(\frac{1}{2} + \eta_f\right)(q_\rho(x_k, 0) - q_\rho(x_k, d_k)). \tag{27}$$

We then have

$$\begin{aligned} & f(x_k) - f(x_k + d_k + d_k^{\text{soc}}) \\ \overset{(22)}{=}\; & \phi_\rho(x_k) - \phi_\rho(x_k + d_k + d_k^{\text{soc}}) - \rho\left(\theta(x_k) - \theta(x_k + d_k + d_k^{\text{soc}})\right) \\ \overset{(27),(21b),(26)}{\geq}\; & \left(\frac{1}{2} + \eta_f\right)(q_\rho(x_k, 0) - q_\rho(x_k, d_k)) + o(\|d_k\|^2) \\ \overset{(23),(26),(3)}{=}\; & -\left(\frac{1}{2} + \eta_f\right)\left(g_k^T d_k + \frac{1}{2} d_k^T H_k d_k\right) + o(\|d_k\|^2). \end{aligned} \tag{28}$$

Before continuing, we recall the step decomposition from the companion paper [14]

$$d_k = q_k + p_k, \tag{29a}$$

$$q_k := Y_k \bar{q}_k \quad \text{and} \quad p_k := Z_k \bar{p}_k, \tag{29b}$$

$$\bar{q}_k := -\left[A_k^T Y_k\right]^{-1} c_k, \tag{29c}$$

$$\bar{p}_k := -\left[Z_k^T H_k Z_k\right]^{-1} Z_k^T (g_k + H_k q_k), \tag{29d}$$

where $Y_k \in \mathbb{R}^{n \times m}$ and $Z_k \in \mathbb{R}^{n \times (n-m)}$ are matrices so that the columns of $[Y_k\ Z_k]$ form an orthonormal basis of $\mathbb{R}^n$, and the columns of Z_k are a basis of the null space of A_k^T. Since Assumptions L guarantee that the quantities (29), as well as λ_k^+, are bounded for k sufficiently large, we can conclude

$$\begin{aligned} & f(x_k) + \eta_f g_k^T d_k - f(x_k + d_k + d_k^{\text{soc}}) \\ \overset{(28)}{\geq}\; & -\frac{1}{2} g_k^T d_k - \left(\frac{1}{4} + \frac{\eta_f}{2}\right) d_k^T H_k d_k + o(\|d_k\|^2) \\ \overset{(3)}{=}\; & \frac{1}{2}\left(d_k^T H_k d_k + d_k^T A_k \lambda_k^+\right) - \left(\frac{1}{4} + \frac{\eta_f}{2}\right) d_k^T H_k d_k + o(\|d_k\|^2) \\ \overset{(3)}{=}\; & \left(\frac{1}{4} - \frac{\eta_f}{2}\right) d_k^T H_k d_k - \frac{1}{2} c(x_k)^T \lambda_k^+ + o(\|d_k\|^2) \\ \overset{(26)}{=}\; & \left(\frac{1}{4} - \frac{\eta_f}{2}\right) d_k^T H_k d_k + o(\|d_k\|^2) \\ \overset{(29)}{=}\; & \left(\frac{1}{4} - \frac{\eta_f}{2}\right) \bar{p}_k^T Z_k^T H_k Z_k \bar{p}_k + O(\|q_k\|) + o(\|d_k\|^2). \end{aligned} \tag{30}$$

Finally, using repeatedly the orthonormality of $[Y_k\ Z_k]$, we have

$$\begin{aligned} q_k &= O(\bar{q}_k) \overset{(29c)}{=} O(\theta(x_k)) \overset{(26)}{=} o(\|d_k\|^2) \\ &\overset{(29a)}{=} o(p_k^T p_k + q_k^T q_k) \overset{(29b)}{=} o(\|\bar{p}_k\|^2) + o(\|q_k\|^2) \end{aligned}$$

and therefore $q_k = o(\|\bar{p}_k\|^2)$, as well as

$$d_k \overset{(29a)}{=} O(\|q_k\|) + O(\|p_k\|) \overset{(29b)}{=} o(\|\bar{p}_k\|^2) + O(\|\bar{p}_k\|) = O(\|\bar{p}_k\|).$$

Since $\bar{p}_k \to 0$ as $x_k \to x_*$, (16) is then implied by (30), Assumption (L3), and $\eta_f < \frac{1}{2}$, if x_k is sufficiently close to x_*. □

It remains to show that also the filter and the sufficient reduction criterion (6) do not interfere with the acceptance of full steps close to x_*. The following technical lemmas address this issue and prepare the proof of the main local convergence theorem.

LEMMA 4.5. *Suppose Assumptions* L *hold. Then there exists a neighborhood* $U_3 \subseteq U_2$ *(with* U_2 *from Lemma* 4.4*) and constants* $\rho_1, \rho_2, \rho_3 > 0$ *with*

$$\rho_3 = (1-\gamma_\theta)\rho_2 - \gamma_f, \tag{31a}$$

$$2\gamma_\theta\rho_2 < (1+\gamma_\theta)(\rho_2-\rho_1) - 2\gamma_f, \tag{31b}$$

$$2\rho_3 \geq (1+\gamma_\theta)\rho_1 + (1-\gamma_\theta)\rho_2, \tag{31c}$$

so that for all $x_k \in U_3$ *we have* $\|\lambda_k^+\|_D < \rho_i$ *for* $i = 1, 2, 3$. *Furthermore, for all* $x_k \in U_3$ *we have*

$$\phi_{\rho_i}(x_k) - \phi_{\rho_i}(x_k + d_k + \bar{d}_k^{\rm soc}) \geq \frac{1+\gamma_\theta}{2}\left(q_{\rho_i}(x_k,0) - q_{\rho_i}(x_k,d_k)\right) \overset{(25)}{\geq} 0 \tag{32}$$

for $i = 2, 3$ *and all choices*

$$\bar{d}_k^{\rm soc} = d_k^{\rm soc}, \tag{33a}$$

$$\bar{d}_k^{\rm soc} = \sigma_k d_k^{\rm soc} + d_{k+1} + \sigma_{k+1} d_{k+1}^{\rm soc}, \tag{33b}$$

$$\bar{d}_k^{\rm soc} = \sigma_k d_k^{\rm soc} + d_{k+1} + \sigma_{k+1} d_{k+1}^{\rm soc} + d_{k+2} + \sigma_{k+2} d_{k+2}^{\rm soc}, \tag{33c}$$

$$\begin{aligned} \textit{or} \qquad \bar{d}_k^{\rm soc} = {} & \sigma_k d_k^{\rm soc} + d_{k+1} + \sigma_{k+1} d_{k+1}^{\rm soc} + d_{k+2} + \sigma_{k+2} d_{k+2}^{\rm soc} \\ & + d_{k+3} + \sigma_{k+3} d_{k+3}^{\rm soc}, \end{aligned} \tag{33d}$$

with $\sigma_k, \sigma_{k+1}, \sigma_{k+2}, \sigma_{k+3} \in \{0,1\}$, *as long as* $x_{l+1} = x_l + d_l + \sigma_l d_k^{\rm soc}$ *for* $l \in \{k, \ldots, k+j\}$ *with* $j \in \{-1, 0, 1, 2\}$, *respectively.*

Proof. Since λ_k^+ is uniformly bounded for all k with $x_k \in U_2$, we can find $\rho_1 > \|\lambda_*\|_D$ with

$$\rho_1 > \|\lambda_k^+\|_D \tag{34}$$

for all k with $x_k \in U_2$. Defining now

$$\rho_2 := \frac{1+\gamma_\theta}{1-\gamma_\theta}\rho_1 + \frac{3\gamma_f}{1-\gamma_\theta}$$

and ρ_3 by (31a), it is then easy to verify that $\rho_2, \rho_3 \geq \rho_1 > \|\lambda_k^+\|_D$ and that (31b) and (31c) hold. Since $(1+\gamma_\theta) < 2$, Lemma 4.2 implies that there exists a neighborhood $U_3 \subseteq U_2$ of x_*, so that (32) holds for $x_k \in U_3$, since according to the second order correction step choices (SOC-3) and (SOC-4) in section 3 all options for $\bar{d}_k^{\rm soc}$ in (33) can be understood as second order correction steps to d_k. □

Before proceeding we give a short graphical motivation of the remainder of the proof and introduce some more notation. Let U_3 and ρ_i be the neighborhood and constants from Lemma 4.5. Since $\lim_k x_k = x_*$, we can find $K_1 \in \mathbb{N}$ so that $x_k \in U_3$ for all $k \geq K_1$. In Figure 1 we see the (θ, f) half-plane with the current filter $\mathcal{F}_{K_1}$. Let us now define the level set

$$M := \{x \in U_3 : \phi_{\rho_3}(x) \leq \phi_{\rho_3}(x_*) + \kappa\}, \tag{35}$$

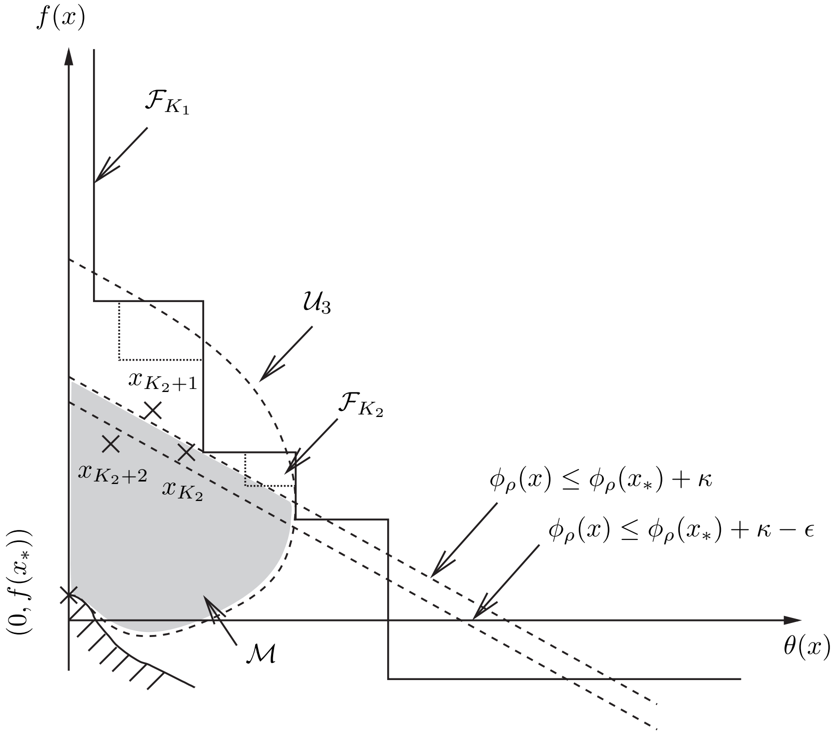


FIG. 1. *Basic idea of the proof.*

where $\kappa > 0$ is chosen so that for all $x \in M$ we have $(\theta(x), f(x)) \notin \mathcal{F}_{K_1}$. This is possible, since $\Theta_{K_1} > 0$ from (20), and since $\max\{\theta(x) : x \in M\}$ converges to zero as $\kappa \to 0$, because x_* is a strict local minimizer of ϕ_{ρ_3} (see [9]). Obviously, $x_* \in M$.

In Figure 1, $\mathcal{M}$ and $\mathcal{U}_3$ are the images of M and U_3 in the (θ, f) half-plane. Let K_2 now be the first iteration $K_2 \geq K_1$ with $x_{K_2} \in M$. This means that no iterate after K_1 and before K_2 is in M, and therefore also that the filter $\mathcal{F}_{K_2}$ overlaps with $\mathcal{M}$ by at most a small area whose size is governed by the parameters γ_f and γ_θ. The (θ, f)-pairs with constant value of the exact penalty function (22) correspond to dashed lines in the diagram, the slope of which is determined by the penalty parameter ρ. The main trick of the proof is to use these dashed lines as frontiers approaching $(0, f(x_*))$, so that the filter always lies to the upper right side of these lines (except for small overlaps coming from (10) in the filter update rule), and at least every other iterate (with or without second order correction step) lies on the lower left side of these lines (see (32)). For technical reasons we have to consider three of those frontiers corresponding to different values of the penalty parameter, in order to deal with sufficient progress with respect to the old filter entries, the current iterate (6), and new filter entries.

We denote the set of iteration indices k, in which the filter is augmented, by $\mathcal{A} \subseteq \mathbb{N}$; i.e.,

$$\mathcal{F}_k \subsetneq \mathcal{F}_{k+1} \qquad \Longleftrightarrow \qquad k \in \mathcal{A}.$$

Also, we define for $k \in \mathbb{N}$ the filter building blocks

$$\mathcal{G}_k := \Big\{(\theta, f) : \theta \geq (1 - \gamma_\theta)\theta(x_k) \quad \text{and} \quad f \geq f(x_k) - \gamma_f \theta(x_k)\Big\}$$

and index sets $I_{k_1}^{k_2} := \{l = k_1, \dots, k_2 - 1 : l \in \mathcal{A}\}$ for $k_1 \leq k_2$. Then it follows from the filter update rule (10) and the definition of $\mathcal{A}$ that for $k_1 \leq k_2$

$$\mathcal{F}_{k_2} = \mathcal{F}_{k_1} \cup \bigcup_{l \in I_{k_1}^{k_2}} \mathcal{G}_l. \tag{36}$$

Also note that $l \in I_{k_1}^{k_2} \subseteq \mathcal{A}$ implies $\theta(x_l) > 0$. Otherwise, we would have from (19) that $g_k^T d_k < 0$, so that (7) holds for all trial step sizes α, and the step must have been accepted in Step 4.4.1 or Step 4.7.1, hence satisfying (8) or (16). This would contradict the filter update condition in Step 6.

The last lemma enables us to show in the main theorem of this section that, once the iterates have reached the level set M, the full step is always acceptable to the current filter.

Lemma 4.6. *Suppose Assumptions* L *hold and let $l \geq K_1$ with $\theta(x_l) > 0$. Then the following statements hold for a given $x \in \mathbb{R}^n$.*

$$\left.\begin{array}{l}\textit{If } \phi_{\rho_2}(x_l) - \phi_{\rho_2}(x) \geq \frac{1+\gamma_\theta}{2}\left(q_{\rho_2}(x_l, 0) - q_{\rho_2}(x_l, d_l)\right),\\ \textit{then } (\theta(x), f(x)) \notin \mathcal{G}_l.\end{array}\right\} \tag{37}$$

$$\left.\begin{array}{l}\textit{If } x \in M \textit{ and } \phi_{\rho_2}(x_{K_2}) - \phi_{\rho_2}(x) \geq \frac{1+\gamma_\theta}{2}\left(q_{\rho_2}(x_{K_2}, 0) - q_{\rho_2}(x_{K_2}, d_{K_2})\right),\\ \textit{then } (\theta(x), f(x)) \notin \mathcal{F}_{K_2}.\end{array}\right\} \tag{38}$$

Proof of (37). Since $\rho_1 > \|\lambda_l^+\|_D$ we have from Lemma 4.3 that $q_{\rho_1}(x_l, 0) - q_{\rho_1}(x_l, d_l) \geq 0$. Hence, using the definition (23) for q_ρ, as well as $A_l^T d_l + c_l = 0$ (from (3)), we obtain

$$\begin{aligned}\phi_{\rho_2}(x_l) - \phi_{\rho_2}(x) &\geq \frac{1+\gamma_\theta}{2}\left(q_{\rho_2}(x_l, 0) - q_{\rho_2}(x_l, d_l)\right)\\ &= \frac{1+\gamma_\theta}{2}\left(q_{\rho_1}(x_l, 0) - q_{\rho_1}(x_l, d_l) + (\rho_2 - \rho_1)\theta(x_l)\right)\\ &\overset{(25)}{\geq} \frac{1+\gamma_\theta}{2}(\rho_2 - \rho_1)\theta(x_l). \qquad (39)\end{aligned}$$

If $f(x) < f(x_l) - \gamma_f \theta(x_l)$, the claim follows immediately. Otherwise, suppose that $f(x) \geq f(x_l) - \gamma_f \theta(x_l)$. In that case, we have together with $\theta(x_l) > 0$ that

$$\begin{aligned}\theta(x_l) - \theta(x) &\overset{(22),(39)}{\geq} \frac{1+\gamma_\theta}{2\rho_2}(\rho_2 - \rho_1)\theta(x_l) + \frac{1}{\rho_2}\left(f(x) - f(x_l)\right)\\ &\geq \frac{1+\gamma_\theta}{2\rho_2}(\rho_2 - \rho_1)\theta(x_l) - \frac{\gamma_f}{\rho_2}\theta(x_l)\\ &\overset{(31b)}{>} \gamma_\theta \theta(x_l),\end{aligned}$$

so that $(\theta(x), f(x)) \notin \mathcal{G}_l$.

Proof of (38). Since $x \in M$, it follows by the choice of κ in (35) that $(\theta(x), f(x)) \notin \mathcal{F}_{K_1}$. Thus, according to (36) it remains to show that for all $l \in I_{K_1}^{K_2}$ we have $(\theta(x), f(x)) \notin \mathcal{G}_l$. Choose $l \in I_{K_1}^{K_2}$. As in (39) we can show that

$$\phi_{\rho_2}(x_{K_2}) - \phi_{\rho_2}(x) \geq \frac{1+\gamma_\theta}{2}(\rho_2 - \rho_1)\theta(x_{K_2}). \tag{40}$$

Since $x \in M$, it follows from the definition of K_2 (as the first iterate after K_1 with $x_{K_2} \in M$) and the fact that $l < K_2$ that

$$\begin{aligned}\phi_{\rho_3}(x_l) &\overset{(35)}{>} \phi_{\rho_3}(x_{K_2}) \overset{(22)}{=} \phi_{\rho_2}(x_{K_2}) + (\rho_3 - \rho_2)\theta(x_{K_2})\\ &\overset{(40)}{\geq} \phi_{\rho_2}(x) + \left(\rho_3 - \frac{1+\gamma_\theta}{2}\rho_1 - \frac{1-\gamma_\theta}{2}\rho_2\right)\theta(x_{K_2})\\ &\overset{(31c)}{\geq} \phi_{\rho_2}(x). \qquad (41)\end{aligned}$$

If $f(x) < f(x_l) - \gamma_f\theta(x_l)$, we immediately have $(\theta(x), f(x)) \notin \mathcal{G}_l$. Otherwise we have $f(x) \geq f(x_l) - \gamma_f\theta(x_l)$ which yields

$$\begin{aligned}\theta(x) &\overset{(22),(41)}{<} \frac{1}{\rho_2}\left(f(x_l) + \rho_3\theta(x_l) - f(x)\right)\\ &\leq \frac{\rho_3 + \gamma_f}{\rho_2}\theta(x_l)\\ &\overset{(31a)}{=} (1-\gamma_\theta)\theta(x_l),\end{aligned}$$

so that $(\theta(x), f(x)) \notin \mathcal{G}_l$ which concludes the proof of (38). ◻

After these preparations we are finally able to show the main local convergence theorem.

Theorem 4.7. *Suppose Assumptions* L *hold. Then, for k sufficiently large, full steps of the form $x_{k+1} = x_k + d_k$ or $x_{k+1} = x_k + d_k + d_k^{\text{soc}}$ are taken, and x_k converges to x_* superlinearly.*

Proof. Recall that $K_2 \geq K_1$ is the first iteration after K_1 with $x_{K_2} \in M \subseteq U_3$. We now show by induction that the following statements are true for $k \geq K_2 + 2$:

$$(\text{i}_k)\phi_{\rho_i}(x_l) - \phi_{\rho_i}(x_k) \geq \frac{1+\gamma_\theta}{2}\left(q_{\rho_i}(x_l, 0) - q_{\rho_i}(x_l, d_l)\right) \quad \text{for } i \in \{2,3\} \text{ and } K_2 \leq l \leq k-2,$$

$$(\text{ii}_k)x_k \in M,$$

$$(\text{iii}_k)x_k = x_{k-1} + d_{k-1} + \sigma_{k-1}d_{k-1}^{\text{soc}} \qquad \text{with } \sigma_{k-1} \in \{0,1\}.$$

We start by showing that these statements are true for $k = K_2 + 2$.

Suppose the point $x_{K_2} + d_{K_2}$ is not accepted by the line search. In that case, define $\bar{x}_{K_2+1} := x_{K_2} + d_{K_2} + d_{K_2}^{\text{soc}}$. Then, from (32) with $i = 3$, $k = K_2$, and (33a), we see from $x_{K_2} \in M$ and the definition of M that $\bar{x}_{K_2+1} \in M$. After applying (32) again with $i = 2$ it follows from (38) that $(\theta(\bar{x}_{K_2+1}), f(\bar{x}_{K_2+1})) \notin \mathcal{F}_{K_2}$, i.e., $\bar{x}_{K_2+1}$ is not rejected in Step 4.6. Furthermore, if the switching condition (7) holds, we see from Lemma 4.4 that the Armijo condition (16) with $k = K_2$ is satisfied for the point $\bar{x}_{K_2+1}$. In the other case, i.e., if (7) is violated (note that then (19) and (7) imply $\theta(x_{K_2}) > 0$), we see from (32) for $i = 2$, $k = K_2$, and (33a), together with (37) for $l = K_2$, that (17) holds. Hence, $\bar{x}_{K_2+1}$ is also not rejected in Step 4.7 and accepted as the next iterate. Summarizing the discussion in this paragraph we can write $x_{K_2+1} = x_{K_2} + d_{K_2} + \sigma_{K_2}d_{K_2}^{\text{soc}}$ with $\sigma_{K_2} \in \{0,1\}$.

Let us now consider iteration $K_2 + 1$. For $\sigma_{K_2+1} \in \{0,1\}$ we have from (32) for $k = K_2$ and (33b) that

$$\begin{aligned}&\phi_{\rho_i}(x_{K_2}) - \phi_{\rho_i}(x_{K_2+1} + d_{K_2+1} + \sigma_{K_2+1}d_{K_2+1}^{\text{soc}})\\ &\geq \frac{1+\gamma_\theta}{2}\left(q_{\rho_i}(x_{K_2}, 0) - q_{\rho_i}(x_{K_2}, d_{K_2})\right) \qquad (42)\end{aligned}$$

for $i = 2, 3$, which yields

$$x_{K_2+1} + d_{K_2+1} + \sigma_{K_2+1} d^{\text{soc}}_{K_2+1} \in M. \tag{43}$$

If $x_{K_2+1} + d_{K_2+1}$ is accepted as the next iterate x_{K_2+2}, we immediately obtain from (42) and (43) that (i$_{K_2+2}$)–(iii$_{K_2+2}$) hold. Otherwise, we consider the case $\sigma_{K_2+1} = 1$. From (42), (43), and (38) we have for $\bar{x}_{K_2+2} := x_{K_2+1} + d_{K_2+1} + d^{\text{soc}}_{K_2+1}$ that $(\theta(\bar{x}_{K_2+2}), f(\bar{x}_{K_2+2})) \notin \mathcal{F}_{K_2}$. If $K_2 \notin I^{K_2+1}_{K_2}$, it immediately follows from (36) that $(\theta(\bar{x}_{K_2+2}), f(\bar{x}_{K_2+2})) \notin \mathcal{F}_{K_2+1}$. Otherwise, we have $\theta(x_{K_2}) > 0$. Then, (42) for $i = 2$ together with (37) implies $(\theta(\bar{x}_{K_2+2}), f(\bar{x}_{K_2+2})) \notin \mathcal{G}_{K_2}$, and hence with (36) we have $(\theta(\bar{x}_{K_2+2}), f(\bar{x}_{K_2+2})) \notin \mathcal{F}_{K_2+1}$, so that $\bar{x}_{K_2+2}$ is not rejected in Step 4.6. Arguing similarly as in the previous paragraph we can conclude that $\bar{x}_{K_2+2}$ is also not rejected in Step 4.7. Therefore, $x_{K_2+2} = \bar{x}_{K_2+2}$. Together with (42) and (43) this proves (i$_{K_2+2}$)–(iii$_{K_2+2}$) for the case $\sigma_{K_2+1} = 1$.

Now suppose that (i$_l$)–(iii$_l$) are true for all $K_2 + 2 \leq l \leq k$ with some $k \geq K_2 + 2$. If $x_k + d_k$ is accepted by the line search, define $\sigma_k := 0$; otherwise, $\sigma_k := 1$. Set $\bar{x}_{k+1} := x_k + d_k + \sigma_k d^{\text{soc}}_k$. From (32) for (33c) we then have for $i = 2, 3$

$$\phi_{\rho_i}(x_{k-1}) - \phi_{\rho_i}(\bar{x}_{k+1}) \geq \frac{1+\gamma_\theta}{2}\left(q_{\rho_i}(x_{k-1}, 0) - q_{\rho_i}(x_{k-1}, d_{k-1})\right) \geq 0. \tag{44}$$

Choose l with $K_2 \leq l < k - 1$ and consider two cases.

Case (a). If $k = K_2 + 2$, then $l = K_2$, and it follows from (32) with (33d) that for $i = 2, 3$

$$\phi_{\rho_i}(x_l) - \phi_{\rho_i}(\bar{x}_{k+1}) \geq \frac{1+\gamma_\theta}{2}\left(q_{\rho_i}(x_l, 0) - q_{\rho_i}(x_l, d_l)\right) \geq 0. \tag{45}$$

Case (b). If $k > K_2 + 2$, we have from (44) that $\phi_{\rho_i}(\bar{x}_{k+1}) \leq \phi_{\rho_i}(x_{k-1})$, and hence from (i$_{k-1}$) it follows that (45) also holds in this case.

In either case, (45) implies in particular that $\phi_{\rho_3}(\bar{x}_{k+1}) \leq \phi_{\rho_3}(x_{K_2})$, and since $x_{K_2} \in M$, we obtain

$$\bar{x}_{k+1} \in M. \tag{46}$$

If $x_k + d_k$ is accepted by the line search, (i$_{k+1}$)–(iii$_{k+1}$) follow from (45), (44), and (46). If $x_k + d_k$ is rejected, we see from (46), (45) for $i = 2$ and $l = K_2$, and (38) that $(\theta(\bar{x}_{k+1}), f(\bar{x}_{k+1})) \notin \mathcal{F}_{K_2}$. Furthermore, for $l \in I^k_{K_2}$ we have from (44) and (45) with (37) that $(\theta(\bar{x}_{k+1}), f(\bar{x}_{k+1})) \notin \mathcal{G}_l$, and hence from (36) that $\bar{x}_{k+1}$ is not rejected in Step 4.6. We can again show as before that $\bar{x}_{k+1}$ is not rejected in Step 4.7, so that $x_{k+1} = \bar{x}_{k+1}$ which implies (i$_{k+1}$)–(iii$_{k+1}$).

That $\{x_k\}$ converges to x_* with a superlinear rate follows from (14) (see, e.g., [11]). □

Remark 4.8. As can be expected, the convergence rate of x_k toward x_* is quadratic if (14) is replaced by

$$(W_k - H_k)d_k = O(\|d_k\|^2)$$

(see, e.g., [3]).

5. Fast local convergence of SQP methods.

5.1. A line search filter SQP method. In the companion paper [14] we propose a filter line search SQP method for solving NLPs with bound constraints, for simplicity stated in the form

$$\min_{x\in\mathbb{R}^n} \quad f(x) \tag{47a}$$

$$\text{subject to} \quad c(x) = 0, \tag{47b}$$

$$x \geq 0. \tag{47c}$$

The filter line search algorithm is essentially identical to the one for solving the equality constrained problem (1), where the search direction d_k is now computed as a solution of the QP

$$\min_{d\in\mathbb{R}^n} \quad g_k^T d + \frac{1}{2} d^T H_k d \tag{48a}$$

$$\text{subject to} \quad A_k^T d + c_k = 0, \tag{48b}$$

$$x_k + d \geq 0. \tag{48c}$$

The QP Hessian H_k is assumed to be positive definite in the null space of the constraints active at x_k and $x_k + d_k$. Since the initial iterate as well as all iterates returned from the feasibility restoration phase are assumed to satisfy the bound constraints, we have from (48c) and (5) that $x_k \geq 0$ for all k. Therefore, the infeasibility is still measured as $\theta(x) := \|c(x)\|$. For details, see [14].

In order to achieve fast local convergence for this active set approach, we can again use second order correction steps. One possibility for computing a second order correction step in an SQP framework is proposed in [5], where the composite step $\tilde{d}_k = d_k + d_k^{\text{soc}}$ is obtained as a solution of

$$\min_{\tilde{d}\in\mathbb{R}^n} \quad g_k^T \tilde{d} + \frac{1}{2} \tilde{d}^T H_k \tilde{d} \tag{49a}$$

$$\text{subject to} \quad A_k^T \tilde{d} + c_k + c(x_k + d_k) = 0, \tag{49b}$$

$$x_k + \tilde{d} \geq 0. \tag{49c}$$

This corresponds to the choice (SOC-2) in section 3.

Let us now assume that the iterates x_k generated by the SQP filter line search algorithm converge to a local solution x_* of (47) satisfying the following second order sufficient conditions:

- There exist multipliers $\lambda_* \in \mathbb{R}^m$ and $v_* \in \mathbb{R}^n$ with $v_* \geq 0$, so that the KKT conditions

$$\begin{aligned} g(x_*) + A(x_*)\lambda_* - v_* &= 0, \\ c(x_*) &= 0, \\ v_* \geq 0, \qquad x_* &\geq 0, \\ v_*^T x_* &= 0 \end{aligned}$$

  hold;
- the gradients of the constraints active at x_* are linearly independent;
- the Hessian of the Lagrangian, $W_* = \nabla^2 f(x_*) + \sum_{j=1,\ldots,m} \lambda_*^{(j)} \nabla^2 c^{(j)}(x_*)$, is positive definite in the null space of the active constraints;

- strict complementarity holds, i.e., $v_*^{(i)} + x_*^{(i)} > 0$ for all $i = 1, \dots, n$.

If we assume that the QP Hessians H_k are uniformly positive definite in the null space of the constraints active at x_k and $x_k + d_k$ (see Assumption (G3*) in [14]), then the bound constraints active at x_* are identical to the bound constraints active at the solution of (48) and (49) if x_k is sufficiently close to x_*. Therefore, for large k, the computation of d_k and $d_k^{\rm soc}$ from the QPs (48) and (49) can be interpreted as the steps obtained from Algorithm I applied to an equality constrained NLP, where the equality constraints consist of the equality constraints (47b) and constraints $x^{(i)} = 0$ for $i \in \{j : x_*^{(j)} = 0\}$. As a consequence, the analysis in the previous section can be applied.

5.2. A trust region filter SQP method.

In [6], Fletcher et al. propose a trust region filter SQP algorithm and analyze its global convergence behavior. The switching rule used there does not imply the relationship (26), and therefore the proof of Lemma 4.4 in our local convergence analysis does not hold for that method. However, it is easy to see that the global convergence analysis in [6] is still valid (in particular, Lemmas 3.7 and 3.10 in [6]) if the switching rule (2.19) in [6] is modified in analogy to (7) and replaced by

$$m_k(x_k) - m_k(x_k + s_k) \geq 0 \qquad \text{and} \qquad [m_k(x_k) - m_k(x_k + s_k)]^{s_f} \Delta_k^{1-s_f} \geq \kappa_\theta \theta_k^\psi,$$

where m_k is a quadratic model of the objective function, s_k is the trial step, Δ_k is the current trust region radius, $\kappa_\theta, \psi > 0$ are constants from [6] satisfying certain relationships, and the new constant $s_f > 0$ satisfies $s_f > 2\psi$. Then the local convergence analysis in section 4 is still valid (also for the quadratic model formulation), assuming that sufficiently close to a strict local solution the trust region is inactive, the trust region radius Δ_k is uniformly bounded away from zero, the (approximate) SQP steps s_k are computed sufficiently exactly, and a second order correction as discussed in section 3 is performed.

6. Conclusions.

We have shown that second order correction steps are able to overcome the Maratos effect within filter methods and that fast local convergence can be obtained. Important for the success of our analysis is a particular switching rule (7), which differs from previous filter methods, such as the one proposed by Fletcher et al. [6].

Acknowledgments. We are very indebted to Jorge Nocedal for intriguing discussions, very helpful comments, and encouragement regarding this paper. We thank Caroline Sainvitu and Philippe Toint for their careful study of the manuscript and for catching several typos. Last but not least, we are also very grateful to several referees for their insightful comments and suggestions, which greatly helped to improve the paper.

REFERENCES

[1] P. T. Boggs, J. W. Tolle, and P. Wang, *On the local convergence of quasi-Newton methods for constrained optimization*, SIAM J. Control Optim., 20 (1982), pp. 161–171.

[2] R. M. Chamberlain, C. Lemarechal, H. C. Pedersen, and M. J. D. Powell, *The watchdog technique for forcing convergence in algorithms for constrained optimization*, Math. Program. Study, 16 (1982), pp. 1–17.

[3] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust Region Methods*, MPS/SIAM Ser. Optim. 1, SIAM, Philadelphia, 2000.

[4] J. E. Dennis, Jr., and J. J. Moré, *Quasi-Newton methods, motivation and theory*, SIAM Rev., 19 (1977), pp. 46–89.

[5] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., John Wiley and Sons, New York, 1987.

[6] R. Fletcher, N. I. M. Gould, S. Leyffer, Ph. L. Toint, and A. Wächter, *Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming*, SIAM J. Optim., 13 (2002), pp. 635–659.

[7] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Math. Program., 91 (2002), pp. 239–269.

[8] R. Fletcher, S. Leyffer, and Ph. L. Toint, *On the global convergence of a filter–SQP algorithm*, SIAM J. Optim., 13 (2002), pp. 44–59.

[9] S.-P. Han, *A globally convergent method for nonlinear programming*, J. Optim. Theory Appl., 22 (1977), pp. 297–309.

[10] N. Maratos, *Exact Penalty Function Algorithms for Finite Dimensional and Control Optimization Problems*, Ph.D. thesis, University of London, London, 1978.

[11] J. Nocedal and M. L. Overton, *Projected Hessian updating algorithms for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 22 (1985), pp. 821–850.

[12] J. Nocedal and S. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.

[13] S. Ulbrich, *On the superlinear local convergence of a filter-SQP method*, Math. Program., 100 (2004), pp. 217–245.

[14] A. Wächter and L. T. Biegler, *Line search filter methods for nonlinear programming: Motivation and global convergence*, Tech. rep., Department of Chemical Engineering, Carnegie Mellon University, 2003, SIAM J. Optim., 16 (2005), pp. 1-31.

SIAM J. OPTIM.
Vol. 16, No. 1, pp. 49–68

A REVISED DUAL PROJECTIVE PIVOT ALGORITHM FOR LINEAR PROGRAMMING*

PING-QI PAN†

Abstract. We revise the *dual projective pivot algorithm* using sparse rectangular LU factors. In each iteration, the proposed algorithm solves at most three triangular systems, while simplex algorithms solve four. Instead of the classical basis, it uses a so-called pseudobasis (a rectangular matrix having fewer columns than rows), thereby solving smaller linear systems with a potentially improved stability compared to simplex algorithms. Most importantly, it generates good search directions at a low cost.

We report encouraging computational results on a set of 50 Netlib standard test problems as well as a set of 15 much larger real-world problems. A code named RDPPA 1.10 based on the proposed algorithm outperformed MINOS 5.51 significantly in terms of both iterations and run time. In particular, it appears that a high proportion of degenerate iterations need not imply many total iterations (contradicting the common belief).

Key words. large-scale linear programming, dual algorithm, projection, pivot, pseudobasis, deficient basis, LU factorization

AMS subject classifications. 65K05, 90C05

DOI. 10.1137/030602253

1. Introduction.

We consider linear programming (LP) problems in the standard form

$$\begin{aligned} &\text{minimize} && c^T x, \\ &\text{subject to} && Ax = b, \quad x \geq 0, \end{aligned} \tag{1.1}$$

where the constraint matrix $A \in R^{m\times n}$, cost vector $c \in R^n$, and right-hand side $b \in R^m$ are assumed to be nonzero ($0 < m < n$). We emphasize that no extra assumption is made either on the rank of A or on the consistency of $Ax = b$.

In essence, the dual simplex algorithm [14, 1, 5] solves (1.1) by handling its dual program, i.e.,

$$\begin{aligned} &\text{minimize} && b^T y, \\ &\text{subject to} && A^T y + z = c, \quad z \geq 0. \end{aligned} \tag{1.2}$$

Ever since it was introduced, the dual simplex algorithm has largely been present just in the literature. However, all that has changed in recent years. Its implementations have become so powerful that they work very competitively compared to primal simplex implementations and are now among the standard choices in modern codes [2]. Such a success encourages us to deal with the LP problem from the dual side.

Similarly, the square matrix termed *basis* has long played a fundamental role in simplex algorithms for solving LP problems [4, 6], yet the basis was extended recently to include a *deficient* case by exploiting primal degeneracy. Described as either primal variants [21, 22, 23, 26] or dual variants [20, 24], all the basis-deficiency-allowing

*Received by the editors December 8, 2003; accepted for publication (in revised form) April 26, 2005; published electronically August 17, 2005.
http://www.siam.org/journals/siopt/16-1/60225.html

†Department of Mathematics, Southeast University, Nanjing 210096, People's Republic of China (panpq@seu.edu.cn). Project 10371017 supported by National Natural Science Foundation of China.

algorithms performed very favorably in their dense implementations. Also, a sparse implementation of such a primal algorithm outperformed MINOS 5.3 significantly [25].

Thus, we are led to pursuing good dual approaches based on the deficient basis. While other authors handle primal degeneracy by taking a dual approach [13], the use of the deficient basis enables us to take *advantage* of primal degeneracy in a dual approach. The *dual projective simplex algorithm* was first derived along this line [20]. This algorithm seemed to be particularly attractive in that it involved a pivot rule as well as a projected search direction using the deficient basis and the QR factorization. The algorithm was then recast systematically into a compact form by handling a series of least-squares problems arising from the constraint matrix [24].

In this paper, a further revision using a sparse rectangular LU factorization is made, allowing the algorithm to proceed without storing any full array and thus making it more amenable to the solution of sparse problems. The revision has been implemented and tested on a set of 50 Netlib standard test problems as well as on a set of 15 much larger real-world problems with up to tens of thousands of rows and columns. A code named RDPPA 1.10 based on it significantly outperformed MINOS 5.51, the latest version of MINOS, in terms of both required iterations and run time. In particular, it appears that a high proportion of degenerate iterations need not imply many total iterations (contradicting common belief).

In the next section, basic definitions and assumptions are presented first. In sections 3 and 4, we focus on the optimality test and search direction. Then, in section 5, we describe the revised dual projective pivot algorithm (RDPPA). In section 6, remarks are made on the proposed algorithm about its motivation and an alternative view of it. In section 7, a dual phase-1 approach is described briefly. Finally, in section 8, computational results are reported, giving an insight into the encouraging behavior of the proposed algorithm.

2. Basic definitions and assumptions. One of the main features of the proposed algorithm is its use of a rectangular "pseudobasis." In this section, we briefly present the associated concepts and ideas and make some assumptions.

DEFINITION 2.1 (pseudobasis). *A pseudobasis is a submatrix consisting of any linearly independent columns of A; the submatrix consisting of the remaining columns is a pseudononbasis.*

DEFINITION 2.2 (basis). *A pseudobasis is a basis if its range space includes the right-hand side b; the associated pseudononbasis is a nonbasis.*

DEFINITION 2.3 (deficient basis). *If the number of columns of a basis is less than the number of its rows, it is a deficient basis; if a basis is square, it is a normal basis.*

Instead of the normal square basis, our algorithm proceeds with a pseudobasis, columns of which change dynamically in the solution process. As a result, the standard but unnatural full row rank assumption on A is no longer needed in our model statement (1.1).

Since real-world LP problems are almost always degenerate, or even highly degenerate, it can be expected that a great majority of bases encountered in practice are deficient. Consequently, the algorithm solves smaller linear systems than those solved in simplex algorithms. Moreover, a deficient basis is potentially better conditioned than a normal basis. Most importantly, as explained in section 5.3, use of the pseudobasis should enable us to form good search directions in the dual space at a low cost. This is a primary goal that we pursue in this work.

Let B be a pseudobasis with $s\,(1 \leq s \leq m)$ columns and let N be the associated pseudononbasis. The corresponding components of vectors and columns of matrices will be called *basic* and *nonbasic*, respectively. Denote by j_i the index of the ith column of B and by k_j the index of the jth column of N. Without confusion, denote *basic* and *nonbasic* ordered index sets again by B and N, respectively, i.e.,

$$B = \{j_1, \ldots, j_s\}, \quad N = \{k_1, \ldots, k_{n-s}\}. \tag{2.1}$$

Without loss of generality, components of vectors and columns of matrices will always be rearranged conformably to the ordered set $\{B, N\}$. The proposed algorithm will be developed with $s < m$ if not indicated otherwise.

Let an LU factorization of B with row and column exchanges be as follows:

$$PBQ = LU, \quad L = \begin{bmatrix} L_1 & \\ L_2 & I \end{bmatrix}, \quad U = \begin{bmatrix} U_1 \\ 0 \end{bmatrix}, \tag{2.2}$$

where $P \in R^{m\times m}$ and $Q \in R^{s\times s}$ are permutations that balance stability and sparsity, $L_1 \in R^{s\times s}$ is unit lower-triangular, and $U_1 \in R^{s\times s}$ is upper-triangular with nonzero diagonals. Accordingly, define the transformed right-hand side $\bar{b}$ by

$$Pb = L\bar{b}, \qquad \bar{b} = \begin{bmatrix} \bar{b}_1 \\ \bar{b}_2 \end{bmatrix} \begin{matrix} s \\ m-s \end{matrix} . \tag{2.3}$$

In what follows, without loss of generality, it might well be assumed that P and Q are the identity permutations.

Assume that a dual feasible solution $(\bar{y}, \bar{z})$ is available, satisfying

$$B^T\bar{y} = c_B, \tag{2.4}$$

$$N^T\bar{y} + \bar{z}_N = c_N, \quad \bar{z}_B = 0, \tag{2.5}$$

$$\bar{z}_N \geq 0. \tag{2.6}$$

Then two different cases arise, depending on whether or not the system

$$Ux_B = \bar{b} \tag{2.7}$$

is compatible, or equivalently $\bar{b}_2$ vanishes.

The dual feasible solution $(\bar{y}, \bar{z})$ and LU factors along with $\bar{b}$ will be updated, iteration by iteration, until optimality is achieved. It is possible to keep L well conditioned during the initial factorization and subsequent Bartels–Golub-type updates. The package LUSOL is suitable for handling *rectangular* pseudobasis factorizations of this kind [10]. In particular, this package returns $\|\bar{b}_2\|_1$ and hence identifies the two cases, as discussed in the next two sections.

3. Optimality test.

An assumption throughout this section is that

$$\bar{b}_2 = 0. \tag{3.1}$$

It is clear that this case holds whenever $s < m$ and B is a deficient basis, but all discussions are also valid when $s = m$ (B is a normal basis).

LEMMA 3.1. *There exists a primal basic solution $\bar{x}$, defined by*

$$U\bar{x}_B = \bar{b}, \quad \bar{x}_N = 0. \tag{3.2}$$

Proof. Setting $x_N = 0$ in the system $Ax = b$ gives $Bx_B = b$, which along with (2.2), (2.3), and (3.1) leads to an upper-triangular and compatible system (2.7). Hence, (3.2) defines a primal basic solution. □

THEOREM 3.2. *The primal objective value at $\bar{x}$ is equal to the dual objective value at $(\bar{y}, \bar{z})$. Moreover, if $\bar{x}_B \geq 0$, then $\bar{x}$ and $(\bar{y}, \bar{z})$ are a pair of primal and dual optimal solutions.*

Proof. From Lemma 3.1 and (2.4), it follows that the primal and dual objective values are identical:

$$c^T\bar{x} = c_B^T\bar{x}_B = \bar{y}^TB\bar{x}_B = \bar{y}^Tb.$$

Further, it is clear that $\bar{x}$ and $\bar{z}$ exhibit complementary slackness. Therefore, if $\bar{x}_B \geq 0$ (and hence $\bar{x}$ is primal feasible), then $\bar{x}$ and $(\bar{y}, \bar{z})$ are a pair of primal and dual optimal solutions, as the latter is assumed to be dual feasible. □

According to Theorem 3.2, optimality is achieved and we are done if $\bar{x}_B \geq 0$. If $\bar{x}_B \ngeq 0$, a leaving index j_p is well defined such that

$$\bar{x}_{j_p} = \min\{\bar{x}_{j_i} \mid \bar{x}_{j_i} < 0,\ i = 1, \ldots, s\} < 0. \tag{3.3}$$

4. Search direction. An assumption throughout this section is that

$$\bar{b}_2 \neq 0. \tag{4.1}$$

By definition, this case holds whenever $s < m$ and the pseudobasis B is not a basis.

We show in this case that an uphill search direction $(\Delta y, \Delta z)$ in the dual space can be determined such that

$$B^T\Delta y = 0, \tag{4.2}$$

$$N^T\Delta y + \Delta z_N = 0, \quad \Delta z_B = 0, \tag{4.3}$$

$$b^T\Delta y > 0. \tag{4.4}$$

The key to satisfying the preceding equations lies in determining a suitable vector $\Delta y \neq 0$ in the null space of B^T at a low cost. Using (2.2) and the transformation

$$L^T\Delta y = \Delta y', \tag{4.5}$$

we convert (4.2) into $[U_1^T \quad 0^T]\Delta y' = 0$, which has infinitely many solutions of the form

$$\Delta y' \triangleq \begin{bmatrix}\Delta y_1' \\ \Delta y_2'\end{bmatrix} = \begin{bmatrix}0 \\ h\end{bmatrix}, \quad \begin{array}{c} s \\ m-s \end{array},$$

where $h \in R^{m-s}$ is any given vector. If we simply take $\Delta y' = [0^T \quad \bar{b}_2^T]^T$, it follows from (4.5) and (4.3) that

$$L^T\Delta y = \begin{bmatrix}0^T & \bar{b}_2^T\end{bmatrix}^T, \tag{4.6}$$

$$\Delta z_B = 0, \tag{4.7}$$

$$\Delta z_N = -N^T\Delta y. \tag{4.8}$$

We justify the eligibility of $(\Delta y, \Delta z)$ for being a search direction in the dual space.

LEMMA 4.1. $(\Delta y, \Delta z)$ *defined by* (4.6)–(4.8) *satisfies* (4.2)–(4.4).

Proof. Condition (4.3) holds clearly. Additionally, it follows from (2.2) and (4.6) that

$$B^T \Delta y = U^T L^T \Delta y = \begin{bmatrix} U_1^T & 0^T \end{bmatrix} \begin{bmatrix} 0^T & \bar{b}_2^T \end{bmatrix}^T = 0.$$

Thus, condition (4.2) holds. Further, from (2.3), (4.6), and (4.1), it follows that

$$b^T \Delta y = \bar{b}^T L^T \Delta y = \begin{bmatrix} \bar{b}_1^T & \bar{b}_2^T \end{bmatrix} \begin{bmatrix} 0^T & \bar{b}_2^T \end{bmatrix}^T = \bar{b}_2^T \bar{b}_2 > 0,$$

which completes the proof. □

Now consider the associated line search scheme in the dual space:

$$\hat{y} = \bar{y} + \alpha \Delta y, \tag{4.9}$$

$$\hat{z}_B = 0, \tag{4.10}$$

$$\hat{z}_N = \bar{z}_N + \alpha \Delta z_N. \tag{4.11}$$

LEMMA 4.2. $(\hat{y}, \hat{z})$ *in* (4.9)–(4.11) *with any given* α *satisfies*

$$B^T \hat{y} = c_B, \quad N^T \hat{y} + \hat{z}_N = c_N, \quad \hat{z}_B = 0.$$

Proof. The result can be easily derived from (4.2)–(4.3) (Lemma 4.1) and the dual feasibility assumptions (2.4)–(2.5). □

The preceding lemma says that $(\hat{y}, \hat{z})$ is a dual solution for any α, just like the base-point $(\bar{y}, \bar{z})$ at $\alpha = 0$. Nevertheless, it might not be feasible unless the value of α is restricted appropriately.

THEOREM 4.3. *If* $\Delta z_N \geq 0$, *then* $(\hat{y}, \hat{z})$ *in* (4.9)–(4.11) *is a dual feasible solution for any* $\alpha > 0$; *hence, the dual program* (1.2) *is unbounded above.*

Proof. By Lemma 4.2, $(\hat{y}, \hat{z})$ satisfies $A^T \hat{y} + \hat{z} = c$ for any α. Note that $\hat{z}_B = 0$ by (4.10). From (4.11) and the nonnegativity of $\bar{z}_N$ and Δz_N, it follows that $\hat{z}_N \geq 0$ for any $\alpha > 0$. That is, $(\hat{y}, \hat{z})$ is a dual feasible solution for any $\alpha > 0$. Moreover, by (4.9), the associated dual objective is

$$b^T \hat{y} = b^T \bar{y} + \alpha b^T \Delta y, \tag{4.12}$$

which goes to infinity with α, since (4.4) holds by Lemma 4.1. □

Note that Δz_N (and hence Δz) vanishes whenever Δy happens to be in the null space of N^T (even though $\Delta y \neq 0$ by (4.4)). By Theorem 4.3, this case implies dual unboundedness and hence primal infeasibility. However, it implies still more.

THEOREM 4.4. *If* $\Delta z_N = 0$, *then* $Ax = b$ *is inconsistent.*

Proof. Assume the opposite, i.e., $\Delta z_N = 0$ but $Ax = b$ is consistent. Then, it follows from (4.8) that

$$N^T \Delta y = 0 \tag{4.13}$$

and that there exists a solution, say $\hat{x}$, such that

$$b = B\hat{x}_B + N\hat{x}_N. \tag{4.14}$$

Transposing and then postmultiplying both sides of (4.14) by Δy leads to

$$b^T \Delta y = \hat{x}_B^T (B^T \Delta y) + \hat{x}_N^T (N^T \Delta y).$$

The preceding equation along with (4.2) and (4.13) gives $b^T \Delta y = 0$. However, this contradicts (4.4) by Lemma 4.1. Therefore, $Ax = b$ is inconsistent if $\Delta z_N = 0$. □

In the case of $\Delta z_N \not\geq 0$, we maximize the step length α subject to $\hat{z} \geq 0$ to achieve the largest possible dual objective value. This leads us to determine α and index k_q such that

$$\alpha = -\bar{z}_{k_q}/\Delta z_{k_q} = \min\{-\bar{z}_{k_j}/\Delta z_{k_j} \mid \Delta z_{k_j} < 0,\ j = 1, \ldots, n-s\} \geq 0. \tag{4.15}$$

As usual, a dual feasible solution $\bar{z}$ is said to be *dual degenerate* when some components of $\bar{z}_N$ are zero. Consequently, α determined by (4.15) could vanish; this is considered to be an undesirable case because the solution, given by (4.9)–(4.11), then coincides with its predecessor. If this is not the case, however, further progress is assured.

THEOREM 4.5. *If $\Delta z_N \not\geq 0$, then $(\hat{y}, \hat{z})$ in (4.9)–(4.11) is a dual feasible solution. It corresponds to a dual objective value strictly greater than before if dual nondegeneracy is assumed.*

Proof. Since $\Delta z_N \not\geq 0$, a step length $\alpha \geq 0$ is well defined by (4.15), and so is $(\hat{y}, \hat{z})$ by (4.9)–(4.11). As in the proof of Theorem 4.3, it can be shown that $(\hat{y}, \hat{z})$ is a dual feasible solution. Further, (4.15) and the dual nondegeneracy together guarantee $\alpha > 0$, which along with (4.12) and (4.4) leads to $b^T\hat{y} > b^T\bar{y}$. □

If we introduce an extra variable z_{n+1} (which might be called the dual objective variable) and add $b^T y - z_{n+1} = 0$ to the dual equality constraints, then the corresponding component of the augmented search direction is $\Delta z_{n+1} = b^T \Delta y$, and hence $\hat{z}_{n+1} = \bar{z}_{n+1} + \alpha \Delta z_{n+1}$. Thus, it is seen from (4.12) that the value of z_{n+1} gives the associated dual objective value. Such a variable might simplify the implementation.

5. Formulation of the algorithm.

Once a leaving or entering index has been selected, there remains the associated pseudobasis to be changed. In this respect, LUSOL might be the only package available for updating LU factors of a *rectangular* pseudobasis [10, 27]. In this section, we address some key points about updating and downdating and then describe the algorithm formally.

5.1. Downdating.

Let us return to the end of section 3. Assume that optimality is not attained, and hence a leaving index j_p has been selected by (3.3). Denote by $\hat{B}$ the matrix resulting from dropping a_{j_p} from B.

PROPOSITION 5.1. *The matrix $\hat{B}$ is a pseudobasis.*

Proof. The pseudobasis B has full column rank, and any subset of the columns of B constitutes a matrix of full column rank. □

We carry out the pseudobasis change by computing the LU factors of $\hat{B}$ from those of B. It is clear that $\hat{B} = LH$, where the upper-Hessenberg H is U with its pth column removed. As in Reid's implementation of the Bartels–Golub update [27], we interchange the pth and sth rows of H and then eliminate entries p through $(s-1)$ of the new sth row by a sequence of Gauss transformations and thus obtain the upper-triangular factor of $\hat{B}$. The factor L is easily updated in the product form.

Accordingly, $\bar{b}$ is updated by interchanging its pth and sth component and applying the same sequence of Gauss transformations.

To complete the pseudobasis change, we move j_p from the basic index set to the end of the nonbasic index set and set $s := s - 1$. Deleting a column of B in this manner is called *downdating*.

5.2. Updating. Assume that an entering index k_q has been chosen by (4.15). We append a_{k_q} to the end of B, obtaining $\tilde{B} = [B \quad a_{k_q}]$.

PROPOSITION 5.2. *The matrix $\tilde{B}$ is a pseudobasis.*

Proof. From (4.8) and (4.15), it follows that

$$-a_{k_q}^T \Delta y = \Delta z_{k_q} < 0. \tag{5.1}$$

Assume now that $\tilde{B}$ is not a pseudobasis. Then, there exists a nonzero vector $u = [u_B^T \quad u_{k_q}]^T$ such that

$$\tilde{B}u = Bu_B + u_{k_q} a_{k_q} = 0. \tag{5.2}$$

Further, it holds that

$$u_{k_q} \neq 0, \tag{5.3}$$

because otherwise B would not have full column rank, contradicting the fact that B is a pseudobasis. Transposing and then postmultiplying both sides of (5.2) by Δy lead to

$$u_B^T B^T \Delta y + u_{k_q} a_{k_q}^T \Delta y = 0,$$

which along with (4.2) and (5.3) gives $a_{k_q}^T \Delta y = 0$. However, this contradicts (5.1). Therefore, $\tilde{B}$ is a pseudobasis. □

The pseudobasis change is made by computing the LU factors of $\tilde{B}$ from those of B. To this end, we solve the triangular system

$$La = a_{k_q}. \tag{5.4}$$

If its solution is partitioned as $a = [a_1^T \quad a_2^T]^T$, we have from (2.2) and (5.4) that

$$\tilde{B} = \begin{bmatrix} B & a_{k_q} \end{bmatrix} = L \begin{bmatrix} U_1 & a_1 \\ 0 & a_2 \end{bmatrix}, \quad \begin{matrix} s \\ m-s \end{matrix}, \tag{5.5}$$

where Proposition 5.2 and the nonsingularity of L together imply that $a_2 \neq 0$. The right-hand factor in (5.5) is upper-triangular except perhaps for its last column. Moving the largest entry of a_2 to the diagonal position and eliminating entries below the diagonal by a Gauss transformation turn it into the upper-triangular factor of $\tilde{B}$. The factor L is easily updated accordingly.

The vector $\bar{b}$ is updated by applying the same Gauss transformation to it. This is an inexpensive operation.

Finally, we move k_q from the nonbasic index set to the end of the basic index set and set $s := s + 1$. Adding a column to B in this way is called an *updating*.

Let us see what will happen after an updating or downdating.

PROPOSITION 5.3. *If the dual program is bounded above, any downdating is followed by an updating.*

Proof. Let $\hat{B}$ result from a downdating. Without loss of generality, assume that $\hat{B}$'s predecessor is $B = [\hat{B} \quad a_{j_p}]$. We only need to show that b is not included in the range of $\hat{B}$, and hence $\bar{b}_2$ is nonzero after the downdating. Assume the opposite that some $x_{\hat{B}}$ satisfies $\hat{B}x_{\hat{B}} = b$. Then $[x_{\hat{B}}^T \quad 0]^T$ solves $Bx_B = b$. From the uniqueness of the solution of the latter equation (B is of full column rank), it follows that $[x_{\hat{B}}^T \quad 0]^T$

is the solution of (3.2). This implies $\bar{x}_{j_p} = 0$, which contradicts $\bar{x}_{j_p} < 0$ (see (3.3)). Therefore, b is not included in the range of $\hat{B}$, and the next pseudobasis change is an updating if dual unboundedness is not detected. □

A downdating and the following updating are said to be *matched*. On the other hand, it is uncertain what happens after an updating. It could be either a downdating or an updating.

5.3. Algorithm. The overall steps can be organized as follows.

Algorithm 1 (RDPPA). Given the ordered index sets (2.1), the LU factorization (2.2) of the associated pseudobasis, and the transformed right-hand side (2.3), assume that an initial dual feasible solution $(\bar{y}, \bar{z})$ (with $\bar{z}_B = 0$) is available.

```
 1. If $s < m$ and $\bar{b}_2 \neq 0$, go to step 7.
 2. Solve $U\bar{x}_B = \bar{b}$ for $\bar{x}_B$ (3.2).
 3. Stop if $\bar{x}_B \geq 0$ (optimality achieved).
 4. Determine a leaving index $j_p$ by (3.3):
        $\bar{x}_{j_p} = \min\{\bar{x}_{j_i} \mid \bar{x}_{j_i} < 0,\ i = 1, \ldots, s\} < 0.$
 5. Update LU factors and $\bar{b}$ by the downdating associated with $j_p$.
 6. Move index $j_p$ from $B$ to the end of $N$ and set $s := s - 1$.
 7. Solve $L^T\Delta y = [0^T \;\; \bar{b}_2^T]^T$ for $\Delta y$ (4.6).
 8. Compute $\Delta z_N$ by $\Delta z_N = -N^T\Delta y$ (4.8).
 9. Stop if $\Delta z_N \geq 0$ (dual unbounded).
10. Determine an entering index $k_q$ and step length $\alpha$ by (4.15):
        $\alpha = -\bar{z}_{k_q}/\Delta z_{k_q} = \min\{-\bar{z}_{k_j}/\Delta z_{k_j} \mid \Delta z_{k_j} < 0,\ j = 1, \ldots, n - s\} \geq 0.$
11. Update $(\bar{y}, \bar{z})$: $\hat{y} = \bar{y} + \alpha\Delta y$, $\hat{z}_B = 0$, $\hat{z}_N = \bar{z}_N + \alpha\Delta z_N$ (4.9)–(4.11).
12. Update LU factors and $\bar{b}$ by the updating associated with $k_q$.
13. Move index $k_q$ from $N$ to the end of $B$ and set $s := s + 1$.
14. Go to step 1.
```

Steps 2–6 perform downdating operations, while steps 7–13 are related to updating. An iteration involving steps 7–13 is called an *updating iteration*, and one involving steps 2–13 is called a *full iteration*. It is clear that any full iteration does not change the number of columns of the pseudobasis, while an updating iteration increases it by 1. All iterations fall into one of these two categories.

Note that an updating iteration involves a triangular solve in step 7 and another in step 12 (for the solution of (5.4)). As an additional triangular system is solved in step 2, a full iteration involves three triangular solves, compared with the four in conventional algorithms. It is maintaining $\bar{b}$ that helps save one of the triangular solves with L.

Moreover, for small s (relative to m), the size of the $s \times s$ system $U_1 x_B = \bar{b}_1$ is small, compared with $m \times m$ systems solved in simplex algorithms. Most importantly, a higher dimension $m - s$ of the null space of B^T would contribute to the formation of a better search direction Δy in the y-space (see (4.2)). Therefore, Algorithm 1 appears to be particularly suitable for solving real-world LP problems, which are often degenerate or even highly degenerate. If s reaches m, on the other hand, its advantages could vanish because it would perform like the dual simplex algorithm. In view of this, a small initial pseudobasis seems to be favorable.

As in simplex contexts, the updated LU factors tend to become increasingly dense as columns enter and leave the pseudobasis. This is controlled by the periodic refactorization. As with other dual algorithms, if $n \gg s$, then the cost associated with Δz_N in step 8 is higher compared to its primal-simplex counterpart with partial pricing. A helpful remedy is suggested in section 8.2.

5.4. Properties of Algorithm 1. We first give the following main result associated with the proposed algorithm.

THEOREM 5.4. *Under the dual nondegeneracy for full iterations, Algorithm* 1 *terminates either at*

(1) *step* 3*, yielding a pair of primal and dual optimal solutions, or*

(2) *step* 9*, detecting dual unboundedness.*

Proof. It is clear that each (full or updating) iteration corresponds to a pseudobasis, and the number of columns in the pseudobasis never decreases in the solution process. Also note that each full iteration produces a primal solution $\bar{x}$ whose primal objective value is equal to the dual objective value of the dual iterates $(\bar{y}, \bar{z})$ (see Lemma 3.1 and Theorem 3.2).

Assume that the process does not terminate. Then, some pseudobases must appear infinitely many times because there are only finitely many. Moreover, such a cycling involves only full iterations because any updating iteration would increase the number of columns in the pseudobasis. Under dual nondegeneracy for full iterations, the dual objective value increases strictly in the cycling, by Theorem 4.5. However, this is clearly a contradiction. Therefore, the algorithm terminates. Termination at step 3 produces a pair of primal and dual optimal solutions, according to Theorem 3.2, while termination at step 9 detects dual unboundedness, according to Theorem 4.3. □

COROLLARY 5.5. *Termination at step* 9 *also indicates primal infeasibility of the program. If* $\Delta z_N = 0$*, moreover, it reveals inconsistency of* $Ax = b$*.*

Proof. The first half of the corollary is easily derived from Theorem 5.4 and the well-known weak duality theorem. The other half is from Theorem 4.4. □

In the preceding corollary, dual nondegeneracy is assumed to prevent cycling in the solution process. Of course, such an assumption is entirely unrealistic—in fact, both primal and dual degeneracy occur all the time. As in conventional contexts, however, experimental results suggest that cycling rarely happens, if at all (see section 8). Therefore, Algorithm 1 should be regarded as finite in practice.

Although it has not been possible to rule out the possibility of cycling in the presence of dual degeneracy, we still have the following desirable result.

PROPOSITION 5.6. *An entering index from an updating never leaves immediately; a leaving index from a downdating never enters immediately.*

Proof. We only show the first half of the proposition, as the other half can be shown similarly. Assume that k_q is an entering index and $\tilde{B} = [B \;\; a_{k_q}]$ is the resulting pseudobasis. If the following is another updating, then no column leaves $\tilde{B}$. So, assume that the following is a downdating. Then $\tilde{B}$ is a *basis*, and hence there exists a vector, say $\bar{x} = [\bar{x}_B^T \;\; \bar{x}_{k_q}]^T$, such that

$$B\bar{x}_B + \bar{x}_{k_q} a_{k_q} = b. \tag{5.6}$$

Transposing and then postmultiplying both sides of (5.6) by Δy give

$$\bar{x}_B^T B^T \Delta y + \bar{x}_{k_q} a_{k_q}^T \Delta y = b^T \Delta y.$$

The preceding equation along with (4.2) and (4.4) (Lemma 4.1) gives $\bar{x}_{k_q}(a_{k_q}^T \Delta y) > 0$, which with (5.1) leads to $\bar{x}_{k_q} > 0$. In view of (3.3), therefore, we assert that k_q never leaves immediately. □

6. Motivation and alternative presentation.

The new algorithm may be derived from two diverging ideas. One is where the present work comes from and the other is due to Saunders [28]. We address them separately in this section.

6.1. Motivation.

It might be accepted that crucial to an LP solver's success is the quality of the search direction used and the complexity of its computation. An "ideal" search direction should be the *orthogonal* projection of b onto the null space of B^T, as it is the *steepest* uphill direction in the space, with respect to the dual objective. Indeed, such a simple idea has achieved great success in practice (see, e.g., [8]).

Along this line, we have solved dense LP problems by handling a sequence of least-squares problems using the QR factorization [20, 24]. Favorable computational results motivate the derivation of a sparse and practical approximation in this paper using the LU factorization (2.2). As a result, the key search direction Δy defined by (4.6) is, in general, no longer the orthogonal projection of b onto the null space of B^T. However, $\Delta y' = [0^T \;\; \bar{b}_2^T]^T$ is the orthogonal projection of the transformed right-hand side $\bar{b}$ onto the null space of U^T, the transpose of the transformed basis, since it equals the residual at the unique solution of the least-squares problem

$$\min_{x_B} \|\bar{b} - Ux_B\|_2. \tag{6.1}$$

As Δy differs from $\Delta y'$ only by a matrix factor L^T, it could be viewed as an oblique projection of b onto the null space of B^T, which is why the proposed algorithm is still described as *projective*, like its dense version. Clearly, Δy would be the orthogonal projection if L were orthogonal. Even though it is not orthogonal, L is well conditioned, as mentioned in section 2.

6.2. Reduced-gradient representation.

An alternative derivation is possible by following Saunders's penetrating view of Algorithm 1 as a reduced-gradient implementation of a normal active-set method for solving the dual linear program [28]. Indeed, the proposed algorithm could be described in terms of active sets such as conventional simplex variants (e.g., see [7]). Nevertheless, it might be far more important that the algorithm turns out to be of a reduced-gradient nature, as explained next.

In the LU factorization of B (2.2), assume that

$$PBQ = \begin{array}{cc} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} & \begin{array}{c} s \\ m-s \end{array} \end{array} \tag{6.2}$$

with B_1 nonsingular. Then the following matrix is a basis for the null space of B^T:

$$Z = P^T \begin{bmatrix} -B_1^{-T} B_2^T \\ I \end{bmatrix}. \tag{6.3}$$

(It is trivial to prove that Z is nonsingular and $B^T Z = 0$.) The following theorem reveals the relation between Z and the key quantities used in the proposed algorithm.

THEOREM 6.1. *Assume that Z is defined by* (6.3) *with* (6.2). *For $\bar{b}$ defined by* (2.3) *and Δy defined by* (4.6), *it holds that*

$$\bar{b}_2 = Z^T b, \tag{6.4}$$

$$\Delta y = PZZ^T b. \tag{6.5}$$

Proof. By (2.2) and (6.2), we have $B_1 = L_1U_1$ and $B_2 = L_2U_1$, which implies

$$B_2B_1^{-1} = L_2L_1^{-1}. \tag{6.6}$$

Also, from (6.3), (2.3), and (2.2) it follows that

$$Z^Tb = [-B_2B_1^{-1} \quad I]Pb = [-B_2B_1^{-1} \quad I]\, L\, \bar{b} = [-B_2B_1^{-1}L_1 + L_2 \quad I][\bar{b}_1^T \quad \bar{b}_2^T]^T,$$

which with (6.6) proves (6.4). Further, by (6.4), (6.3), (2.2), and (6.6), we have

$$L^TPZZ^Tb = L^T(PZ)\bar{b}_2 = L^T[-B_2B_1^{-1} \quad I]^T\bar{b}_2 = [0 \quad I]^T\bar{b}_2 = [0^T \quad \bar{b}_2^T]^T,$$

implying that PZZ^Tb solves (4.6); thus, (6.5) holds and the proof is complete. □

Theorem 6.1 says that the key quantity $\bar{b}_2$ is equal to the reduced-gradient Z^Tb, and the dual search direction Δy is equal to PZZ^Tb (or ZZ^Tb if P is assumed to be the identity permutation, as in previous sections). It is, therefore, clear that all of the linear algebra associated with the proposed algorithm can be described in terms of Z or B_1 and B_2. Consequently, we would be led to a reduced-gradient alternative, using the LU factorization of just B_1, not all of B.

Algorithms that are equivalent theoretically could perform very differently in practice. Indeed, Saunders's alternative implementation of the proposed algorithm deserves further investigation.

7. Dual phase-1. To get Algorithm 1 started, a dual phase-1 procedure is needed to produce an initial dual feasible solution. Several approaches are available for this purpose (see [20, 24]). In this section, we briefly present an auxiliary problem similar to that described in [24], and show how to use it to fit our needs.

Let B be any given pseudobasis and let N be the associated pseudononbasis. Assume that $[0^T \quad \bar{c}_N^T]^T$ is the corresponding reduced cost and that $[0^T \quad g_N^T]^T \geq 0$ is any given n-vector. Introducing an artificial variable x_{n+1}, we construct the following auxiliary program:

$$\begin{aligned} &\text{minimize} \quad \bar{c}_N^T x_N, \\ &\text{subject to} \quad \begin{bmatrix} B & N & 0 \\ 0 & (\bar{c}_N - g_N)^T & -1 \end{bmatrix} \begin{bmatrix} x_B \\ x_N \\ x_{n+1} \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad x,\ x_{n+1} \geq 0. \end{aligned} \tag{7.1}$$

Now consider the associated dual problem

$$\begin{aligned} &\text{maximize} \quad -y_{m+1}, \\ &\text{subject to} \quad \begin{bmatrix} B^T & 0 \\ N^T & \bar{c}_N - g_N \\ 0 & -1 \end{bmatrix} \begin{bmatrix} y \\ y_{m+1} \end{bmatrix} + \begin{bmatrix} z_B \\ z_N \\ z_{n+1} \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{c}_N \\ 0 \end{bmatrix}, \quad z,\ z_{n+1} \geq 0. \end{aligned} \tag{7.2}$$

A feasible solution to (7.2) is readily available, namely,

$$\begin{bmatrix} \bar{y} \\ \bar{y}_{m+1} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad \begin{bmatrix} \bar{z}_B \\ \bar{z}_N \\ \bar{z}_{n+1} \end{bmatrix} = \begin{bmatrix} 0 \\ g_N \\ 1 \end{bmatrix}. \tag{7.3}$$

Thereby, one can get Algorithm 1 started to solve (7.2). There are infinitely many choices of g_N resulting in a variety of initial feasible solutions to (7.2). Among them, it might be preferable to choose the one with all g_N's components set to 1.

The following result, the proof of which is omitted, clarifies the use of (7.2).

THEOREM 7.1. *The dual program* (7.2) *has an optimal solution with value of no more than zero. If the optimal value is less than zero, then the original dual program* (1.2) *has no feasible solution; otherwise, the* (y, z) *part of the optimal solution gives a feasible solution to* (1.2).

Thus, an optimal solution to (7.2) provides a feasible solution to (1.2).

8. Computational results.

Numerical experiments were performed to gain an insight into the behavior of the proposed algorithm. In this section, we report results obtained and make final remarks.

8.1. Test codes.

Our code, named RDPPA 1.10, was coded in Fortran 77. It consisted of two phases: phase-2 was based on Algorithm 1; phase-1 solved the auxiliary program (7.2), using Algorithm 1 and starting with the dual feasible solution (7.3) with $g_N = [1, \dots, 1]^T$.

We decided to use MINOS for making a comparison, as it is accepted by the community to be a good benchmark for such purposes. In fact, code RDPPA 1.10 was developed using MINOS 5.3 as a platform. Consequently, the two codes shared such features as preprocessing, scaling, LUSOL [10], etc. Only the Mi50lp and Mi25bfac modules were replaced by programs written by the author. Very limited changes were made to other parts. Subroutine M2crsh in MINOS 5.3 searches for a (permuted) triangular initial basis. Since the required input of the new algorithm is only a pseudobasis, M2crsh was modified by deleting its last lines filling up gaps with logical columns with the limitation of the number s of initial basic columns not to exceed m by 80%. In addition, as Harris's two-pass ratio test [12] was used to select a leaving index in MINOS 5.3, we incorporated the same in RDPPA 1.10 to select an entering index. In a word, we made every endeavor to ensure a fair competition between the primal simplex algorithm and the proposed algorithm.

We made a comparison with MINOS 5.3 in an earlier version of this paper. Later, when Saunders kindly provided us with the latest version of MINOS, we were able to test and compare with MINOS 5.51 [15]. We now only report our computational results obtained with the latter, although those with the former were much more favorable.

Like MINOS 5.51, the new code carried out an LU refactorization whenever LU factors were updated more than 98 times, or more than 19 times, but the sum of nonzeros of L and U exceeded two times that of the fresh factors from the previous refactorization. MINOS 5.51 worked with the default threshold pivoting tolerances $\tau_F = 100$ for factorization and $\tau_U = 10$ for updating. The large value of τ_F favors sparsity over stability whenever the basis is refactorized. (It allows the subdiagonals of L to be as large as 100.) In RDPPA 1.10, there is a reason to keep L better conditioned in order to improve the choice of B_1 from the *rectangular* pseudobasis (6.2). We, therefore, set $\tau_F = \tau_U = 10$. Even smaller values such as $\tau_F = 5$ or 2.5 may be desirable [11, section 4.5]. They would improve the condition of B_1 and probably the quality of the search directions at the expense of slightly denser LU factors.

Compiled using Visual Fortran 5.0, both MINOS 5.51 and RDPPA 1.10 were run under the Windows 98 system on a Pentium III 550E personal computer with 256 MB of memory and about 16 digits of precision. In RDPPA 1.10, both the primal and dual feasibility tolerances were taken to be 10^{-6}, and $\|\bar{b}_2\|_1 > 10^{-6}$ was used in place of $\bar{b}_2 \neq 0$. All the reported CPU times were measured in seconds with utility routine CPU_TIME, excluding the time spent on preprocessing and scaling.

In running both MINOS 5.51 and RDPPA 1.10, the usual default options were used, except for Rows 32000; Columns 250000; Elements 4000000; Iterations 1000000; Scale yes; Solution no; Log Frequency 0; Print Level 0.

8.2. Partial pricing tactics. Since Partial Price 10 is the default in MINOS 5.51 for LPs, partial pricing p was invoked with $p = 10$ for moderate-sized problems with $n < \max(1000, 4m)$; for the other problems, partial pricing was carried out with $p = n/2 \times m$, where m and n are the number of rows and columns, respectively [15].

Unfortunately, there is no corresponding partial pricing for dual algorithms. As a remedy, the following tactic was taken in RDPPA 1.10. Define the index set as

$$J = \{k_j \mid \Delta z_{k_j} < -\epsilon_1,\ \bar{z}_{k_j} < \epsilon_0,\ j = 1, \ldots, n - s\},$$

where $\epsilon_1 \approx 10^{-11}$ and $\epsilon_0 \approx 10^{-13}$. Before the two-pass ratio test, a procedure was inserted for choosing an entering index k_q such that

$$\Delta z_{k_q} = \min \{\Delta z_{k_j} \mid k_j \in J\};$$

hence, the ratio test was carried out only when the index set J was empty. When J was nonempty, only degenerate components of Δz_{k_j}, that is, those indexed by elements in the set $\{k_j \mid \bar{z}_{k_j} < \epsilon_0,\ j = 1, \ldots, n - s\}$ needed to be computed. Viewed as a kind of "partial pricing," this tactic turned out to be efficient in practice.

8.3. Results for set 1. All of our test problems were standard LP problems that do not have Bounds and Ranges sections in their MPS files, since our current code cannot handle such problems implicitly [9]. Our test set 1 included 50 problems from Netlib.[1] In fact, they were all of the Netlib problems for which $m + n \leq 10000$. Of the largest 4 Netlib problems for which $m + n > 10000$, MAROS-R7 and STOCFOR3 were left for test set 2, and QAP12 and QAP15 were not included in our tests because they are too time-consuming to solve for both MINOS 5.51 and RDPPA 1.10.

Numerical results obtained with set 1 are displayed in Tables 8.1 and 8.2, in the order of increasing sum $m + n$ before slack variables are added. In these tables, the total iterations and time required for solving each problem are listed in the columns labeled Itns and Time; percentages of total degenerate iterations are given in the columns labeled % Degen. We point out that the column labeled Itns in Table 8.2 lists *full* iteration counts, because for each run of the new code, all updating iterations should be considered together with the pseudobasis factorization and refactorizations. Final objective values reached are not listed, as they are the same as those given in the Netlib index file. Table 8.3 compares the performance of the two codes by giving iteration and time ratios of MINOS 5.51 to RDPPA 1.10 for each problem.

These results are summarized in Table 8.4, where the 50 problems are categorized into three groups: group Small includes the first 20 problems (from AFIRO to SCTAP1), Medium includes the next 15 problems (from SCFXM1 to SHIP04L), and Large includes the last 15 problems (from QAP8 to TRUSS). The bottom four lines of Table 8.4 may serve as an overall comparison between the two codes. From the bottom line labeled Total there, it is seen that the total iteration and total time ratios are 1.37 and 1.24, respectively. Therefore, RDPPA 1.10 outperformed MINOS 5.51 with set 1, although the small time ratio (relative to the iteration ratio) reveals that the computational effort per iteration for RDPPA 1.10 was overall greater than that for MINOS 5.51 (using partial pricing).

[1] http://www.netlib.org/lp/data/

TABLE 8.1
MINOS 5.51 *statistics for set* 1 *of* 50 *Netlib problems.*

Problem	m	n	$m+n$	Itns	Time	% Degen
AFIRO	27	32	60	9	0.11	33.3
SC50B	50	48	99	16	0.11	68.8
SC50A	50	48	99	14	0.16	50.0
ADLITTLE	56	97	154	105	0.11	1.9
BLEND	74	83	158	78	0.16	35.9
SHARE2B	96	79	176	97	0.10	7.2
SC105	105	103	209	27	0.16	48.1
STOCFOR1	117	111	229	86	0.17	24.4
SCAGR7	129	140	270	79	0.11	21.5
ISRAEL	174	142	317	273	0.28	1.8
SHARE1B	117	225	343	189	0.22	0.5
SC205	205	203	409	52	0.17	40.4
BEACONFD	173	262	436	45	0.22	17.8
LOTFI	153	308	462	248	0.16	10.5
BRANDY	220	249	470	443	0.44	7.9
E226	223	282	506	468	0.43	14.3
AGG	488	163	652	117	0.28	23.1
SCORPION	357	358	716	181	0.27	45.9
BANDM	305	472	778	476	0.55	8.6
SCTAP1	300	480	781	311	0.32	21.5
SCFXM1	330	457	788	410	0.44	11.5
AGG2	516	302	819	153	0.38	7.2
AGG3	516	302	819	169	0.39	10.1
SCSD1	77	760	838	443	0.27	61.4
SCAGR25	471	500	972	489	0.60	13.9
DEGEN2	442	534	977	837	1.05	53.6
FFFFF800	524	854	1379	488	0.82	20.3
SCSD6	147	1350	1498	1028	0.72	50.2
SCFXM2	660	914	1575	729	1.15	13.7
SCRS8	490	1169	1660	737	0.88	29.3
BNL1	643	1175	1819	1236	1.70	11.0
SHIP04S	402	1458	1861	169	0.33	14.8
SCFXM3	990	1371	2362	1100	2.15	12.9
25FV47	821	1571	2393	7129	15.16	9.7
SHIP04L	402	2118	2521	262	0.50	14.5
QAP8	912	1632	2545	11055	52.90	33.4
WOOD1P	244	2594	2839	816	3.95	43.9
SCTAP2	1090	1880	2971	767	1.65	51.9
SCSD8	397	2750	3148	2779	3.51	42.8
SHIP08S	778	2387	3166	256	0.77	21.1
DEGEN3	1503	1818	3322	7301	36.36	54.7
SHIP12S	1151	2763	3915	414	1.21	20.3
SCTAP3	1480	2480	3961	915	2.52	54.9
STOCFOR2	2157	2031	4189	1922	7.14	41.4
SHIP08L	778	4283	5062	438	1.27	14.4
BNL2	2324	3489	5814	4682	22.08	16.5
SHIP12L	1151	5427	6579	831	2.52	24.1
D2Q06C	2171	5167	7339	46638	303.51	8.9
WOODW	1098	8405	9504	3710	14.12	38.2
TRUSS	1000	8806	9807	13253	46.03	37.0

TABLE 8.2
RDPPA 1.10 *statistics for set* 1 *of* 50 *Netlib problems.*

Problem	m	Itns	Time	% Degen	As/m (%)	Fs/m (%)
AFIRO	27	5	0.05	60.0	91.4	96.4
SC50B	50	7	0.11	42.9	98.0	98.0
SC50A	50	12	0.00	33.3	96.1	98.0
ADLITTLE	56	138	0.06	15.9	96.2	98.2
BLEND	74	32	0.05	6.3	96.7	98.7
SHARE2B	96	114	0.11	23.7	95.9	99.0
SC105	105	29	0.11	24.1	98.0	99.1
STOCFOR1	117	67	0.06	10.4	98.3	99.2
SCAGR7	129	97	0.11	1.0	95.3	99.2
ISRAEL	174	506	0.33	35.6	98.8	99.4
SHARE1B	117	194	0.11	5.7	97.6	99.2
SC205	205	44	0.06	4.5	98.7	99.5
BEACONFD	173	140	0.11	2.9	88.0	89.1
LOTFI	153	143	0.11	69.2	96.2	99.4
BRANDY	220	237	0.16	2.1	80.5	82.4
E226	223	469	0.33	2.6	97.5	99.1
AGG	488	328	0.28	19.5	99.5	99.8
SCORPION	357	29	0.16	0.0	98.4	99.4
BANDM	305	471	0.44	1.5	98.1	99.7
SCTAP1	300	176	0.16	85.2	96.4	98.7
SCFXM1	330	314	0.27	15.9	95.6	99.7
AGG2	516	246	0.27	28.9	99.4	99.8
AGG3	516	263	0.33	27.8	99.5	99.8
SCSD1	77	333	0.16	66.4	98.7	98.7
SCAGR25	471	301	0.27	1.0	94.1	97.0
DEGEN2	442	643	0.88	0.8	97.6	99.8
FFFFF800	524	329	0.38	43.2	98.0	98.5
SCSD6	147	1079	0.60	75.3	99.3	99.3
SCFXM2	660	622	0.94	15.8	97.1	99.7
SCRS8	490	430	0.55	14.9	98.7	99.6
BNL1	643	1205	1.48	41.5	96.0	98.9
SHIP04S	402	340	0.39	10.9	84.7	88.6
SCFXM3	990	981	1.98	15.6	96.9	99.7
25FV47	821	6545	15.87	11.0	99.3	99.8
SHIP04L	402	393	0.66	7.6	85.5	88.6
QAP8	912	10273	41.30	28.9	81.3	81.3
WOOD1P	244	984	7.30	51.4	89.0	99.2
SCTAP2	1090	379	0.55	91.8	93.7	96.8
SCSD8	397	4324	5.44	58.1	99.7	99.7
SHIP08S	778	390	0.72	0.8	81.2	81.8
DEGEN3	1503	4245	22.95	3.0	98.9	99.8
SHIP12S	1151	624	1.53	5.9	86.4	88.7
SCTAP3	1480	429	0.83	90.2	92.5	96.2
STOCFOR2	2157	1757	5.22	3.0	98.9	100.0
SHIP08L	778	816	2.64	3.9	81.3	81.8
BNL2	2324	2420	10.76	35.3	91.1	96.6
SHIP12L	1151	1437	6.86	9.9	86.7	88.7
D2Q06C	2171	25948	198.94	14.9	99.5	100.0
WOODW	1098	6378	46.08	55.5	99.8	99.9
TRUSS	1000	6063	47.62	18.9	99.9	99.9

Table 8.3
Ratios of MINOS 5.51 *to RDPPA* 1.10 *for set* 1.

Problem	m	n	Itns	Time	% Degen
AFIRO	27	32	1.80	2.20	0.56
SC50B	50	48	2.29	1.00	1.60
SC50A	50	48	1.17	–	1.50
ADLITTLE	56	97	0.76	1.83	0.12
BLEND	74	83	2.44	3.20	5.70
SHARE2B	96	79	0.85	0.91	0.30
SC105	105	103	0.93	1.45	2.00
STOCFOR1	117	111	1.28	2.83	2.35
SCAGR7	129	140	0.81	1.00	21.50
ISRAEL	174	142	0.54	0.85	0.05
SHARE1B	117	225	0.97	2.00	0.09
SC205	205	203	1.18	2.83	8.98
BEACONFD	173	262	0.32	2.00	6.14
LOTFI	153	308	1.73	1.45	0.15
BRANDY	220	249	1.87	2.75	3.76
E226	223	282	1.00	1.30	5.50
AGG	488	163	0.36	1.00	1.18
SCORPION	357	358	6.24	1.69	–
BANDM	305	472	1.01	1.25	5.73
SCTAP1	300	480	1.77	2.00	0.25
SCFXM1	330	457	1.31	1.63	0.72
AGG2	516	302	0.62	1.41	0.25
AGG3	516	302	0.64	1.18	0.36
SCSD1	77	760	1.33	1.69	0.92
SCAGR25	471	500	1.62	2.22	13.90
DEGEN2	442	534	1.30	1.19	67.00
FFFFF800	524	854	1.48	2.16	0.47
SCSD6	147	1350	0.95	1.20	0.67
SCFXM2	660	914	1.17	1.22	0.87
SCRS8	490	1169	1.71	1.60	1.97
BNL1	643	1175	1.03	1.15	0.27
SHIP04S	402	1458	0.50	0.85	1.36
SCFXM3	990	1371	1.12	1.09	0.83
25FV47	821	1571	1.09	0.96	0.88
SHIP04L	402	2118	0.67	0.76	1.91
QAP8	912	1632	1.08	1.28	1.16
WOOD1P	244	2594	0.83	0.54	0.85
SCTAP2	1090	1880	2.02	3.00	0.57
SCSD8	397	2750	0.64	0.65	0.74
SHIP08S	778	2387	0.66	1.07	26.38
DEGEN3	1503	1818	1.72	1.58	18.23
SHIP12S	1151	2763	0.66	0.79	3.44
SCTAP3	1480	2480	2.13	3.04	0.61
STOCFOR2	2157	2031	1.09	1.37	13.80
SHIP08L	778	4283	0.54	0.48	3.69
BNL2	2324	3489	1.93	2.05	0.47
SHIP12L	1151	5427	0.58	0.37	2.43
D2Q06C	2171	5167	1.80	1.53	0.60
WOODW	1098	8405	0.58	0.31	0.69
TRUSS	1000	8806	2.19	0.97	1.96

Table 8.4
Summary for set 1.

	Problem	Itns	Time	% Degen
For MINOS 5.51	Small (20)	3314	4.53	483.4
	Medium (15)	15379	26.54	334.1
	Large (15)	95777	499.54	503.5
	Total	114470	530.61	1321.0
For RDPPA 1.10	Small (20)	3238	2.91	446.4
	Medium (15)	14024	25.03	376.6
	Large (15)	66467	398.74	471.5
	Total	83729	426.68	1294.5
Ratios of M.5.51 to R.1.10	Small (20)	1.02	1.56	1.08
	Medium (15)	1.10	1.06	0.89
	Large (15)	1.44	1.25	1.07
	Total	1.37	1.24	1.02

8.4. Results for set 2. In order to see how the codes perform as the problem size increases, our test set 2 included 15 test problems larger than those in the first set; $m + n > 10000$ holds for most problems in set 2.

The associated numerical results obtained with MINOS 5.51 and RDPPA 1.10 are listed in Tables 8.5 and 8.6, respectively, where the first eight problems (from CRE-C to OSA-60) are from Kennington,[2] the following five problems (from RAT7A to DBIR2) from BPMPD,[3] and the last two problems from Netlib. In fact, all Kennington and BPMPD problems were included that do not have Bounds and Ranges sections in their MPS files and that are more than 500 kB in a compressed form.

Table 8.4 gives iteration and time ratios of MINOS 5.51 to RDPPA 1.10 for each problem in set 2. It can be seen from Table 8.4 that both the total iteration ratio 4.58 and the total time ratio 2.14 are even higher than those associated with set 1; this is also the case with either Kennington or BPMPD group alone. In short, the new code outperformed MINOS 5.51 remarkably on set 2, although the relatively small time ratio indicates that the computational effort per iteration for RDPPA 1.10 was greater than that for MINOS 5.51 (with partial pricing), as in the case of set 1.

We do not list numerical results associated with phase-1, but only offer that the phase-1 iteration and phase-1 time ratios are 2.46 and 1.47 for set 1, and those associated with set 2 are 6.44 and 2.16. Thus, our dual phase-1 worked quite well.

8.5. Effects of degeneracy. To show how large the pseudobases used in the new code were, relative to normal bases, the columns labeled As/m (%) and Fs/m (%) (in Tables 8.2 and 8.6) give average and final s/m percentages, respectively. It is seen from the columns labeled Fs/m (%) that RDPPA 1.10 terminated at a deficient basis for all the test problems, except for two problems from set 1 and four problems from set 2. From Table 8.9, which gives total average s/m (%) and total final s/m (%) (total s to total m), it is seen that these percentages are not low: they are around 95%, roughly speaking. This is not very surprising though if we recall that the initial pseudobases could have high s/m approaching 80% (see the second paragraph of section 8.1). In general, high primal degeneracy should lead to low ratio s/m. To exploit primal degeneracy to a large extent, it might be favorable to have a low initial ratio s/m and some effective tactic to limit subsequent fill-in in the LU factors.

[2] http://www-fp.mcs.anl.gov/otc/Guide/TestProblems/LPtest/

[3] http://www.sztaki.hu/~meszaros/bpmpd/

Table 8.5
MINOS 5.51 *statistics for set* 2 *of* 15 *test problems.*

Problem	m	n	$m+n$	Itns	Time	% Degen
CRE-C	3068	3678	6747	4252	25.32	34.5
CRE-A	3516	4067	7584	3642	24.94	32.7
OSA-07	1118	23949	25068	1919	14.22	2.3
OSA-14	2337	52460	54798	4115	52.29	1.1
CRE-D	8926	69980	78907	295538	6633.90	41.0
CRE-B	9648	72447	82096	188706	4708.10	43.6
OSA-30	4350	100024	104375	8141	192.90	0.5
OSA-60	10280	232966	243247	17098	909.12	0.4
RAT7A	3136	9408	12545	3801	844.20	0.2
NSCT1	22901	14981	37883	2151	127.81	6.0
NSCT2	23003	14981	37985	12298	643.78	44.3
DBIR1	18804	27355	46160	1710	113.53	5.4
DBIR2	18906	27355	46262	53839	2874.96	67.0
MAROS-R7	3136	9408	12545	2520	74.04	0.0
STOCFOR3	16675	15695	32371	14189	532.89	42.2

Table 8.6
RDPPA 1.10 *statistics for set* 2 *of* 15 *test problems.*

Problem	m	Itns	Time	% Degen	As/m (%)	Fs/m (%)
CRE-C	3068	1909	8.73	68.9	89.9	94.0
CRE-A	3516	2019	11.04	68.8	91.7	94.6
OSA-07	1118	750	16.32	40.0	98.1	99.5
OSA-14	2337	2061	80.14	52.4	99.2	99.9
CRE-D	8926	39897	1510.45	89.2	72.4	72.4
CRE-B	9648	35547	1535.11	84.1	74.9	75.0
OSA-30	4350	3353	233.87	60.1	99.6	99.9
OSA-60	10280	6687	1046.44	62.8	99.8	100.0
RAT7A	3136	2531	175.87	0.0	100.0	100.0
NSCT1	22901	3753	520.91	29.9	98.7	98.7
NSCT2	23003	3872	502.67	13.1	95.6	98.0
DBIR1	18804	6463	1011.17	75.2	99.4	99.4
DBIR2	18906	7355	1060.33	15.0	97.6	99.1
MAROS-R7	3136	2345	62.01	0.0	100.0	100.0
STOCFOR3	16675	15392	538.33	23.4	99.1	100.0

Table 8.7
Ratios of MINOS 5.51 *to RDPPA* 1.10 *for set* 2.

Problem	m	n	Itns	Time	% Degen
CRE-C	3068	3678	2.23	2.90	0.50
CRE-A	3516	4067	1.80	2.26	0.48
OSA-07	1118	23949	2.56	0.87	0.06
OSA-14	2337	52460	2.00	0.65	0.02
CRE-D	8926	69980	7.41	4.39	0.46
CRE-B	9648	72447	5.31	3.07	0.52
OSA-30	4350	100024	2.43	0.82	0.01
OSA-60	10280	232966	2.56	0.87	0.01
RAT7A	3136	9408	1.50	4.80	–
NSCT1	22901	14981	0.57	0.25	0.20
NSCT2	23003	14981	3.18	1.28	3.38
DBIR1	18804	27355	0.26	0.11	0.07
DBIR2	18906	27355	7.32	2.71	4.47
MAROS-R7	3136	9408	1.07	1.19	–
STOCFOR3	16675	15695	0.92	0.99	1.80

Table 8.8
Summary for set 2.

	Problem	Itns	Time	% Degen
For MINOS 5.51	Kennington (8)	523411	12560.79	156.1
	BPMPD (5)	73799	4604.28	122.9
	Total	613919	17772.00	321.2
For RRDPPA 1.10	Kennington (8)	92223	4442.10	526.3
	BPMPD (5)	23974	3270.95	133.2
	Total	133934	8313.39	682.9
Ratios of M.5.51 to R.1.10	Kennington (8)	5.68	2.83	0.30
	BPMPD (5)	3.08	1.41	0.92
	Total	4.58	2.14	0.47

Table 8.9
Total s/m (%).

Problem	Average	Final
Set 1	94.5	95.4
Set 2	94.7	96.6

On the other hand, from the bottom row labeled Total and the column labeled % Degen in Table 8.4, it is seen that the ratio of percentages of total degenerate iterations is 1.02. Thus, the overall effects of degeneracy are about the same for the two codes with test set 1.

Interestingly enough, for set 2 the situation is quite the contrary: % Degen ratio 0.47 in Table 8.4 reveals that the percentage of total degenerate iterations associated with MINOS 5.51 is much lower than for RDPPA 1.10. We were astonished initially by the fact that MINOS 5.51 required 4.58 times as many total iterations as those required by RDPPA 1.10 despite such a low ratio of percentages of total degenerate iterations! This is also true for either the Kennington problems or the BPMPD problems alone. Such a striking contrast is quite encouraging as it provides a clue to the merit of the proposed algorithm.

We emphasize that even if the proportion of degenerate iterations is high, the total iterations could still be low; in other words, *there is no inevitable correlation between an algorithm's inefficiency and degeneracy*, contradicting a widespread belief that degeneracy is a primary cause of inefficiency of pivot algorithms. Such an observation coincides with the recent experiments carried out with a sparse implementation of a generalized revised simplex algorithm [25]. This is also supported by experiments with steepest-edge pivot rules, which outperformed conventional rules by large margins even though the proportions of degenerate iterations were similar [8].

Finally, much work remains to be done. As an early version, RDPPA 1.10 still has much room for improvement. First of all, the steepest-edge rule is important to both primal and dual simplex algorithms. According to Bixby [2], its application is the major thrust that has driven the dual simplex algorithm to become a competitor of the primal simplex algorithm. We expect that this rule is equally important to the algorithm presented in this paper although it is still open how to implement it in our context efficiently. Other techniques known to be good in practice should also be considered. We leave all these to our future research.

Acknowledgment. The author is grateful to Professor Michael A. Saunders for making very valuable and detailed comments and suggestions that improved this paper considerably, and for providing author the MINOS 5.51 package. Indeed, the author has benefited greatly from discussions with him in recent years.

REFERENCES

[1] E. M. L. Beale, *An alternative method for linear programming*, Proc. Cambridge Philos. Soc., 50 (1954), pp. 513–523.

[2] R. E. Bixby, *Solving real-world linear programs: A decade and more of progress*, Oper. Res., 50 (2002), pp. 3–15.

[3] W. J. Carolan, J. E. Hill, J. L. Kennington, S. Niemi, and S. J. Wichmann, *An empirical evaluation of the KORBX algorithms for military airlift applications*, Oper. Res., 38 (2002), pp. 240–248.

[4] G. B. Dantzig, *Maximization of a linear function of variables subject to linear inequalities*, in Activity Analysis of Production and Allocation, T. C. Koopmans, ed., Wiley, New York, 1951, pp. 339–347.

[5] G. B. Dantzig, *The Dual Simplex Algorithm*, RAND report RM-1270, The RAND Corporation, Santa Monica, CA, 1954.

[6] G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ, 1963.

[7] R. Fletcher, *Practical Methods of Optimization*, Vol. 2, Wiley, Chichester, UK, 1981.

[8] J. J. H. Forrest and D. Goldfarb, *Steepest-edge simplex algorithms for linear programming*, Math. Program., 57 (1992), pp. 341–374.

[9] D. M. Gay, *Electronic mail distribution of linear programming test problems*, Math. Program. Soc. COAL Newslett., 13 (1985), pp. 10–12.

[10] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *Maintaining LU factors of a general sparse matrix*, Linear Algebra Appl., 88/89 (1987), pp. 239–270.

[11] P. E. Gill, W. Murray, and M. A. Saunders, *SNOPT: An SQP algorithm for large-scale constrained optimization*, SIAM J. Optim., 12 (2002), pp. 979–1006.

[12] P. M. J. Harris, *Pivot selection methods of the Devex LP code*, Math. Program., 5 (1974), pp. 1–28.

[13] B. Hattersley and J. Wilson, *A dual approach to primal degeneracy*, Math. Program., 42 (1988), pp. 135–145.

[14] C. E. Lemke, *The dual method of solving the linear programming problem*, Naval Res. Logist. Quart., 1 (1954), pp. 36–47.

[15] B. A. Murtagh and M. A. Saunders, *MINOS* 5.5 *User's Guide*, Technical report SOL 83-20R, Department of Operations Research, Stanford University, Stanford, CA, 1998.

[16] P.-Q. Pan, *Practical finite pivoting rules for the simplex method*, OR Spectrum, 12 (1990), pp. 219–225.

[17] P.-Q. Pan, *A simplex-like method with bisection for linear programming*, Optimization, 22 (1991), pp. 717–743.

[18] P.-Q. Pan, *A modified bisection simplex method for linear programming*, J. Comput. Math., 14 (1996), pp. 249–255.

[19] P.-Q. Pan, *The most-obtuse-angle row pivot rule for achieving dual feasibility: A computational study*, European J. Oper. Res., 101 (1997), pp. 167–176.

[20] P.-Q. Pan, *A dual projective simplex method for linear programming*, Comput. Math. Appl., 35 (1998), pp. 119–135.

[21] P.-Q. Pan, *A basis-deficiency-allowing variation of the simplex method*, Comput. Math. Appl., 36 (1998), pp. 33–53.

[22] P.-Q. Pan, *A projective simplex method for linear programming*, Linear Algebra Appl., 292 (1999), pp. 99–125.

[23] P.-Q. Pan, *A projective simplex algorithm using LU factorization*, Comput. Math. Appl., 39 (2000), pp. 187–208.

[24] P.-Q. Pan, *A dual projective pivot algorithm for linear programming*, Comput. Optim. Appl., 29 (2004), pp. 333–344.

[25] P.-Q. Pan, *A generalization of the revised simplex algorithm for linear programming*, Comput. Optim. Appl., to appear.

[26] P.-Q. Pan and Y. Pan, *A phase*-1 *approach to the generalized simplex algorithm*, Comput. Math. Appl., 41 (2001), pp. 1455–1464.

[27] J. K. Reid, *A sparsity-exploiting variant of the Bartels–Golub decomposition for linear programming bases*, Math. Program., 24 (1982), pp. 55–69.

[28] M. A. Saunders, *private communication*, June 8, 2004.

SIAM J. Optim.
Vol. 16, No. 1, pp. 69–95

POLYHEDRAL RISK MEASURES IN STOCHASTIC PROGRAMMING[*]

ANDREAS EICHHORN[†] AND WERNER RÖMISCH[†]

Abstract. We consider stochastic programs with risk measures in the objective and study stability properties as well as decomposition structures. Thereby we place emphasis on dynamic models, i.e., multistage stochastic programs with multiperiod risk measures. In this context, we define the class of polyhedral risk measures such that stochastic programs with risk measures taken from this class have favorable properties. Polyhedral risk measures are defined as optimal values of certain linear stochastic programs where the arguments of the risk measure appear on the right-hand side of the dynamic constraints. Dual representations for polyhedral risk measures are derived and used to deduce criteria for convexity and coherence. As examples of polyhedral risk measures we propose multiperiod extensions of the Conditional-Value-at-Risk.

Key words. stochastic programming, convex risk measure, coherent, polyhedral, mean-risk, quantitative stability, probability metrics, dual decomposition

AMS subject classifications. 90C15, 91B30

DOI. 10.1137/040605217

1. Introduction.

Stochastic programs are essentially known to minimize, maximize, or bound expected values. From a theoretical point of view they easily offer the possibility to minimize or bound risk functionals since they rest upon stochastic models. This idea goes back to [14]. However, in practice it may happen that incorporating risk measures in stochastic programs makes them much harder to solve, especially if integer decisions are included. In addition, other favorable properties like stability with respect to approximations or duality results may get lost. In this paper considerations are made about the question as to how risk measures should be designed so that stochastic programs incorporating them show similar properties as stochastic programs based on expected values only. As a result, the class of polyhedral risk measures is introduced.

Of course, when analyzing risk measures with respect to their practicability for stochastic programs, one has to determine first of all what is understood by the expression *risk measure* and what properties are required from the viewpoint of economic considerations. Here, a (one-period) risk measure ρ will be understood as a functional from some set of real random variables to the real numbers. Random variables will be denoted by the letter z, they will represent uncertain (usually monetary) values for which larger outcomes are preferred to lower ones. The value $\rho(z)$ gives information about the riskiness of z, i.e., a high value $\rho(z)$ indicates a high danger of reaching low values.

Risk measures are broadly discussed in financial mathematics. For one-period risk measures, i.e., for risk measures that depend on one random variable only, there is a relatively high degree of agreement among the community about the desirable

[*] Received by the editors March 15, 2004; accepted for publication (in revised form) September 13, 2004; published electronically August 17, 2005. This research was supported by the DFG Research Center Matheon *Mathematics for Key Technologies* at Berlin and by a grant of EDF Electricité de France.

http://www.siam.org/journals/siopt/16-1/60521.html

[†] Humboldt-University Berlin, Institute of Mathematics, 10099 Berlin, Germany (eichhorn@math.hu-berlin.de, romisch@math.hu-berlin.de, www.math.hu-berlin.de/~romisch).

properties. Possibly the most important work in this context is the axiomatic characterization of coherent risk measures [1], where the risk $\rho(z)$ is understood as the minimal amount of additional (risk-free) capital that is required to make the position z acceptable. Several generalizations of this paper followed, e.g., [6, 13, 10, 28]; see also Chapter 4 in the monograph [11]. Further desirable properties, namely, the consistency of risk measures with stochastic dominance rules, were suggested in [15, 17, 18, 19]. In addition, there are papers dealing with specific risk measures, e.g., [27, 20, 38]; see also the volumes [7, 41]. Recently, a theory for convex optimization of convex risk measures has been developed in [35].

Currently, generalizations of one-period risk measures to different dynamic settings are discussed in the literature. Such generalizations become necessary when information is revealed gradually with the passing of time and a sequence of random variables $z_1, \dots, z_T$ is to be assessed with respect to its riskiness. In the literature, the settings as well as the postulated properties for risk functionals differ more than in the one-period case. Generally speaking, there are two classes of settings depending on whether liquidity risk over a time period is considered or intermediate monitoring by supervisors is to be anticipated. In the latter case an entire risk measure process $\rho_1, \dots, \rho_T$ is defined; see [25, 42] and also [3, 2]. The more important case from the viewpoint of optimization is the case where one has one real number $\rho(z_1, \dots, z_T)$ that represents the risk of the entire process (multiperiod risk). Such concepts are presented in [22, 36, 21] and again in [3, 2]. As in the one-period case, the number $\rho(z_1, \dots, z_T)$ can be understood as minimal capital requirement for the overall time period so that the strategy corresponding to $z_1, \dots, z_T$ is acceptable.

In the present paper, we consider (mixed-integer) multistage stochastic programs of the form

$$\text{(1.1)} \qquad \min \left\{ \mathbb{E}\left[\sum_{t=1}^{T} \langle b_t(\xi_t), x_t \rangle\right] \;\middle|\; \begin{array}{l} x_t \text{ is } \mathcal{F}_t\text{-measurable}, \\ \sum_{\tau=0}^{t-1} A_{t,\tau}(\xi_t) x_{t-\tau} = h_t(\xi_t) \text{ a.s.}, \\ x_t \in X_t \text{ a.s. } (t = 1, \dots, T) \end{array} \right\}$$

as starting point, where $(\xi_t)_{t=1}^T$ is a stochastic process and $\mathcal{F}_t = \sigma(\xi_1, \dots, \xi_t)$, the sets X_t are closed and have polyhedral convex hulls, $b_t(\cdot)$ are cost coefficients, $h_t(\cdot)$ are right-hand sides, and $A_{t,\tau}(\cdot)$, $\tau = 0, \dots, t-1$, are matrices having appropriate dimensions and possibly depending on ξ_t for $t = 1, \dots, T$.

Much is known for expectation-based stochastic programs, e.g., on optimality and duality, decomposition methods, and statistical approximations and stability (cf. [34]). Most of these results are essentially based on the fact that $\mathbb{E}$ is a linear operator. As will be seen below in section 2, risk measures are usually by no means linear. Hence, if we change from expectation to a risk measure in (1.1), many known results will no longer be valid. Nevertheless, there are results about incorporating certain risk functionals into (stochastic) optimization problems, e.g., [38, 35, 37]. In particular, the Conditional-Value-at-Risk turns out to behave very opportunely in stochastic programs because it allows a reformulation of the risk aversive problem as an expectation-based problem with additional variables (cf. [27, 20, 40]).

However, from an economic point of view not every risk measure is suitable for any application. In particular, for multistage stochastic programs it may become necessary to incorporate multiperiod risk measures, i.e., to minimize $\rho(z_1, \dots, z_T)$ with $z_t = -\sum_{\tau=1}^{t} \langle b_\tau(\xi_\tau), x_\tau \rangle$ denoting the intermediate values. Hence, it would be convenient to have an entire class of risk measures at hand such that every risk measure from this class behaves opportunely in stochastic programs.

Such a class will be introduced in section 2 for the one-period case, namely the class of polyhedral risk measures. Conditions implying that polyhedral risk measures are coherent and consistent with second order stochastic dominance are provided. In section 3 this class will be extended to the multiperiod case. Briefly, polyhedral risk measures are defined as optimal values of certain simple linear stochastic programs. In section 4 it will be shown that, indeed, several properties of expectation-based stochastic programs remain valid for stochastic programs with polyhedral risk measures as objectives. This is due to the fact that a problem of the form (1.1) with $\mathbb{E}$ replaced by a polyhedral risk measure ρ can easily be transformed into a stochastic program with additional variables and an objective consisting of the expectation of a linear function. In particular, we present stability results for two-stage stochastic programs with polyhedral risk measures and show that dual decomposition structures are maintained.

2. Polyhedral risk measures. We consider a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and the linear space of real random variables $L_p(\Omega, \mathcal{F}, \mathbb{P})$ with some $p \in [1, \infty]$. According to [10, 11] a functional $\rho : L_p(\Omega, \mathcal{F}, \mathbb{P}) \to \bar{\mathbb{R}}$ is called a *risk measure* if it satisfies the following two conditions for all $z, \tilde{z} \in L_p(\Omega, \mathcal{F}, \mathbb{P})$:

(i) If $z \leq \tilde{z}$ a.s., then $\rho(z) \geq \rho(\tilde{z})$ *(monotonicity).*

(ii) For each $r \in \mathbb{R}$ we have $\rho(z + r) = \rho(z) - r$ *(translation invariance).*

A risk measure ρ is called *convex* if it satisfies the condition

$$\rho(\mu z + (1-\mu)\tilde{z}) \leq \mu\rho(z) + (1-\mu)\rho(\tilde{z})$$

for all $z, \tilde{z} \in L_p(\Omega, \mathcal{F}, \mathbb{P})$ and $\mu \in [0, 1]$. A convex risk measure is called *coherent* if it is *positively homogeneous*, i.e., $\rho(\mu z) = \mu\rho(z)$ for all $\mu \geq 0$ and $z \in L_p(\Omega, \mathcal{F}, \mathbb{P})$. There is a number of representation theorems for convex and especially for coherent risk measures in the literature emerging from convex duality. Next, we cite one of these representations adapted to our needs. Therefore, we set

$$\mathcal{D} := \{f \in L_1(\Omega, \mathcal{F}, \mathbb{P}) : f \geq 0 \text{ a.s.}, \mathbb{E}[f] = 1\},$$

the set of all density functions for $(\Omega, \mathcal{F}, \mathbb{P})$.

Theorem 2.1. *Let $\rho : L_p(\Omega, \mathcal{F}, \mathbb{P}) \to \bar{\mathbb{R}}$ with $p \in [1, \infty]$. Assume that ρ is lower semicontinuous. Then ρ is a coherent risk measure if and only if the following condition holds:*

$$\exists\, \mathcal{P}_\rho \subseteq \mathcal{D} \text{ convex} : \rho(z) = \sup_{f \in \mathcal{P}_\rho} \mathbb{E}[-zf] \;\; \forall\, z \in L_p(\Omega, \mathcal{F}, \mathbb{P}).$$

Proof. "⇒" is stated in [35, Corollary 1] and "⇐" is easily seen by checking the four properties of the definition above; see also [11, 6, 28]. □

Now we are ready to define the class of polyhedral risk measures.

Definition 2.2. *A risk measure ρ on $L_p(\Omega, \mathcal{F}, \mathbb{P})$ with some $p \in [1, \infty]$ will be called* polyhedral *if there exist $k_1, k_2 \in \mathbb{N}$, $c_1, w_1 \in \mathbb{R}^{k_1}$, $c_2, w_2 \in \mathbb{R}^{k_2}$, a nonempty polyhedral set $Y_1 \subseteq \mathbb{R}^{k_1}$, and a polyhedral cone $Y_2 \subseteq \mathbb{R}^{k_2}$ such that*

$$\rho(z) = \inf \left\{ \langle c_1, y_1 \rangle + \mathbb{E}[\langle c_2, y_2 \rangle] \;\middle|\; \begin{array}{l} y_1 \in Y_1, \\ y_2 \in L_p(\Omega, \mathcal{F}, \mathbb{P}),\ y_2 \in Y_2 \text{ a.s.}, \\ \langle w_1, y_1 \rangle + \langle w_2, y_2 \rangle = z \text{ a.s.} \end{array} \right\} \tag{2.1}$$

for every $z \in L_p(\Omega, \mathcal{F}, \mathbb{P})$. Here, $\mathbb{E}$ denotes the expectation on $(\Omega, \mathcal{F}, \mathbb{P})$ and $\langle \cdot, \cdot \rangle$ a scalar product on $\mathbb{R}^{k_1}$ or $\mathbb{R}^{k_2}$.

Hence, expressed in the language of stochastic programming, a polyhedral risk measure is given as the optimal value of a certain two-stage stochastic program with random right-hand side. We use the term *polyhedral* because, for $\#\Omega < \infty$, the space $L_p(\Omega, \mathcal{F}, \mathbb{P})$ can be identified with $\mathbb{R}^{\#\Omega}$ and in this case a risk measure defined by (2.1) is indeed a polyhedral function on $\mathbb{R}^{\#\Omega}$.

Remark 2.3. Of course, the negative expectation is a polyhedral risk measure. Moreover, a convex combination of (negative) expectation and a polyhedral risk measure is again a polyhedral risk measure: Let $\mu \in [0,1]$ and ρ be a polyhedral risk measure with dimensions k_t, vectors c_t and w_t $(t=1,2)$, and polyhedral set/cone Y_1 / Y_2. Then the risk measure $\hat{\rho} := \mu\rho - (1-\mu)\mathbb{E}$ is polyhedral with the same dimensions k_t and the same sets Y_t and vectors $\hat{w}_1 := w_1$, $\hat{w}_2 := w_2$, $\hat{c}_1 := \mu c_1 - (1-\mu)w_1$, and $\hat{c}_2 := \mu c_2 - (1-\mu)w_2$. Thus, so-called *mean-risk models*, where expectation and risk are optimized simultaneously, do not need to be considered separately.

Next, we derive dual representations for (2.1). To this end, we do not need to assume that ρ is a risk measure in the sense of [10, 11], i.e., that it is monotone and translation invariant. We conclude in our first result that ρ is a convex functional. To state this result, we use the notation[1]

$$D_{\rho,t} := \{u \in \mathbb{R} : c_t + uw_t \in -Y_t^*\} \quad (t=1,2)$$

for the so-called *dual feasible sets.*

THEOREM 2.4. *Let ρ be a functional of the form* (2.1) *on* $L_p(\Omega, \mathcal{F}, \mathbb{P})$ *with some* $p \in [1, \infty)$. *Assume*

(i) *complete recourse:* $\langle w_2, Y_2 \rangle = \mathbb{R}$,

(ii) *dual feasibility:* $D_{\rho,1} \cap D_{\rho,2} \neq \emptyset$.

Then ρ is finite, convex, and continuous. Further, the representation

$$\rho(z) = \inf_{y_1 \in Y_1} \left\{ \langle c_1, y_1 \rangle + \mathbb{E}\left[\max_{\ell=1,2} u_\ell \left(\langle w_1, y_1 \rangle - z \right) \right] \right\} \tag{2.2}$$

holds with two real numbers u_1 and u_2 that are the endpoints of $D_{\rho,2}$ which is a compact interval in $\mathbb{R}$. Furthermore, with $\frac{1}{p} + \frac{1}{p'} = 1$, ρ admits the dual representation

$$\rho(z) = \sup \left\{ -\mathbb{E}[\lambda z] + \inf_{y_1 \in Y_1} \langle c_1 + \mathbb{E}[\lambda] w_1, y_1 \rangle \;\middle|\; \begin{array}{l} \lambda \in L_{p'}(\Omega, \mathcal{F}, \mathbb{P}), \\ c_2 + \lambda w_2 \in -Y_2^* \ \textit{a.s.} \end{array} \right\}. \tag{2.3}$$

In particular, if Y_1 is a cone, then ρ is positively homogeneous and (2.3) *becomes*

$$\rho(z) = \sup \left\{ -\mathbb{E}[\lambda z] \;\middle|\; \begin{array}{l} \lambda \in L_{p'}(\Omega, \mathcal{F}, \mathbb{P}), \\ c_1 + \mathbb{E}[\lambda] w_1 \in -Y_1^*,\ c_2 + \lambda w_2 \in -Y_2^* \ \textit{a.s.} \end{array} \right\}. \tag{2.4}$$

Proof. Finiteness, convexity, continuity, and the representations (2.3) and (2.4) will be proved in a more general framework in section 3, Theorem 3.9. Representation (2.2) follows from LP duality applied to the second stage program. (Note that due to [29, Theorem 14.60] the minimization for the second stage can be carried out pointwise on Ω.) Namely, it holds for each $y_1 \in Y_1$ and each $z \in \mathbb{R}$ that

$$\begin{aligned} &\min \{ \langle c_2, y_2 \rangle : y_2 \in Y_2,\ \langle w_1, y_1 \rangle + \langle w_2, y_2 \rangle = z \} \\ &= \max \{ u (\langle w_1, y_1 \rangle - z) : c_2 + uw_2 \in -Y_2^* \}. \end{aligned}$$

[1] Thereby Y_t^* denotes the *polar cone* of Y_t. For a nonempty set Y the polar cone Y^* is defined by $Y^* = \{y^* : \langle y, y^* \rangle \leq 0 \ \forall y \in Y\}$.

Due to complete recourse and dual feasibility the feasible sets of both problems are nonempty and the joint optimal value is finite for each $y_1 \in Y_1$ and each $z \in \mathbb{R}$. Since the expression $\langle w_1, y_1 \rangle - z$ can reach any real number and the feasible set of the right problem $D_{\rho,2}$ does not depend on y_1 and z, it is clear that the latter is bounded, i.e., it is a compact interval in $\mathbb{R}$. Of course, the maximum is attained for u being an endpoint of $D_{\rho,2}$. $\square$

If a functional ρ on $L_p(\Omega, \mathcal{F}, \mathbb{P})$ is defined by formula (2.1), the question arises for which choice of c_t, w_t, and Y_t $(t = 1, 2)$ this functional is a (convex) risk measure in the sense of [10, 11]. Formula (2.4) provides a sufficient criterion for a functional of the form (2.1) to be a coherent risk measure in case Y_1 is a cone.

COROLLARY 2.5. *Let ρ be a functional on $L_p(\Omega, \mathcal{F}, \mathbb{P})$ of the form* (2.1) *with Y_1 being a polyhedral cone and $1 \leq p < \infty$. Let the conditions of Theorem* 2.4 *be satisfied (complete recourse, dual feasibility) and assume that*

$$\Lambda_\rho := \left\{ \lambda \in L_{p'}(\Omega, \mathcal{F}, \mathbb{P}) \;\middle|\; \begin{array}{l} c_1 + \mathbb{E}\,[\lambda]\, w_1 \in -Y_1^*, \\ c_2 + \lambda w_2 \in -Y_2^* \ \ a.s. \end{array} \right\} \subseteq \mathcal{D}. \tag{2.5}$$

Then ρ is a coherent risk measure.

Proof. The proof follows immediately from Theorems 2.1 and 2.4 with $\mathcal{P}_\rho := \Lambda_\rho$ since, of course, continuity implies lower semicontinuity. $\square$

The following result provides a sufficient criterion for a functional of the form (2.1) to be a convex risk measure in case Y_1 is *not* a cone.

PROPOSITION 2.6. *Let ρ be a functional on $L_p(\Omega, \mathcal{F}, \mathbb{P})$ of the form* (2.1) *with $p \in [1, \infty)$. Assume that complete recourse and dual feasibility hold and that $D_{\rho,2} \subseteq \mathbb{R}_+$ and let c_1, w_1, and Y_1 be of the form $c_1 = (\hat{c}_1, 1)$, $w_1 = (\hat{w}_1, -1)$, and $Y_1 = \hat{Y}_1 \times \mathbb{R}$, where $\hat{w}_1, \hat{c}_1 \in \mathbb{R}^{k_1 - 1}$, and $\hat{Y}_1 \subseteq \mathbb{R}^{k_1-1}$. Then ρ is a (polyhedral) convex risk measure.*

Proof. Finiteness and convexity of ρ follow from Theorem 2.4. The monotonicity property (i) follows from the representation (2.2) and the fact that u_1 and u_2 are nonnegative. Indeed, let $z, \tilde{z} \in L_p(\Omega, \mathcal{F}, \mathbb{P})$ be such that $z \leq \tilde{z}$ a.s.; then we have $\mathbb{E}[\max_{\ell=1,2} u_\ell(\langle w_1, y_1 \rangle - z)] \geq \mathbb{E}[\max_{\ell=1,2} u_\ell(\langle w_1, y_1 \rangle - \tilde{z})]$ for every $y_1 \in Y_1$. The translation invariance condition (ii) follows by setting $y_1 = (\hat{y}_1, \bar{y}_1)$, $\tilde{y}_1 := \bar{y}_1 + r \in \mathbb{R}$ as a consequence of the identity

$$\begin{aligned} &\rho(z + r) \\ &= \inf \left\{ \langle \hat{c}_1, \hat{y}_1 \rangle + \bar{y}_1 + \mathbb{E}\left[\max\nolimits_{\ell=1,2} u_\ell \left(\langle \hat{w}_1, \hat{y}_1 \rangle - \bar{y}_1 - (z + r) \right)\right] : \hat{y}_1 \in \hat{Y}_1,\ \bar{y}_1 \in \mathbb{R} \right\} \\ &= \inf \left\{ \langle \hat{c}_1, \hat{y}_1 \rangle + \tilde{y}_1 + \mathbb{E}\left[\max\nolimits_{\ell=1,2} u_\ell \left(\langle \hat{w}_1, \hat{y}_1 \rangle - \tilde{y}_1 - z \right)\right] : \hat{y}_1 \in \hat{Y}_1,\ \tilde{y}_1 \in \mathbb{R} \right\} - r \\ &= \rho(z) - r \end{aligned}$$

for each $r \in \mathbb{R}$ and $z \in L_p(\Omega, \mathcal{F}, \mathbb{P})$. $\square$

The assumptions of Proposition 2.6 guarantee even a stronger type of monotonicity than imposed earlier for risk measures. Such stronger monotonicity properties are based on so-called *integral stochastic orders* or *stochastic dominance rules* (see [15] for a recent survey). For real random variables z and $\tilde{z}$ in $L_1(\Omega, \mathcal{F}, \mathbb{P})$, stochastic dominance rules are defined by classes $\mathcal{F}$ of measurable real-valued functions on $\mathbb{R}$. A stochastic dominance rule is defined by

$$z \preceq_{\mathcal{F}} \tilde{z} \quad \text{if} \quad \mathbb{E}[f(z)] \leq \mathbb{E}[f(\tilde{z})]$$

for each $f \in \mathcal{F}$ such that the expectations exist. Important special cases are the class of $\mathcal{F}_{nd}$ of nondecreasing functions and the class $\mathcal{F}_{ndc}$ of nondecreasing concave functions. In these cases the rules are also called *first order stochastic dominance* and *second order stochastic dominance* and denoted by $\preceq_{FSD}$ and $\preceq_{SSD}$, respectively. Clearly, $z \preceq_{FSD} \tilde{z}$ implies $z \preceq_{SSD} \tilde{z}$. The relation $z \preceq_{FSD} \tilde{z}$ is equivalent to $\mathbb{P}(z \leq t) \geq \mathbb{P}(\tilde{z} \leq t)$ for each $t \in \mathbb{R}$. Furthermore, $z \preceq_{SSD} \tilde{z}$ is equivalent to the condition $\mathbb{E}[\min\{z,t\}] \leq \mathbb{E}[\min\{\tilde{z},t\}]$ for each $t \in \mathbb{R}$ (cf. [15, section 8]). Other equivalent characterizations of $z \preceq_{SSD} \tilde{z}$ are $\int_{-\infty}^{\eta} \mathbb{P}(z \leq t)dt \geq \int_{-\infty}^{\eta} \mathbb{P}(\tilde{z} \leq t)dt$ for each $t \in \mathbb{R}$ (cf. [17, 18]) and $\int_0^p q_\alpha(z)d\alpha \leq \int_0^p q_\alpha(\tilde{z})d\alpha$ for each $p \in (0,1]$ (cf. [19]) with $q_\alpha(z) = \inf\{r \in \mathbb{R} : \mathbb{P}(z \leq r) \geq \alpha\}$ denoting the (lower) α-quantile of the random variable z.

In [19, 17, 18] the consistency of risk measures ρ with certain stochastic dominance rules $\preceq_{\mathcal{F}}$ is studied. In particular, it is said that ρ is *consistent with second order stochastic dominance* if $z \preceq_{SSD} \tilde{z}$ implies $\rho(z) \geq \rho(\tilde{z})$.

PROPOSITION 2.7. *Let ρ be a functional on $L_p(\Omega, \mathcal{F}, \mathbb{P})$ of the form* (2.1) *with $p \in [1,\infty)$. Assume that complete recourse and dual feasibility hold and that $D_{\rho,2} \subseteq \mathbb{R}_+$. Then ρ is consistent with second order stochastic dominance.*

Proof. Due to Theorem 2.4 the representation (2.2) holds with $u_1, u_2 \in \mathbb{R}_+$. Define for $y_1 \in Y_1$ the real-valued function g_{y_1} given by

$$g_{y_1}(t) := \langle c_1, y_1 \rangle + \max_{\ell=1,2} u_\ell \left(\langle w_1, y_1 \rangle - t \right)$$

for $t \in \mathbb{R}$. Note that g_{y_1} is convex and, because of $u_1, u_2 \geq 0$, nonincreasing. Let $z \preceq_{SSD} \tilde{z}$. Then $\mathbb{E}[-g_{y_1}(z)] \leq \mathbb{E}[-g_{y_1}(\tilde{z})]$ for all $y_1 \in Y_1$ and, thus, $\rho(z) = \inf_{y_1 \in Y_1} \mathbb{E}[g_{y_1}(z)] \geq \inf_{y_1 \in Y_1} \mathbb{E}[g_{y_1}(\tilde{z})] = \rho(\tilde{z})$. □

Remark 2.8. For a risk measure ρ on $L_p(\Omega, \mathcal{F}, \mathbb{P})$ the *acceptance set* $\mathcal{A}_\rho$ is defined by $\mathcal{A}_\rho = \{z \in L_p(\Omega, \mathcal{F}, \mathbb{P}) : \rho(z) \leq 0\}$ [3, 11]; let the conditions of Theorem 2.4 be satisfied. Then, since ρ is a convex functional, $\mathcal{A}_\rho$ is a convex set. If, in addition, Y_1 is a cone, then $\mathcal{A}_\rho$ is a convex cone. Regarding (2.5) it is obvious that

$$\mathcal{A}_\rho = \{z \in L_p(\Omega, \mathcal{F}, \mathbb{P}) \mid \forall\, \lambda \in \Lambda_\rho : \mathbb{E}\left[\lambda z\right] \geq 0\} = -\Lambda_\rho^*$$

in this case. Of course, if $\Omega = \{\omega_1, \dots, \omega_S\}$, then Λ_ρ is a polyhedron in $\mathbb{R}^S$, thus $\mathcal{A}_\rho = -\Lambda_\rho^*$ is a polyhedral cone.

For stability analysis of stochastic programs (cf. section 4.1), it is important to know whether first stage solution sets are bounded or not. For a polyhedral risk measure ρ satisfying complete recourse and dual feasibility, the first stage solution set $S(\rho(z)) \subseteq Y_1$ can be written according to the dual representation (2.2) as

$$S(\rho(z)) := \{y_1 \in Y_1 : \langle c_1, y_1 \rangle + \mathbb{E}\left[\max\nolimits_{\ell=1,2} u_\ell \left(\langle w_1, y_1 \rangle - z \right)\right] = \rho(z)\}. \tag{2.6}$$

The following proposition provides a sufficient criterion for the boundedness of $S(\rho(z))$ for a large class of polyhedral risk measures.

PROPOSITION 2.9. *Let ρ be a functional on $L_p(\Omega, \mathcal{F}, \mathbb{P})$ of the form* (2.1) *with $p \in [1,\infty)$. Let the conditions of Theorem* 2.4 *be satisfied (complete recourse, dual feasibility) and assume that $S(\rho(0))$ is a nonempty, bounded subset in $\mathbb{R}^{k_1}$. Then $S(\rho(z))$ is nonempty, convex, and compact for any $z \in L_p(\Omega, \mathcal{F}, \mathbb{P})$.*

Proof. Clearly, Theorem 2.4 implies convexity and closedness of $S(\rho(z))$. It remains to be seen whether $S(\rho(z))$ is nonempty and bounded. The polyhedral set Y_1 can be represented in the form $Y_1 = P_1 + C_1$, where P_1 is a bounded polyhedron and C_1 a polyhedral cone (e.g., [29, Corollary 3.53]). Let $0 \neq d_1 \in C_1$ (hence, $\mu d_1 \in C_1$ for any $\mu \geq 0$) and $g_{d_1}(0) = \langle c_1, d_1 \rangle + \max_{\ell=1,2} u_\ell \langle w_1, d_1 \rangle$. Next we show $g_{d_1}(0) > 0$. Suppose $g_{d_1}(0) < 0$ and let $p_1 \in P_1$, $\mu > 0$. Then $p_1 + \mu d_1 \in Y_1$ and we obtain

$\rho(0) \leq g_{p_1}(0) + \mu g_{d_1}(0)$. This contradicts to the finiteness of ρ since $\mu > 0$ may be chosen arbitrarily large. If $g_{d_1}(0) = 0$, the set $S(\rho(0))$ would contain the unbounded subset $\{\bar{y}_1 + \mu d_1 : \mu \geq 0\}$ for some $\bar{y}_1 \in S(\rho(0))$. Now, let $z \in L_p(\Omega, \mathcal{F}, \mathbb{P})$ and let $(y_{1,n})$ be a sequence with $y_{1,n} = p_{1,n} + d_{1,n} \in Y_1$, $p_{1,n} \in P_1$, $d_{1,n} \in C_1$, and

$$\langle c_1, y_{1,n} \rangle + \mathbb{E}\left[\max_{\ell=1,2} u_\ell\left(\langle w_1, y_{1,n} \rangle - z\right)\right] \to \rho(z).$$

Since P_1 is bounded, we may assume without loss of generality that $(p_{1,n})$ is convergent to some $\bar{p}_1 \in P_1$. Suppose that $(y_{1,n})$ is unbounded. Then we may assume without loss of generality that $\|d_{1,n}\| \to \infty$ and $\frac{d_{1,n}}{\|d_{1,n}\|} \to \bar{d}_1 \in C_1$. It follows that

$$\rho(z) = \lim_{n\to\infty} \left(\langle c_1, y_{1,n} \rangle + \mathbb{E}\left[\max_{\ell=1,2} u_\ell\left(\langle w_1, y_{1,n} \rangle - z\right)\right]\right) = \lim_{n\to\infty} \|d_{1,n}\| \alpha_n$$

with $\alpha_n := \langle c_1, \frac{y_{1,n}}{\|d_{1,n}\|} \rangle + \mathbb{E}[\max_{\ell=1,2} u_\ell(\langle w_1, \frac{y_{1,n}}{\|d_{1,n}\|} \rangle - \frac{z}{\|d_{1,n}\|})]$. Obviously, it holds that $\alpha_n \to g_{\bar{d}_1}(0) > 0$, hence $\rho(z) = \lim_{n\to\infty} \|y_{1,n}\| \alpha_n = \infty$. This is a contradiction. It follows that each minimizing sequence $(y_{1,n})$ in Y_1 is always bounded. This implies both existence of a solution and boundedness of the solution set $S(\rho(z))$. □

Example 2.10. We consider the *Conditional-* or *Average-Value-at-Risk* at level $\alpha \in (0,1)$ ($CVaR_\alpha$ or $AVaR_\alpha$) defined by

$$CVaR_\alpha(z) := \frac{1}{\alpha} \int_0^\alpha VaR_\gamma(z) d\gamma = \inf_{r\in\mathbb{R}} \left\{ r + \frac{1}{\alpha} \mathbb{E}\left[(r+z)^-\right] \right\}, \tag{2.7}$$

where $VaR_\gamma(z) := \inf\{r \in \mathbb{R} : \mathbb{P}(z+r < 0) \leq \gamma\} = -\bar{q}_\gamma(z)$ is the Value-at-Risk at level $\gamma \in (0,1)$ (see [11, section 4.4] and [27]) and $a^- = \max\{0, -a\}$ denotes the negative part of a real number a. The number $\bar{q}_\gamma(z)$ is also called the upper γ-quantile of z. Introducing variables for positive and negative parts of the infimum representation in (2.7), respectively, leads to

$$CVaR_\alpha(z) = \inf \left\{ y_1 + \frac{1}{\alpha}\mathbb{E}\left[y_2^{(2)}\right] \;\middle|\; \begin{array}{l} y_1 \in \mathbb{R},\ y_2 \in L_1(\Omega, \mathcal{F}, \mathbb{P}), \\ y_2 \in \mathbb{R}_+ \times \mathbb{R}_+ \text{ a.s.}, \\ y_2^{(1)} - y_2^{(2)} = z + y_1 \text{ a.s.} \end{array} \right\}. \tag{2.8}$$

Thus, $CVaR_\alpha$ is of the form (2.1) by setting $k_1 = 1$, $k_2 = 2$, $w_1 = -1$, $c_1 = 1$, $c_2 = (0, \frac{1}{\alpha})$, $w_2 = (1, -1)$, $Y_1 = \mathbb{R}$, and $Y_2 = \mathbb{R}^2_+$, and, hence, is a polyhedral risk measure. Moreover, $\langle w_2, Y_2 \rangle = \mathbb{R}$, $D_{\rho,1} = D_{\rho,1} \cap D_{\rho,2} = \{1\}$, and $D_{\rho,2} = [0, \frac{1}{\alpha}] \subseteq \mathbb{R}_+$, thus the dual representation (2.4) holds and $CVaR_\alpha$ is consistent with second order stochastic dominance. The representation (2.2) holds with $u_1 = 0$ and $u_2 = \frac{1}{\alpha}$. The condition $c_2 + \lambda w_2 \in -Y_2^*$ in the dual representation (2.4) is equivalent to $\lambda \in [0, \frac{1}{\alpha}]$. Hence, (2.4) becomes

$$CVaR_\alpha(z) = \sup\left\{ -\mathbb{E}\left[\lambda z\right] : \lambda \in L_{p'}(\Omega, \mathcal{F}, \mathbb{P}),\ \lambda \in \left[0, \tfrac{1}{\alpha}\right] \text{ a.s.}, \mathbb{E}\left[\lambda\right] = 1 \right\} \tag{2.9}$$

for each $z \in L_p(\Omega, \mathcal{F}, \mathbb{P})$, $1 \leq p < \infty$. Corollary 2.5 applies thus, $CVaR$ is a coherent risk measure, too. Similar results have already been shown in [28, 19]. Furthermore, it is shown in [27] that the set $\{r \in \mathbb{R} : CVaR_\alpha(z) = r + \frac{1}{\alpha}\mathbb{E}[(r+z)^-]\}$ of first stage solutions is just the interval $[-\bar{q}_\alpha(z), -q_\alpha(z)]$, i.e., the set of all negative α-quantiles of z. Indeed, Proposition 2.9 is inspired by the latter result.

Example 2.11. Consider the *expected regret* or *expected loss* defined by

$$EL(z) = \mathbb{E}\left[(z - \gamma)^-\right]$$

with some fixed target $\gamma \in \mathbb{R}$. This functional, too, can be written in the form (2.1) with $k_1 = 1$, $k_2 = 2$, $w_1 = 1$, $c_1 = 0$, $c_2 = (0,1)$, $w_2 = (1,-1)$, $Y_1 = \{\gamma\}$, $Y_2 = \mathbb{R}_+ \times \mathbb{R}_+$. Note that, actually, Y_1 is *not* a cone here. Further, $\langle w_2, Y_2\rangle = \mathbb{R}$, $D_{\rho,1} \cap D_{\rho,2} \neq \emptyset$, and $D_{\rho,2} = [0,1] \subseteq \mathbb{R}_+$, thus the dual representations (2.2) and (2.3) hold and ρ is consistent with second order stochastic dominance. However, ρ is not translation invariant, i.e., not a risk measure in the sense of [10, 11]. Nevertheless, it is used as a risk measure in some applications.

Example 2.12. The utilization of deviation and semideviation measures in stochastic optimization goes back to [14] and is further discussed, e.g., in [17, 18, 19, 28]. For $k \geq 1$ deviation and semideviation are defined by

$$D_k(z) := \left(\mathbb{E}\left[|z - \mathbb{E}[z]|^k\right]\right)^{1/k} \qquad SD_k(z) := \left(\mathbb{E}\left[\left((z - \mathbb{E}[z])^-\right)^k\right]\right)^{1/k},$$

respectively. They are closely related to coherent risk measures (cf. [28]), $-\mathbb{E} + \beta \cdot D_k$ and $-\mathbb{E}+\beta\cdot SD_k$ with $\beta \geq 0$ are translation invariant in the sense of [10, 11] and, hence, candidates for coherent risk measure. However, they are *not* within the framework of polyhedral risk measures, even $SD_1 = \frac{1}{2}D_1$ cannot be written in the form (2.1). But, if we change from expectation $\mathbb{E}[z]$ to the median $q_{\frac{1}{2}}(z)$, then we obtain the median-deviation which is a special case of the so-called dispersion measures at level $\alpha \in (0,1)$ given by

$$d_\alpha(z) := \mathbb{E}\left[\alpha(z - q_\alpha)^+ + (1-\alpha)(z - q_\alpha)^-\right] \qquad d_{\frac{1}{2}}(z) = \tfrac{1}{2}\mathbb{E}\left[\left|z - q_{\frac{1}{2}}(z)\right|\right]$$

(cf. [19, 40]). These functionals are polyhedral with $k_1 = 1$, $k_2 = 2$, $c_1 = 0$, $c_2 = (\alpha, 1-\alpha)$, $w_1 = 1$, $w_2 = (1,-1)$, $Y_1 = \mathbb{R}$, and $Y_1 = \mathbb{R}_+ \times \mathbb{R}_+$. Again, $\rho := -\mathbb{E}+\beta\cdot d_\alpha$ is a candidate for a coherent risk measure. According to Remark 2.3 also ρ is polyhedral with $c_1 = -1$, $c_2 = (\alpha\beta-1, (1-\alpha)\beta+1)$, and w_t and Y_t as above. Hence, $D_{\rho,1} = \{1\}$, $D_{\rho,2} = [1-\alpha\beta, 1+(1-\alpha)\beta]$, and $\Lambda_\rho = \{\lambda : \mathbb{E}[\lambda] = 1, \lambda \in [1-\alpha\beta, 1+(1-\alpha)\beta] \text{ a.s.}\}$, i.e., ρ is coherent and second order stochastic dominance consistent if $\beta \leq \frac{1}{\alpha}$ (see also [19]). However, the latter representation reveals that $\rho = -(1-\alpha\beta)\mathbb{E} + \alpha\beta \cdot CVaR_\alpha$, i.e., quantile dispersion and Conditional-Value-at-Risk is basically the same thing.

3. Multiperiod risk. When random variables $z_1, \ldots, z_T$ with $z_t \in L_p(\Omega, \mathcal{F}_t, \mathbb{P})$, $p \geq 1$, are considered and the available information is revealed with the passing of time, it may become necessary to use multiperiod risk measures (see [3, 2, 22, 25, 42, 36]). We assume that a filtration of σ-fields $\mathcal{F}_t$, $t = 1, \ldots, T$, is given, i.e., $\mathcal{F}_t \subseteq \mathcal{F}_{t+1} \subseteq \mathcal{F}$, and that $\mathcal{F}_1 = \{\emptyset, \Omega\}$, i.e., that z_1 is always deterministic. We will now generalize the concepts of the previous section to this multiperiod framework.

Remark 3.1. When dealing with multiperiod risk measures one has to determine whether the random variables represent (potentially financial) *incomes* or *payments* as, e.g., in [22, 36, 42], or if they have to be understood in a cumulative sense, i.e., as a *wealth* or *value process* as in [3, 2]. Of course, the one can easily be transformed into the other: If Z_t is an income, then one can consider accumulation $z_t = Z_1 + \cdots + Z_t$, and if z_t is an accumulated value, then the income is given by $Z_t = z_t - z_{t-1}$. Throughout this paper we consider $z = (z_1, \ldots, z_T)$ to be a value process.

We give the definition of coherence in the multiperiod case as introduced[2] in [3, 2].

[2] In [3, 2] the definition is slightly different since another framework was considered: The first time stage (i.e., the deterministic stage) was denoted by index 0. Here, the formulation is adapted to our framework with index 1 for the deterministic time stage (i.e., $\mathcal{F}_1 = \{\emptyset, \Omega\}$).

Definition 3.2. *A functional ρ on $\times_{t=1}^T L_p(\Omega, \mathcal{F}_t, \mathbb{P})$ is called a* multiperiod coherent risk measure *if the following:*

(i) *if $z_t \le \tilde{z}_t$ a.s., $t = 1, \dots, T$, then $\rho(z_1, \dots, z_T) \ge \rho(\tilde{z}_1, \dots, \tilde{z}_T)$* (monotonicity)*;*

(ii) *for each $r \in \mathbb{R}$ we have $\rho(z_1+r, \dots, z_T+r) = \rho(z) - r$* (translation invariance)*;*

(iii) *$\rho(\mu z_1 + (1-\mu)\tilde{z}_1, \dots, \mu z_T + (1-\mu)\tilde{z}_T) \le \mu\rho(z_1, \dots, z_T) + (1-\mu)\rho(\tilde{z}_1, \dots, \tilde{z}_1)$ for $\mu \in [0,1]$* (convexity)*;*

(iv) *for $\mu \ge 0$ we have $\rho(\mu z_1, \dots, \mu z_T) = \mu\rho(z_1, \dots, z_T)$* (positive homogeneity).

Remark 3.3. How translation invariance is to be defined in the multiperiod case is still subject to discussion in the ongoing research in financial mathematics. Different suggestions were made, e.g., in [36, 25, 42] such that nonrandom amounts can be shifted in time by means of credits. However, from the viewpoint of capital requirement and optimization it appears reasonable to keep with [3, 2].

Example 3.4. In [3, Example 3] it was shown that $\rho(z) := -\mathbb{E}[\min\{z_1, \dots, z_T\}]$ with $z = (z_1, \dots, z_T)$ is a multiperiod coherent risk measure on $\times_{t=1}^T L_\infty(\Omega, \mathcal{F}_t, \mathbb{P})$.

Remark 3.5. Let ρ_t be (one-period) coherent risk measures on $L_p(\Omega, \mathcal{F}_t, \mathbb{P})$, $t = 1, \dots, T$. Let further $\emptyset \ne S \subseteq \{1, \dots, T\}$. Then $\rho(z) := \max_{t \in S} \rho_t(z_t)$ is multiperiod coherent. Let $\mu_t \in \mathbb{R}_+$, $t = 1, \dots, T$, with $\sum_{t=1}^T \mu_t = 1$. Then also $\rho(z) := \sum_{t=1}^T \mu_t \rho_t(z_t)$ is a multiperiod coherent risk measure. This can easily be verified by checking the four properties of Definition 3.2.

As shown in [3, 2], the representation result for (one-period) risk measures (Theorem 2.1) can be carried over to the multiperiod case. Therefore, the set of densities $\mathcal{D}$ is extended such that the integrals of the time steps sum up to one,

$$\mathcal{D}_T := \left\{ f \in \times_{t=1}^T L_1(\Omega, \mathcal{F}_t, \mathbb{P}) : f_t \ge 0 \text{ a.s. } (t = 1, \dots, T),\ \textstyle\sum_{t=1}^T \mathbb{E}\left[f_t\right] = 1 \right\}.$$

Theorem 3.6. *Let $\rho : \times_{t=1}^T L_p(\Omega, \mathcal{F}_t, \mathbb{P}) \to \bar{\mathbb{R}}$ and assume that ρ is lower semicontinuous. Then ρ is a multiperiod coherent risk measure if and only if the following condition holds:*

$$\exists\, \mathcal{P}_\rho \subseteq \mathcal{D}_T \text{ convex} : \rho(z) = \sup\left\{ \textstyle\sum_{t=1}^T \mathbb{E}\left[-z_t f_t\right] : f \in \mathcal{P}_\rho \right\}. \tag{3.1}$$

Proof. We follow the ideas of [3, 2], but in reverse order. Obviously, ρ is coherent if and only if the corresponding one-period risk measure ρ' on $L_p(\Omega', \mathcal{F}', \mathbb{P}')$ is coherent in the usual sense, where $(\Omega', \mathcal{F}', \mathbb{P}')$ and ρ' are defined as follows:

$$\begin{aligned}
&\Omega' := \Omega \times \{1, \dots, T\} \\
&\mathcal{F}' := \left\{ \textstyle\bigcup_{t=1}^T (A_t \times \{t\}) : A_t \in \mathcal{F}_t \right\} \\
&\mathbb{P}'\left(\textstyle\bigcup_{t=1}^T (A_t \times \{t\}) \right) := \tfrac{1}{T} \textstyle\sum_{t=1}^T \mathbb{P}(A_t) \\
&\rho'(z') := \rho\left(z(z')\right)
\end{aligned}$$

and $z(z')$ is defined by $z(z')(\omega) := (z'(\omega, 1), z'(\omega, 2), \dots, z'(\omega, T))$. Theorem 2.1 says that there exists a convex set of density functions $\mathcal{P}'_\rho \subseteq \mathcal{D}$ such that, for $z \in \times_{t=1}^T L_p(\Omega, \mathcal{F}_t, \mathbb{P})$,

$$\rho(z) = \rho'(z'(z)) = \sup\left\{ \mathbb{E}'\left[-z' f'\right] : f' \in \mathcal{P}'_\rho \right\}$$

with $z'(z)(\omega, t) := z_t(\omega)$. Note that also the conditions from Definition 3.2 are equivalent to those from Theorem 2.1 for $(\Omega', \mathcal{F}', \mathbb{P}')$ and that lower semicontinuity of ρ is equivalent to lower semicontinuity of ρ'. By setting

$$\mathcal{P}_\rho := \left\{ f = \left(\tfrac{1}{T} f'(.,1), \tfrac{1}{T} f'(.,2), \ldots, \tfrac{1}{T} f'(.,T) \right) : f' \in \mathcal{P}'_\rho \right\},$$

the assertion follows. □

Now we are ready to extend Definition 2.2 to the multiperiod case.

Definition 3.7. *A multiperiod risk measure ρ on $\times_{t=1}^{T} L_p(\Omega, \mathcal{F}_t, \mathbb{P})$ with $p \in [1,\infty]$ is called* multiperiod polyhedral *if there are $k_t \in \mathbb{N}$, $c_t \in \mathbb{R}^{k_t}$, $t = 1, \ldots, T$, $w_{t\tau} \in \mathbb{R}^{k_{t-\tau}}$, $t = 1, \ldots, T$, $\tau = 0, \ldots, t-1$, a polyhedral set $Y_1 \subseteq \mathbb{R}^{k_1}$, and polyhedral cones $Y_t \subseteq \mathbb{R}^{k_t}$, $t = 2, \ldots, T$, such that*

$$\rho(z) = \inf \left\{ \mathbb{E}\left[\sum_{t=1}^{T} \langle c_t, y_t \rangle \right] \;\middle|\; \begin{array}{ll} y_t \in L_p(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}^{k_t}), & \\ y_t \in Y_t \ \text{a.s.}, & (t = 1, \ldots, T) \\ \sum_{\tau=0}^{t-1} \langle w_{t,\tau}, y_{t-\tau} \rangle = z_t \ \text{a.s.} & \end{array} \right\}. \tag{3.2}$$

Remark 3.8. The reader might wonder why, for $T = 2$, this definition does not precisely coincide with the Definition 2.2 for the one-period case. This is due to the fact that, in the literature, the risk of a process $z_1, \ldots, z_T$ is allowed to depend also on z_1 although this value is constant, i.e., deterministic (see [3, 2, 25]), whereas one-period risk depends on one scalar random variable only. Nevertheless, the one-period case can be regarded as a special case of Definition 3.7 because for $T = 2$ the parameters Y_1, c_1, and $w_{1,0}$ can easily be chosen such that z_1 does not contribute to the optimal value of (3.2).

Theorem 3.9. *Let ρ be a functional of the form* (3.2) *on $\times_{t=1}^{T} L_p(\Omega, \mathcal{F}_t, \mathbb{P})$ with $p \in [1, \infty)$. Assume*

(i) *complete recourse: $\langle w_{t,0}, Y_t \rangle = \mathbb{R}$ $(t = 1, \ldots, T)$,*

(ii) *dual feasibility: $\{u \in \mathbb{R}^T : c_t + \sum_{\nu=t}^{T} u_\nu w_{\nu,\nu-t} \in -Y_t^* \ (t = 1, \ldots, T)\} \neq \emptyset$.*

Then ρ is finite, convex, and continuous on $\times_{t=1}^{T} L_p(\Omega, \mathcal{F}_t, \mathbb{P})$ and with $\frac{1}{p} + \frac{1}{p'} = 1$ the following dual representation holds:

$$\begin{aligned} &\rho(z) \\ &= \sup \left\{ \begin{array}{l} \inf\limits_{y_1 \in Y_1} \left\langle c_1 + \sum_{\nu=1}^{T} \mathbb{E}\left[\lambda_\nu\right] w_{\nu,\nu-1}, y_1 \right\rangle \\ - \mathbb{E}\left[\sum_{t=1}^{T} \lambda_t z_t \right] \end{array} \;\middle|\; \begin{array}{l} \lambda_t \in L_{p'}(\Omega, \mathcal{F}_t, \mathbb{P}) \ (t = 1, \ldots, T), \\ c_t + \sum_{\nu=t}^{T} \mathbb{E}\left[\lambda_\nu | \mathcal{F}_t\right] w_{\nu,\nu-t} \in -Y_t^* \\ \text{a.s.} \ (t = 2, \ldots, T) \end{array} \right\}. \end{aligned} \tag{3.3}$$

If, in addition, Y_1 is a polyhedral cone, then ρ is positively homogeneous and (3.3) *simplifies to*

$$\rho(z) = \sup \left\{ -\mathbb{E}\left[\sum_{t=1}^{T} \lambda_t z_t \right] \;\middle|\; \begin{array}{l} \lambda_t \in L_{p'}(\Omega, \mathcal{F}_t, \mathbb{P}), \\ c_t + \sum\limits_{\nu=t}^{T} \mathbb{E}\left[\lambda_\nu | \mathcal{F}_t\right] w_{\nu,\nu-t} \in -Y_t^* \ \text{a.s.} \\ (t = 1, \ldots, T) \end{array} \right\}. \tag{3.4}$$

Proof. We use results on conjugate duality (see [26] and [5, section 2.5.1]). Consider the Banach spaces and their duals

$$\begin{array}{ll} E := \times_{t=1}^{T} L_p(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}^{k_t}) & E^* = \times_{t=1}^{T} L_{p'}(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}^{k_t}) \\ Z := \times_{t=1}^{T} L_p(\Omega, \mathcal{F}_t, \mathbb{P}) & Z^* = \times_{t=1}^{T} L_{p'}(\Omega, \mathcal{F}_t, \mathbb{P}) \end{array}$$

with bilinear forms $\langle e, e^*\rangle_{E/E^*} = \sum_{t=1}^T \mathbb{E}[\langle e_t, e_t^*\rangle_{\mathbb{R}^{k_t}}]$ and $\langle z, z^*\rangle_{Z/Z^*} = \sum_{t=1}^T \mathbb{E}[z_t z_t^*]$, respectively. Due to the complete recourse assumption it holds that $\rho(z) < \infty$ for all $z = (z_1, \dots, z_T) \in E$. Define $Y := \{y \in E : y_t(\omega) \in Y_t \ (t = 1, \dots, T) \text{ for a.e. } \omega \in \Omega\}$, $K = \sum_{t=1}^T k_t$ and

$$\begin{aligned} \varphi : E \times Z &\to \bar{\mathbb{R}} \\ (y, z) &\mapsto \varphi(y, z) := \langle y, c\rangle_{E/E^*} + \delta_Y(y) + \delta_{\{0\}}(Wy - z) \end{aligned}$$

with δ denoting the indicator function (taking values 0 and $+\infty$ only) and with

$$c = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_T \end{pmatrix} \in \mathbb{R}^K \qquad W = \begin{pmatrix} w'_{1,0} & 0 & 0 & \cdots & 0 \\ w'_{2,1} & w'_{2,0} & 0 & \cdots & 0 \\ w'_{3,2} & w'_{3,1} & w'_{3,0} & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ w'_{T,T-1} & w'_{T,T-2} & w'_{T,T-3} & \cdots & w'_{T,0} \end{pmatrix} \in \mathbb{R}^{T \times K}.$$

Note that φ is proper, lower semicontinuous, and convex since Y is convex. With these notations Definition 3.7 reads $\rho(z) = \inf_{y \in E} \varphi(y, z)$ and due to [5, Proposition 2.143] ρ is convex. The (conjugate) dual problem according to [5] is given by

$$\rho^*(z) = \sup\Big\{\langle z, z^*\rangle_{Z/Z^*} - \varphi^*(0, z^*) : z^* \in Z^*\Big\} \tag{3.5}$$

in which the conjugate φ^* is given by

$$\begin{aligned} \varphi^*(y^*, z^*) &= \sup\Big\{\langle y, y^*\rangle_{E/E^*} + \langle z, z^*\rangle_{Z/Z^*} - \varphi(y, z) : y \in E,\, z \in Z\Big\} \\ &= \sup\Big\{\langle y, y^* - c\rangle_{E/E^*} + \langle z, z^*\rangle_{Z/Z^*} : y \in Y,\, z = Wy \text{ a.s.}\Big\} \\ &= \sup\Big\{\langle y, y^* - c\rangle_{E/E^*} + \langle Wy, z^*\rangle_{Z/Z^*} : y \in Y\Big\} \\ &= \sup\Big\{\langle y, y^* - c + W^* z^*\rangle_{E/E^*} : y \in Y\Big\} \end{aligned}$$

with $W^* : Z^* \to E^*$ denoting the adjoint operator implicitly defined by the relation $\langle Wy, z^*\rangle_{Z/Z^*} = \langle y, W^* z^*\rangle_{E/E^*}$ for $y \in E$, $z^* \in Z^*$. Thereby, the matrix W is understood as mapping from E to Z. For the adjoint operator W^* it holds that

$$\begin{aligned} \langle y, W^* z^*\rangle_{E/E^*} = \langle Wy, z^*\rangle_{Z/Z^*} &= \textstyle\sum_{t=1}^T \mathbb{E}\left[z_t^* \sum_{\tau=0}^{t-1} \langle w_{t,\tau}, y_{t-\tau}\rangle_{\mathbb{R}^{k_{t-\tau}}}\right] \\ &= \mathbb{E}\left[\textstyle\sum_{t=1}^T \sum_{\tau=0}^{t-1} \langle z_t^* w_{t,\tau}, y_{t-\tau}\rangle_{\mathbb{R}^{k_{t-\tau}}}\right] \\ &= \mathbb{E}\left[\textstyle\sum_{s=1}^T \sum_{\nu=s}^{T} \langle z_\nu^* w_{\nu,\nu-s}, y_s\rangle_{\mathbb{R}^{k_s}}\right] \\ &= \textstyle\sum_{s=1}^T \mathbb{E}\left[\left\langle \sum_{\nu=s}^T z_\nu^* w_{\nu,\nu-s}, y_s\right\rangle_{\mathbb{R}^{k_s}}\right] \\ &= \textstyle\sum_{s=1}^T \mathbb{E}\left[\left\langle \sum_{\nu=s}^T \mathbb{E}\left[z_\nu^* | \mathcal{F}_s\right] w_{\nu,\nu-s}, y_s\right\rangle_{\mathbb{R}^{k_s}}\right], \end{aligned}$$

hence $W^* z^* = (\sum_{\nu=1}^T \mathbb{E}[z_\nu^*] w_{\nu,\nu-1}, \sum_{\nu=2}^T \mathbb{E}[z_\nu^* | \mathcal{F}_2] w_{\nu,\nu-2}, \dots, z_T^* w_{T,0}) \in E^*$. Utilizing the fact that Y_t are cones for $t = 2, \dots, T$ results in

$$\begin{aligned}\rho^*(z) &= \sup\Big\{\langle z, z^*\rangle_{Z/Z^*} - \sup\Big\{\langle y, W^*z^* - c\rangle_{E/E^*} : y \in Y\Big\} : z^* \in Z^*\Big\}\\ &= \sup\Big\{\langle z, z^*\rangle_{Z/Z^*} + \inf\Big\{\langle y, c - W^*z^*\rangle_{E/E^*} : y \in Y\Big\} : z^* \in Z^*\Big\}\\ &= \sup\left\{\begin{array}{l|l} \langle z, z^*\rangle_{Z/Z^*} + & z^* \in Z^*, \\ \inf\limits_{y_1 \in Y_1} \Big\langle y_1, c_1 - \sum_{t=1}^T \mathbb{E}\left[z_t^*\right] w_{t,t-1}\Big\rangle & c_t - \sum_{\nu=t}^T \mathbb{E}\left[z_\nu^*|\mathcal{F}_t\right] w_{\nu,\nu-t} \in -Y_t^* \\ & \text{a.s. } (t = 2, \dots, T)\end{array}\right\}\end{aligned}$$

and this is exactly (3.3) with $\lambda = -z^*$. Weak duality holds (cf. [5, section 2.5.1]), i.e., $\rho^*(z) \le \rho(z)$, and dual feasibility ensures $\rho^*(z) > -\infty$, hence

$$-\infty < \rho^*(z) \le \rho(z) < +\infty \quad \forall\, z \in Z.$$

Now, [5, Proposition 2.152] provides that $\rho(z)$ is continuous. In turn, for this case [5, Theorem 2.151] guarantees strong duality, i.e., $\rho^*(z) = \rho(z)$. □

As for the one-period case, we define the set of dual multipliers by

$$\Lambda_\rho := \left\{\lambda \in \times_{t=1}^T L_{p'}(\Omega, \mathcal{F}_t, \mathbb{P}) \;\middle|\; \begin{array}{l} c_t + \sum_{\nu=t}^T \mathbb{E}\left[\lambda_\nu|\mathcal{F}_t\right] w_{\nu,\nu-t} \in -Y_t^* \text{ a.s.} \\ (t = 1, \dots, T)\end{array}\right\}. \tag{3.6}$$

Again, comparing the dual representations (3.1) and (3.4) provides a criterion for a polyhedral functional to be a multiperiod coherent risk measure.

Corollary 3.10. *Let ρ be a functional on $\times_{t=1}^T L_p(\Omega, \mathcal{F}_t, \mathbb{P})$ of the form* (3.2) *with Y_1 being a polyhedral cone. Let the conditions of Theorem* 3.9 *be satisfied (complete recourse, dual feasibility) and assume $\Lambda_\rho \subseteq \mathcal{D}_T$. Then ρ is a multiperiod coherent risk measure.*

Proof. Analogously to Corollary 2.5, the assertion here is an immediate consequence of Theorems 3.6 and 3.9 since $\mathcal{P}_\rho := \Lambda_\rho$ does the job. □

Example 3.11. A straightforward approach to incorporate risk in terms of the Conditional-Value-at-Risk at all time stages consists in considering a weighted sum

$$\rho_1(z) := \sum_{t=2}^T \gamma_t CVaR_{\alpha_t}(z_t)$$

with some weights $\gamma_t \ge 0$ (e.g., $\gamma_t = \frac{1}{T-1}$) and some confidence levels $\alpha_2, \alpha_3, \dots, \alpha_T \in (0, 1)$. Note that

$$\begin{aligned}\rho_1(z) &= \textstyle\sum_{t=2}^T \gamma_t \inf_{r_t \in \mathbb{R}} \Big\{r_t + \frac{1}{\alpha_t}\mathbb{E}\Big[(z_t + r_t)^-\Big]\Big\}\\ &= \textstyle\inf_{(r_2,\dots,r_T)\in\mathbb{R}^{T-1}} \Big\{\sum_{t=2}^T \gamma_t \Big(r_t + \frac{1}{\alpha_t}\mathbb{E}\Big[(z_t + r_t)^-\Big]\Big)\Big\}\\ &= \inf\left\{\textstyle\sum_{t=2}^T \gamma_t \Big(y_1^{(t)} + \frac{1}{\alpha_t}\mathbb{E}\Big[y_t^{(2)}\Big]\Big) \;\middle|\; \begin{array}{l} y_1 \in \mathbb{R}^T,\ y_1^{(1)} = z_1, \\ y_t \in L_1(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}^2), \\ y_t^{(1)} - y_t^{(2)} = z_t + r_t \text{ a.s.}, \\ y_t \in \mathbb{R}_+ \times \mathbb{R}_+ \text{ a.s. } (t = 2, \dots, T)\end{array}\right\}\end{aligned}$$

(set $y_1^{(t)} = r_t$), i.e., ρ_1 is of the form (3.2) with $k_1 = T$, $k_t = 2$ $(t = 2, \dots, T)$, $c_1 = (0, \gamma_2, \dots, \gamma_T)$, $c_t = (0, \frac{\gamma_t}{\alpha_t})$ $(t = 2, \dots, T)$, $w_{1,0} = e_1$, $w_{t,0} = (1, -1)$ $(t = 2, \dots, T)$, $w_{t,t-1} = -e_t$ $(t = 2, \dots, T)$, $w_{t,\tau} = 0$ $(\tau = 1, \dots, t-2,\ t = 3, \dots, T)$, $Y_1 = \mathbb{R}^T$, $Y_t = \mathbb{R}_+ \times \mathbb{R}_+$ $(t = 2, \dots, T)$ (with e_t denoting the tth standard basis vector in $\mathbb{R}^T$).

Thus, the risk measure ρ_1 is *multiperiod polyhedral.* Due to Remark 3.5 it is *multiperiod coherent*, too, if $\sum_{t=2}^T \gamma_t = 1$. This can also be seen by means of Corollary 3.10. The set of feasible multipliers is given here by

$$\Lambda_{\rho_1} = \left\{ \lambda \in \times_{t=1}^T L_{p'}(\Omega, \mathcal{F}_t, \mathbb{P}) \;\middle|\; \begin{array}{l} \lambda_1 = 0, \\ 0 \le \lambda_t \le \frac{\gamma_t}{\alpha_t} \text{ a.s. } (t = 2, \dots, T), \\ \mathbb{E}\left[\lambda_t\right] = \gamma_t \end{array} \right\} \tag{3.7}$$

and, of course, $\Lambda_{\rho_1} \subseteq \mathcal{D}_T$. Moreover, the conditions of Theorem 3.9 are satisfied, i.e., complete recourse and dual feasibility hold (take $u = (0, \gamma_2, \dots, \gamma_T)$).

Next we present more involved examples, which extend the Conditional-Value-at-Risk to the multiperiod situation. The characteristic thing about $CVaR$ is that, in the dual representation, the density functions, i.e., the Lagrangian multipliers are bounded pointwise from above (cf. Example 2.10). This idea will be found somehow in all of the following examples.

Example 3.12. In this example, we define a multiperiod coherent risk measure where not every time step contributes with a fixed weight. When looking at the dual representation (3.3) and at Corollary 3.10, it becomes obvious that each of the dual constraints $c_t + \sum_{\nu=t}^T \mathbb{E}[\lambda_\nu|\mathcal{F}_t] w_{\nu,\nu-t} \in -Y_t^*$ has to imply $\lambda_t \ge 0$ for $t = 1, \dots, T$. A natural candidate for implying $\sum_{\nu=1}^T \mathbb{E}[\lambda_\nu] = 1$ is the corresponding constraint for $t = 1$, which reads $c_1 + \sum_{\nu=1}^T \mathbb{E}[\lambda_\nu] w_{\nu,\nu-1} \in -Y_1^*$.

Now, setting $k_t = 2$ $(t = 1, \dots, T)$, $c_1 = (1, 0)$, $c_t = (0, \beta_t)$ with some $\beta_t > 0$ $(t = 2, \dots, T)$ such that $\sum_{t=2}^T \beta_t \ge 1$, $w_{1,0} = (0, 1)$, $w_{t,0} = (1, -1)$ $(t = 1, \dots, T)$, $w_{t,t-1} = (-1, 0)$ $(t = 2, \dots, T)$, and $w_{t,\tau} = 0$ $(\tau = 1, \dots, t-2,\ t = 3, \dots, T)$, $Y_1 = \mathbb{R} \times \mathbb{R}$, $Y_t = \mathbb{R}_+ \times \mathbb{R}_+$ $(t = 2, \dots, T)$ leads to

$$\begin{array}{lcl} c_1 + \sum_{\nu=1}^T \mathbb{E}\left[\lambda_\nu\right] w_{\nu,\nu-1} \in -Y_1^* & \iff & \lambda_1 = 0 \quad \text{and} \quad \sum_{\nu=1}^T \mathbb{E}\left[\lambda_\nu\right] = 1, \\ c_t + \sum_{\nu=t}^T \mathbb{E}\left[\lambda_\nu|\mathcal{F}_t\right] w_{\nu,\nu-t} \in -Y_t^* & \iff & 0 \le \lambda_t \quad \text{and} \quad \lambda_t \le \beta_t \ (t = 2, \dots, T) \end{array}$$

since $Y_1^* = \{0\} \times \{0\}$ and $Y_t^* = \mathbb{R}_- \times \mathbb{R}_-$ $(t = 2, \dots, T)$. Hence, the dual set Λ_{ρ_2} is of the form

$$\Lambda_{\rho_2} = \left\{ \lambda \in \times_{t=1}^T L_{p'}(\Omega, \mathcal{F}_t, \mathbb{P}) \;\middle|\; \begin{array}{l} \lambda_1 = 0, \\ 0 \le \lambda_t \le \beta_t \text{ a.s. } (t = 2, \dots, T), \\ \sum_{t=1}^T \mathbb{E}[\lambda_t] = 1 \end{array} \right\}. \tag{3.8}$$

Note that complete recourse and dual feasibility hold. Thus, Corollary 3.10 implies that the functional

$$\rho_2(z) := \inf \left\{ y_1^{(1)} + \sum_{t=2}^T \beta_t \mathbb{E}\left[y_t^{(2)}\right] \;\middle|\; \begin{array}{l} y_t \in L_p(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}^2) \ (t = 1, \dots, T), \\ y_1 \in \mathbb{R} \times \mathbb{R},\ y_t \in \mathbb{R}_+ \times \mathbb{R}_+ \text{ a.s. } (t = 2, \dots, T), \\ y_1^{(2)} = z_1, \\ y_t^{(1)} - y_t^{(2)} = z_t + y_1^{(1)} \text{ a.s. } (t = 2, \dots, T) \end{array} \right\}$$

or simply $\rho_2(z) = \inf_{r \in \mathbb{R}} \{r + \sum_{t=2}^T \beta_t \mathbb{E}[(z_t + r)^-]\}$ is a *multiperiod polyhedral* and *coherent risk measure.*

The remaining examples present multiperiod polyhedral coherent risk measures that depend on the filtration $\{\mathcal{F}_t\}_{t=1}^T$, i.e., on the information flow over time.

Example 3.13. To incorporate the information structure we adapt the previous example in such a manner that successive time steps are associated. We choose everything as before, only the assignment $w_{t,\tau} = 0$ $(\tau = 1, \ldots, t-2,\ t = 3, \ldots, T)$ is replaced by $w_{t,1} = (0, -1)$ $(t = 3, \ldots, T)$ and $w_{t,\tau} = 0$ $(\tau = 2, \ldots, t-2,\ t = 4, \ldots, T)$. In addition, we set $c_t = (0, \delta_t)$ with $\delta_t > 0$ for $t = 2, \ldots, T$. Hence, the dual set Λ_{ρ_3} is of the form

$$\Lambda_{\rho_3} = \left\{ \lambda \in \times_{t=1}^{T} L_{p'}(\Omega, \mathcal{F}_t, \mathbb{P}) \;\middle|\; \begin{array}{l} \lambda_1 = 0,\ \sum_{t=1}^{T} \mathbb{E}[\lambda_t] = 1, \\ 0 \le \lambda_t,\ \lambda_t + \mathbb{E}[\lambda_{t+1} | \mathcal{F}_t] \le \delta_t \text{ a.s.} \\ (t = 2, \ldots, T-1), \\ 0 \le \lambda_T \le \delta_T \text{ a.s.} \end{array} \right\}. \tag{3.9}$$

Again, the complete recourse condition is satisfied and dual feasibility holds if the parameters δ_t are chosen sufficiently large. Altogether, Corollary 3.10 implies that the functional

$$\rho_3(z) := \inf \left\{ y_1^{(1)} + \sum_{t=2}^{T} \delta_t \mathbb{E}\left[y_t^{(2)} \right] \;\middle|\; \begin{array}{l} y_t \in L_p(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}^2)\ (t = 1, \ldots, T), \\ y_1 \in \mathbb{R} \times \mathbb{R},\ y_t \in \mathbb{R}_+ \times \mathbb{R}_+ \text{ a.s. } (t = 2, \ldots, T), \\ y_1^{(2)} = z_1, \\ y_2^{(1)} - y_2^{(2)} = z_2 + y_1^{(1)} \text{ a.s.}, \\ y_t^{(1)} - y_t^{(2)} = z_t + y_1^{(1)} + y_{t-1}^{(2)} \text{ a.s. } (t = 3, \ldots, T) \end{array} \right\}$$

is a *polyhedral multiperiod coherent risk measure.*

Example 3.14. In this approach, the concatenation of the time steps is even stronger than in the previous example. We set $k_t = 2$ $(t = 1, \ldots, T)$, $c_1 = (\frac{1}{T-1}, 0)$, $c_t = (0, \mu_t)$ $(t = 2, \ldots, T)$ with some numbers $\frac{1}{T-1} < \mu_2 \le \mu_3 \le \cdots \le \mu_T$, $w_{1,0} = (0, 1)$, $w_{t,0} = (1, -1)$ $(t = 2, \ldots, T)$, $w_{t,1} = (-1, 0)$ $(t = 2, \ldots, T)$, $w_{t,\tau} = 0$ for $\tau > 1$, $Y_1 = \mathbb{R} \times \mathbb{R}$, $Y_t = \mathbb{R} \times \mathbb{R}_+$ $(t = 2, \ldots, T-1)$, $Y_T = \mathbb{R}_+ \times \mathbb{R}_+$.

The dual constraints $c_t + \sum_{\nu=t}^{T} \mathbb{E}[\lambda_\nu | \mathcal{F}_t] w_{\nu,\nu-t} \in -Y_t^*$ imply that λ has to be a *martingale* with respect to the filtration $(\mathcal{F}_t)_{t=1}^{T}$. This implies $\mathbb{E}[\lambda_2] = \cdots = \mathbb{E}[\lambda_T]$ and $\lambda_t \ge 0$ since $\lambda_T \ge 0$. Together with (3.6) we obtain

$$\Lambda_{\rho_4} = \left\{ \lambda \in \times_{t=1}^{T} L_{p'}(\Omega, \mathcal{F}_t, \mathbb{P}) \;\middle|\; \begin{array}{l} \lambda_1 = 0, \\ 0 \le \lambda_t \le \mu_t \text{ a.s. } (t = 2, \ldots, T), \\ \lambda_t = \mathbb{E}\left[\lambda_{t+1} | \mathcal{F}_t\right]\ (t = 2, \ldots, T-1), \\ \mathbb{E}\left[\lambda_2\right] = \cdots = \mathbb{E}\left[\lambda_T\right] = \frac{1}{T-1} \end{array} \right\}. \tag{3.10}$$

Complete recourse is satisfied and dual feasibility holds since the vector $u \in \mathbb{R}^T$ with $u_1 = 0$ and $u_t = \frac{1}{T-1}$ for $t = 2, \ldots, T$ defines a (constant) element of Λ_{ρ_4}. Hence, Corollary 3.10 applies and the resulting functional

$$\rho_4(z) := \inf \left\{ \tfrac{1}{T-1} y_1^{(1)} + \sum_{t=2}^{T} \mu_t \mathbb{E}\left[y_t^{(2)} \right] \;\middle|\; \begin{array}{l} y_t \in L_p(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}^2)\ (t = 1, \ldots, T), \\ y_1 \in \mathbb{R} \times \mathbb{R},\ y_T \in \mathbb{R}_+ \times \mathbb{R}_+ \text{ a.s.}, \\ y_t \in \mathbb{R} \times \mathbb{R}_+ \text{ a.s. } (t = 2, \ldots, T-1), \\ y_1^{(2)} = z_1, \\ y_t^{(1)} - y_t^{(2)} = z_t + y_{t-1}^{(1)} \text{ a.s. } (t = 2, \ldots, T) \end{array} \right\}$$

is a polyhedral *multiperiod coherent risk measure.*

Comparing (3.10) for $\mu_t = \frac{1}{(T-1)\alpha}$ with the dual representation of the Conditional-Value-at-Risk (2.9) it turns out that the multiperiod risk measure ρ_4 defined in this way is a kind of canonical extension of the Conditional-Value-at-Risk in terms of [3, sections 4 and 5] and of [25].[3]

The next example is motivated from the viewpoint of the value of information (cf. [21, 22]).

Example 3.15. In [22], the following multiperiod risk measure was suggested. Given some constants $b_T = 0 \leq d \leq b_{T-1} \leq \cdots \leq b_2 \leq b_1$ and $b_{t-1} \leq q_t$ for $t = 2, \ldots, T$, this risk measure is defined[4] on $\times_{t=1}^{T} L_p(\Omega, \mathcal{F}_t, \mathbb{P})$ by

$$\rho_5(Z) = -\sup \left\{ \begin{array}{l} \mathbb{E}\left[b_1 A_1 + \sum_{t=2}^{T-1} (b_t A_t - q_t M_t) + d K_T - q_T M_T \right] : \\ A_t \in L_p(\Omega, \mathcal{F}_t, \mathbb{P}) \quad (t = 1, \ldots, T), \\ K_t = [K_{t-1} + Z_t - A_{t-1}]^+ \quad (t = 2, \ldots, T), \\ M_t = [K_{t-1} + Z_t - A_{t-1}]^- \quad (t = 2, \ldots, T) \end{array} \right\}$$

with $K_1 := 0$. However, in [22] $Z = (Z_1, \ldots, Z_T)$ is understood as income process with $Z_1 = 0$, thus this definition does not fit in our framework.

Therefore, we rewrite this definition taking the value processes $z = (z_1, \ldots, z_T)$ with $z_1 = Z_1 = 0$, $z_t = \sum_{\tau=1}^{T} Z_\tau$, i.e., $Z_t = z_t - z_{t-1}$ for $t > 2$. This reformulation leads to the representation (3.2) with $k_t = 3$ $(t = 1, \ldots, T)$, $Y_1 = \mathbb{R} \times \mathbb{R} \times \{0\}$, $Y_t = \mathbb{R} \times \mathbb{R}_+ \times \mathbb{R}_+$ $(t = 2, \ldots, T)$, $y_t = (A_t, M_t, K_t)$, $w_{t,0} = (0, -1, 1)$ $(t = 1, \ldots, T)$, $w_{t,\tau} = (1, -1, 0)$ $(\tau = 1, \ldots, t-2,\ t = 3, \ldots, T)$, $w_{t,t-1} = (1, 0, 0)$ $(t = 2, \ldots, T)$, $c_1 = (-b_1, 0, 0)$, $c_t = (-b_t, q_t, 0)$ $(t = 2, \ldots, T-1)$, $c_T = (0, q_T, -d)$.

To understand this reformulation note that $w_{1,0} = (0, -1, 1)$ implies $M_1 = -z_1 = 0$ and that for $t = 2, \ldots, T$ the recursion $K_t - M_t = K_{t-1} + Z_t - A_{t-1}$ with $K_t \geq 0$ and $M_t \geq 0$ must hold. This recursion can be transformed into a recursion of the type of definition of multiperiod polyhedrality

$$z_t = K_t + \sum_{\tau=1}^{t-1} A_\tau - \sum_{\tau=2}^{t} M_\tau \quad (t = 2, \ldots, T)$$

with $K_1 = 0$. Thus, this risk measure fits into the framework of *multiperiod polyhedral* risk measures.

Furthermore, it is *multiperiod coherent* if $b_1 = 1$. This can be shown by means of Corollary 3.10. Note that

$$c_1 + \sum_{\nu=1}^{T} \mathbb{E}[\lambda_\nu]\, w_{\nu,\nu-1} \in -Y_1^* \iff \sum_{\nu=2}^{T} \mathbb{E}[\lambda_\nu] = b_1 \text{ and } \lambda_1 = 0$$

and

$$\begin{array}{l} c_t + \sum_{\nu=t}^{T} \mathbb{E}[\lambda_\nu | \mathcal{F}_t]\, w_{\nu,\nu-t} \in -Y_t^* \ (t = 2, \ldots, T) \iff \\ d \leq \lambda_T \leq q_T,\ 0 \leq \lambda_t \leq q_t - b_t,\ \sum_{\nu=t+1}^{T} \mathbb{E}[\lambda_\nu | \mathcal{F}_t] = b_t \ (t = 2, \ldots, T-1), \end{array}$$

[3] The framework in these papers assumes that the multiperiod risk measure is determined only by a set of (scalar) density functions $\mathcal{P}_\rho \subseteq L_1(\Omega, \mathcal{F}, \mathbb{P})$ rather than $\mathcal{P}_\rho \subseteq \times_{t=1}^{T} L_1(\Omega, \mathcal{F}_t, \mathbb{P})$. Then, the risk $\rho(z)$ is given by expressions like $\sup\{-\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[f z_t] : f \in \mathcal{P}_\rho\}$ [25] or $\sup\{-\mathbb{E}[f z_\tau] : f \in \mathcal{P}_\rho,$ τ stopping time$\}$ [3]. Indeed, Λ_{ρ_4} is nothing else but the set of densities for the Conditional-Value-at-Risk (2.9) in terms of [25], i.e., all density functions bounded by $\frac{1}{\alpha}$.

[4] In [22], ρ_5 is called a (negative) utility measure rather than a risk measure. Moreover, the first time stage (i.e., the deterministic stage) is denoted by index 0 there. Here, the formulation is adapted to our framework with index 1 for the deterministic time stage (i.e., $\mathcal{F}_1 = \{\emptyset, \Omega\}$). In addition, the notations c_t and a_t were replaced by the definitions $b_t := c_{t+1}$ and $A_t := a_{t+1}$.

thus

$$\Lambda_{\rho_5} = \left\{ \lambda \in \times_{t=1}^{T} L_{p'}(\Omega, \mathcal{F}_t, P) \;\middle|\; \begin{array}{l} \lambda_1 = 0, \\ 0 \le \lambda_t \le q_t - b_t \text{ a.s. } (t = 2, \dots, T-1), \\ d \le \lambda_T \le q_T \text{ a.s.}, \\ \mathbb{E}\left[\lambda_t | \mathcal{F}_{t-1}\right] = b_{t-1} - b_t \; (t = 2, \dots, T) \end{array} \right\}.$$

Further, complete recourse is obviously satisfied and dual feasibility holds since the vector $u \in \mathbb{R}^T$ with $u_1 = 0$, $u_T = b_{T-1}$, and $u_t = b_{t-1} - b_t$ for $t = 2, \dots, T-1$ defines a (constant) element of Λ_{ρ_5}. Furthermore, $\sum_{t=1}^{T} \mathbb{E}[\lambda_t] = b_1$ for $\lambda \in \Lambda_{\rho_5}$, thus the inclusion $\Lambda_{\rho_5} \subseteq \mathcal{D}_T$ holds indeed if $b_1 = 1$.

An interesting specific case appears for $d = 0$, $b_t = \frac{T-t}{T-1}$, and $q_t = b_t + \frac{1}{(T-1)\alpha_t}$ $(t = 1, \dots, T)$ with $\alpha_t \in (0,1)$. Then we obtain

$$\Lambda_{\rho_5} = \left\{ \lambda \in \times_{t=1}^{T} L_{p'}(\Omega, \mathcal{F}_t, P) \;\middle|\; \begin{array}{l} \lambda_1 = 0, \quad 0 \le \lambda_t \le \frac{1}{(T-1)\alpha_t} \text{ a.s.,} \\ \mathbb{E}\left[\lambda_t | \mathcal{F}_{t-1}\right] = \frac{1}{T-1} \end{array} \quad (t = 2, \dots, T) \right\}$$

and the risk measure ρ_5 on $\times_{t=1}^{T} L_p(\Omega, \mathcal{F}_t, P)$ takes the form

$$\rho_5(z) = \tfrac{1}{T-1} \textstyle\sum_{t=2}^{T} \inf \left\{ \mathbb{E}\left[u_{t-1} + \tfrac{1}{\alpha_t}(z_t + u_{t-1})^- \right] \;\middle|\; u_t \in L_p(\Omega, \mathcal{F}_t, \mathbb{P}) \right\}. \tag{3.11}$$

The tth summand can be interpreted as the expectation of the Conditional-Value-at-Risk of z_t conditioned with respect to the σ-field $\mathcal{F}_{t-1}$. Clearly, (3.11) boils down to the one-period $CVaR$ (2.8) for $T = 2$.

Remark 3.16. Of course, it is interesting to compare these examples. To this end, it is useful to consider the dual representations, i.e., the Lagrange multiplier sets Λ_{ρ_j} $(j = 1, \dots, 5)$. Hence, regarding formulas (3.8), (3.9), and (3.10), it is obvious that for $\beta_t = \delta_t = \mu_t$ it holds that $\Lambda_{\rho_4} \subseteq \Lambda_{\rho_2} \supseteq \Lambda_{\rho_3}$, thus, since

$$\rho_j(z) = \sup \left\{ - \textstyle\sum_{t=1}^{T} \mathbb{E}\left[\lambda_t z_t\right] : \lambda \in \Lambda_{\rho_j} \right\}, \tag{3.12}$$

the relation $\rho_4 \le \rho_2 \ge \rho_3$ is valid. On the other hand, comparing ρ_3 and ρ_4 for the case $\delta_t = 2\mu_t$ leads to $\Lambda_{\rho_4} \subseteq \Lambda_{\rho_3}$, thus $\rho_4 \le \rho_3$. Hence, ρ_3 is more cautious than ρ_4 in this case. Moreover, if we set $\gamma_t = \frac{1}{T-1}$ and $\beta_t = \mu_t = \frac{1}{(T-1)\alpha_t}$, formula (3.7) shows $\Lambda_{\rho_4} \subseteq \Lambda_{\rho_1} \subseteq \Lambda_{\rho_2}$, hence $\rho_4 \le \rho_1 \le \rho_2$. Thus, ρ_2 is the most cautious or most pessimistic of these risk measures.

More precisely, for a fixed random variable z let $\lambda^j = \lambda^j(z) \in \Lambda_{\rho_j}$ be a maximizer for the dual representations (3.12) of ρ_j, respectively. Then, roughly speaking, λ^j is big where z is small in compliance with the respective restrictions. For $j = 1$ and $j = 4$, the weighting of the time steps is fixed in advance since $\mathbb{E}[\lambda_t^j]$ is fixed. For $j = 2$ the weighting of the time steps is variable, hence the available probability mass of λ^2 is concentrated at time steps at which z is low. Thus, ρ_2 is a kind of worst time step risk measure. This might be desirable or not, depending on the application.

Comparing ρ_1 with ρ_4, one sees that in the first case λ_t^1 is big where z_t is small, independent of the other time steps. In the second case, λ^4 is completely determined by λ_T^4 since $\lambda_t^4 = \mathbb{E}[\lambda_T^4 | \mathcal{F}_t]$ because of the martingale property. This means that the maximization (3.12) takes all time steps into account simultaneously, i.e., the maximization occurs along the paths of the treelike information structure given by the filtration $(\mathcal{F}_t)_{t=1}^{T}$. This latter approach seems to be more efficient in case the risk of paths is of interest. Then, ρ_1 may be more pessimistic than necessary. Furthermore, it does not incorporate the information structure of the problem. On the other hand, the martingale property of ρ_4 seems very restrictive.

Comparing ρ_3 and ρ_4 for the case $\delta_t = 2\mu_t$ leads to $\Lambda_{\rho_4} \subseteq \Lambda_{\rho_3}$, thus $\rho_4 \leq \rho_3$. Hence, ρ_3 is more cautious than ρ_4 in this case. Regarding the dual sets for ρ_5, one obtains $\Lambda_{\rho_5} \subseteq \Lambda_{\rho_1}$ for $\gamma_t = b_{t-1} - b_t$ and $\alpha_t = (b_{t-1} - b_t)/(q_t - b_t)$, and $\Lambda_{\rho_5} \subseteq \Lambda_{\rho_3}$ for $\delta_t = q_t - b_{t+1}$. Hence, $\rho_1 \geq \rho_5 \leq \rho_3$, i.e., ρ_5 is less cautious for this choice of the coefficients.

However, cautiousness is not necessarily a desirable property, because in applications one usually has to pay a price for being cautious. Which risk measure to take depends highly on the intention of the application. It seems that ρ_3 may be a good compromise, since the information structure is taken into account and there is no fixed weighting of the time steps. For initial numerical results we refer to [9].

4. Risk measures in stochastic programs. In this section we study the effect of replacing expectation-based objectives of stochastic programming problems by polyhedral risk measures. In particular, we are interested in consequences for structural and stability properties of the resulting models. We assume that randomness occurs as a (possibly multivariate) stochastic data process $(\xi_t)_{t=1}^T$ and set $\mathcal{F}_t = \sigma(\xi_1, \ldots, \xi_t)$, $t = 1, \ldots, T$. We consider multistage stochastic programs of the form

$$\min \left\{ \mathbb{E}\left[\sum_{t=1}^{T} \langle b_t(\xi_t), x_t \rangle \right] \;\middle|\; \begin{array}{ll} x_t \in X_t, & \\ H_t(x_t) = 0, & \\ B_t(\xi_t) x_t \leq d_t(\xi_t), & (t = 1, \ldots, T) \\ \sum_{\tau=0}^{t-1} A_{t,\tau}(\xi_t) x_{t-\tau} = h(\xi_t) & \end{array} \right\} \tag{4.1}$$

with closed sets X_t having the property that their convex hull is polyhedral, and with cost coefficients $b_t(\cdot)$, right-hand sides $d_t(\cdot)$ and $h_t(\cdot)$, and matrices $A_{t,\tau}(\cdot)$, $\tau = 0, \ldots, t-1$, and $B_t(\cdot)$ all having suitable dimensions and possibly depending affine linearly on ξ_t for $t = 1, \ldots, T$. The constraints consist of four groups, where the first $x_t \in X_t$ models simple fixed constraints, the second $H_t(z) := z - \mathbb{E}[z|\mathcal{F}_t] = 0$ ensures the nonanticipativity of the decisions x_t, and the third and fourth are the coupling and the dynamic constraints, respectively. By $\mathcal{X}(\xi)$ we denote the set of decisions satisfying all constraints of (4.1).

When replacing the expectation of the stochastic overall costs $\sum_{t=1}^{T} \langle b_t(\xi_t), x_t \rangle$ by some polyhedral multiperiod risk measure ρ applied to the random vector

$$z(x, \xi) := \Big(-\langle b_1(\xi_1), x_1 \rangle, -\langle b_1(\xi_1), x_1 \rangle - \langle b_2(\xi_2), x_2 \rangle, \ldots, -\textstyle\sum_{\tau=1}^{T} \langle b_\tau(\xi_\tau), x_\tau \rangle \Big)$$

of negative intermediate costs, we arrive at the following risk averse alternative to problem (4.1):

$$\min \left\{ \rho\left(z(x, \xi) \right) \mid x \in \mathcal{X}(\xi) \right\}. \tag{4.2}$$

The polyhedral risk measure ρ is defined by the minimization problem

$$\rho(z) = \inf \left\{ \mathbb{E}\left[\textstyle\sum_{t=1}^{T} \langle c_t, y_t \rangle \right] \;\middle|\; \begin{array}{ll} H_t(y_t) = 0, \ y_t \in Y_t, & (t = 1, \ldots, T) \\ \sum_{\tau=0}^{t-1} \langle w_{t,\tau}, y_{t-\tau} \rangle = z_t & \end{array} \right\}.$$

This gives rise to the question whether (4.2) is equivalent to the optimization model

$$(4.3)\quad \min\left\{\mathbb{E}\left[\sum_{t=1}^{T}\langle c_t, y_t\rangle\right] \;\middle|\; \begin{array}{l} x\in\mathcal{X}(\xi), \\ H_t(y_t)=0,\ y_t\in Y_t\ (t=1,\ldots,T), \\ \sum_{\tau=0}^{t-1}\langle w_{t,\tau}, y_{t-\tau}\rangle + \sum_{\tau=1}^{t}\langle b_\tau(\xi_\tau), x_\tau\rangle = 0\ (t=1,\ldots,T) \end{array}\right\},$$

where the minimization with respect to the original decision x and the variable y defining ρ is carried out simultaneously. Of course, the answer is positive.

Proposition 4.1. *Minimizing* (4.2) *with respect to x is equivalent to minimizing* (4.3) *with respect to all pairs (x, y) in the following sense: The optimal values of* (4.2) *and* (4.3) *coincide and a pair (x^*, y^*) is a solution of* (4.3) *if and only if x^* solves* (4.2) *and y^* is a solution of the minimization problem defining $\rho(z(x^*, \xi))$.*

Proof. The minimization with respect to all feasible pairs (x, y) of (4.3) can be carried out by minimizing with respect to y and then by minimizing the latter residual with respect to $x \in \mathcal{X}(\xi)$. Hence, the optimal values coincide and, if the pair (x^*, y^*) solves (4.3), its first component x^* is a solution of (4.2) and y^* is a solution of the problem

$$(4.4)\quad \min\left\{\mathbb{E}\left[\sum_{t=1}^{T}\langle c_t, y_t\rangle\right] \;\middle|\; \begin{array}{l} H_t(y_t)=0,\ y_t\in Y_t, \\ \sum_{\tau=0}^{t-1}\langle w_{t,\tau}, y_{t-\tau}\rangle + \sum_{\tau=1}^{t}\langle b_\tau(\xi_\tau), x^*_\tau\rangle = 0 \end{array}\right\},$$

whose optimal value is just $\rho(z(x^*, \xi))$. Conversely, if x^* is a solution of (4.2) and y^* a solution of (4.4), the pair (x^*, y^*) has to be a solution of (4.3). □

Thus, minimizing a stochastic program with a polyhedral risk measure in the objective leads to a "traditional" stochastic program with linear expectation-based objective and with additional variables y and constraints, respectively. Both the variables and the constraints are convenient for stochastic programs since the variables are nicely constrained by polyhedral sets (no integer requirements). Thus, if the original expectation-based stochastic program (4.1) has convenient properties, there is good reason to expect that these properties are maintained when using a polyhedral risk measure for risk aversion.

4.1. Stability of stochastic programs. Stability of solutions and optimal values of stochastic programs with respect to the perturbation of the underlying probability measure is an important issue since in applications the true measure $\mathbb{P}$ is usually unknown and has to be approximated by some other measure $\mathbb{Q}$. Such an approximation may be gained by sampling techniques.

In [30] various stability results involving distances $d(\mathbb{P}, \mathbb{Q})$ of probability measures are developed for different types of (mainly) expectation-based stochastic programs. It is shown there that certain ideal probability metrics (see [23] for an exposition) may be associated with classes of stochastic programs. Here, we briefly show that these stability results remain valid for important classes if the expectation is replaced by a polyhedral risk measure. We restrict ourselves to the two-stage case here since stability properties are best understood for such programs. In the context of distances of probability measures it turns out to be useful to assume that $\Omega = \Xi \subseteq \mathbb{R}^n$ and $\mathcal{F} = \mathcal{B}(\Xi)$.

4.1.1. Linear two-stage programs. In [24, Theorem 3.3] and [30] it is shown that two-stage stochastic programs with fixed recourse of the form

$$(4.5)\quad \min\left\{\langle b, x_1\rangle + \mathbb{E}_{\mathbb{P}}\left[\langle p(\cdot), x_2(\cdot)\rangle\right] \;\middle|\; \begin{array}{l} Wx_2(\xi) = h(\xi) - T(\xi)x_1, \\ x_1\in X_1,\ x_2(\xi)\in X_2 \end{array}\right\},$$

with X_1 and Ξ being polyhedral sets, X_2 being a polyhedral cone, and $p(\cdot)$, $h(\cdot)$, $T(\cdot)$ being affine linear functions (of $\xi \in \Xi$), are stable[5] at $\mathbb{P}$ with respect to the probability metric ζ_2 given by

$$\zeta_2(\mathbb{P},\mathbb{Q}) = \sup \left\{ |\mathbb{E}_{\mathbb{P}}[F] - \mathbb{E}_{\mathbb{Q}}[F]| \;\middle|\; \begin{array}{l} F : \Xi \to \mathbb{R}, \\ |F(\xi) - F(\xi')| \le \max\{1, \|\xi\|, \|\xi'\|\} \cdot \|\xi - \xi'\| \\ \forall\, \xi, \xi' \in \Xi \end{array} \right\}$$

if the following four conditions hold:

(A1) $\forall\,(x_1,\xi) \in X_1 \times \Xi \;\exists\, x_2 \in X_2 : Wx_2 = h(\xi) - T(\xi)x_1$
(relatively complete recourse).

(A2) $\forall\,\xi \in \Xi\;\exists\, z : W'z - p(\xi) \in X_2^*$ (dual feasibility).

(A3) $\mathbb{E}_{\mathbb{P}}\|\xi\|^2 < \infty$ (finite second moments).

(A4) The first stage solution set $S_{\mathbb{E}} \subseteq X_1$ is nonempty and bounded.

The program (4.5) is equivalent to $\min\{\mathbb{E}_{\mathbb{P}}[z(x_1)] : x_1 \in X_1\}$ using the notations $z(x_1) := \langle b, x_1\rangle + \Phi(p(\cdot), h(\cdot) - T(\cdot)x_1)$ and the second stage value function $\Phi(u,t) := \inf\{\langle u, x_2\rangle : x_2 \in X_2,\, Wx_2 = t\}$ (cf. [34, 29, 30]). Hence, the first stage solution set is given by $S_{\mathbb{E}} := \{x_1 \in X_1 : \mathbb{E}[z(x_1)] = v_{\mathbb{E}}\}$ with $v_{\mathbb{E}} := \inf\{\mathbb{E}[z(x_1)] : x_1 \in X_1\}$ denoting the optimal value.

If we exchange from expectation to a (one-period) polyhedral risk measure $\rho = \rho_{\mathbb{P}}$ according to Definition 2.2, we obtain the problem

$$\min \left\{ \rho\left[-\langle b, x_1\rangle - \langle p(.), x_2(.)\rangle\right] \;\middle|\; \begin{array}{l} Wx_2(\xi) = h(\xi) - T(\xi)x_1, \\ x_1 \in X_1,\; x_2(\xi) \in X_2 \end{array} \right\}, \tag{4.6}$$

which is equivalent to $\min\{\rho[-z(x_1)] : x_1 \in X_1\}$ and, too, equivalent to

$$\min \left\{ \begin{array}{l} \langle c_1, y_1\rangle + \\ \mathbb{E}\left[\langle c_2, y_2(.)\rangle\right] \end{array} \;\middle|\; \begin{array}{l} x_1 \in X_1,\; x_2(\xi) \in X_2, \\ y_1 \in Y_1,\; y_2(\xi) \in Y_2, \\ Wx_2(\xi) = h(\xi) - T(\xi)x_1, \\ \langle p(\xi), x_2(\xi)\rangle + \langle w_2, y_2(\xi)\rangle = -\langle b, x_1\rangle - \langle w_1, y_1\rangle \end{array} \right\}. \tag{4.7}$$

The latter program has almost the same structure as (4.5) with
$\hat{x}_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$, $\hat{x}_2 = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$, $\hat{h}(\xi) = \begin{pmatrix} h(\xi) \\ 0 \end{pmatrix}$, $\hat{b} = \begin{pmatrix} 0 \\ c_1 \end{pmatrix}$, $\hat{p} = \begin{pmatrix} 0 \\ c_2 \end{pmatrix}$,
$\hat{W}(\xi) = \begin{pmatrix} W & 0 \\ p(\xi)' & w_2' \end{pmatrix}$, $\hat{T}(\xi) = \begin{pmatrix} T(\xi) & 0 \\ b & w_1' \end{pmatrix}$, $\hat{X}_1 = X_1 \times Y_1$, $\hat{X}_2 = X_2 \times Y_2$,
but now the recourse matrix $\hat{W}$ is random while the cost coefficient $\hat{p}$ is nonrandom.

Moreover, if we also impose complete recourse and dual feasibility for the polyhedral risk measure ρ in the sense of section 2, i.e., (i) $\langle w_2, Y_2\rangle = \mathbb{R}$ and (ii) $D_{\rho,1} \cap D_{\rho,2} \neq \emptyset$, $D_{\rho,2} \subseteq \mathbb{R}_+$, then we can conclude both relatively complete recourse and dual feasibility for the risk aversive alternative (4.7):

(A1) Relatively complete recourse:
Let $(x_1, y_1, \xi) \in X_1 \times Y_1 \times \Xi$; then $\exists\, x_2 \in X_2 : Wx_2 = h(\xi) - T(\xi)x_1$ and $y_2 \in Y_2$ can be chosen such that $\langle w_2, y_2\rangle + \langle p(\xi), x_2\rangle = -\langle b, x_1\rangle - \langle w_1, y_1\rangle$ because $\langle w_2, Y_2\rangle = \mathbb{R}$, thus $\hat{W}(\xi)\hat{x}_2 = \hat{h}(\xi) - \hat{T}(\xi)\hat{x}_1$.

[5] We do not give a precise definition of stability here; see [30] for this. Briefly, stability means that optimal values and (first stage) solution sets behave (quantitatively) continuous at the original measure $\mathbb{P}$ with respect to a distance $d(\mathbb{P},\mathbb{Q})$.

(A2) Dual feasibility:
Let $\xi \in \Xi$. Choose $v \in D_{\rho,2} = \{u \in \mathbb{R} : -(c_2 + uw_2) \in Y_2^*\} \subseteq \mathbb{R}_+$ and z such that $W'z + p(\xi) \in X_2^*$, set $\hat{z} = (vz', -v)'$; then one obtains

$$\hat{W}(\xi)'\hat{z} - \hat{p} = \begin{pmatrix} v(W'z - p(\xi)) \\ -vw_2 - c_2 \end{pmatrix} \in X_2^* \times Y_2^* = \hat{X}_2^*,$$

by making use of the fact that X_2 is a cone.

Since the randomness enters only the last row of $\hat{W}(\xi)$ except for the coefficient in the main diagonal, the stability results from [32] for the random recourse situation with only lower diagonal randomness apply. The model (4.7) with nonrandom costs, however, is again stable with respect to the same metric ζ_2 as for (4.5) if the (first stage) solution set $\bar{S} \subseteq X_1 \times Y_1$ of (4.7) is nonempty and bounded.

PROPOSITION 4.2. *Let ρ be a polyhedral risk measure on $L_1(\Omega, \mathcal{F}, \mathbb{P})$ of the form* (2.1). *Assume that the conditions* (i) $\langle w_2, Y_2 \rangle = \mathbb{R}$ *and* (ii) $D_{\rho,1} \cap D_{\rho,2} \neq \emptyset$, $D_{\rho,2} \subseteq \mathbb{R}_+$, *are satisfied and that the set $S(\rho(0))$ (see* (2.6)*) is nonempty and bounded. Then the set $\bar{S} \subseteq X_1 \times Y_1$ is nonempty and bounded if the solution set $S_\rho := \{x_1 \in X_1 : \rho[-z(x_1)] = \inf_{\hat{x}_1 \in X_1} \rho[-z(\hat{x}_1)]\}$ of* (4.6) *is nonempty and bounded. Hence, the stochastic program* (4.7) *is stable at $\mathbb{P}$ with respect to the metric ζ_2 if the conditions* (A1)–(A3) *are valid and S_ρ is nonempty and bounded.*

Proof. Proposition 4.1 implies that the set $\bar{S}$ is nonempty and bounded if S_ρ is nonempty and bounded and the subset $\bigcup_{x_1 \in S_\rho} S(\rho[-z(x_1)])$ of Y_1 is bounded. Here, $S(\rho(z))$ is defined by (2.6) and is nonempty and bounded due to Proposition 2.9. Clearly, nothing has to be shown if Y_1 is bounded. Now, let Y_1 be unbounded. Suppose $\bigcup_{x_1 \in S_\rho} S(\rho[-z(x_1)])$ is unbounded. Then there exist sequences $(y_{1,n})$ and $(x_{1,n})$ such that $x_{1,n} \in S_\rho$, $y_{1,n} \in S(\rho[-z(x_{1,n})])$ and $\|y_{1,n}\| \geq n$ for $n \in \mathbb{N}$. Because S_ρ is compact, we may assume without loss of generality that $x_{1,n} \to x_{1,0} \in S_\rho$. Since Φ is Lipschitz in t (cf. [43]) we have $z(x_{1,n}) \to z(x_{1,0})$ in $L_1(\Xi)$. Hence, the sequence of probability distributions of $z(x_{1,n})$ converges to the distribution of $z(x_{1,0})$ with respect to the Fortet–Mourier metric ζ_1 (cf. [23, section 5.1]). Now, the set $S(\rho[-z(x_{1,0})])$ is nonempty and bounded. Therefore, the stability result [30, Corollary 25] for two-stage stochastic programs with random right-hand side implies that there must exist an index $n_0 \in \mathbb{N}$ such that for $n \geq n_0$ the sets $S(\rho[-z(x_{1,n})])$ are contained in a fixed bounded neighborhood of $S(\rho[-z(x_{1,0})])$. This contradicts $\|y_{1,n}\| \geq n$, thus $\bigcup_{x_1 \in S_\rho} S(\rho[-z(x_1)])$ must be bounded. □

4.1.2. Linear mixed-integer two-stage programs. In [30, Theorem 35], it is shown that programs of the form

(4.8)
$$\min \left\{ \mathbb{E}_\mathbb{P}\left[\langle b, x_1 \rangle + \langle p, x_2(.) \rangle + \langle \bar{p}, \bar{x}_2(.) \rangle\right] \;\middle|\; \begin{array}{l} x_1 \in X_1, \\ x_2(\xi) \in X_2 \cap \mathbb{Z}^m, \ \bar{x}_2(\xi) \in \bar{X}_2, \\ Wx_2(\xi) + \bar{W}\bar{x}_2(\xi) = h(\xi) - T(\xi)x_1 \end{array} \right\}$$

with a closed Euclidean set X_1, a polyhedral set Ξ, and polyhedral cones X_2 and $\bar{X}_2$ are stable with respect to the probability metric ζ_{1,ph_k} with some $k \in \mathbb{N}$ if the following conditions are satisfied:

(B1) $\forall\, (x_1, \xi) \in X_1 \times \Xi\ \exists\, x_2 \in X_2 \cap \mathbb{Z}^m,\ \bar{x}_2 \in \bar{X}_2 : Wx_2 + \bar{W}\bar{x}_2 = h(\xi) - T(\xi)x_1$ (relatively complete recourse).

(B2) $\exists\, z \in \mathbb{R}^r : W'z + p \in X_2^*$ and $\bar{W}'z + \bar{p} \in \bar{X}_2^*$ (dual feasibility).

(B3) $\mathbb{E}_{\mathbb{P}}\|\xi\| < \infty$ (finite first moments).

(B4) W and $\bar{W}$ have rational coefficients only (rational recourse).

(B5) The first stage solution set $S_{\mathbb{E}} \subseteq X_1$ is nonempty and bounded.

The metric ζ_{1,ph_k} is given by

$$\zeta_{1,ph_k}(\mathbb{P},\mathbb{Q}) = \sup\left\{ |\mathbb{E}_{\mathbb{P}}[\chi_B \cdot F] - \mathbb{E}_{\mathbb{Q}}[\chi_B \cdot F]| \;\middle|\; \begin{array}{l} B \in \mathcal{B}_{ph_k}(\Xi),\ F : \Xi \to \mathbb{R} \\ |F(\xi) - F(\xi')| \le \|\xi - \xi'\| \\ \forall\, \xi, \xi' \in \Xi \end{array} \right\},$$

where $\mathcal{B}_{ph_k}(\Xi)$ is the set of polyhedra contained in Ξ with at most k faces and χ denotes the characteristic function, i.e., $\chi_B(\xi) = 1$ if $\xi \in B$ and $= 0$ otherwise.

If we exchange in (4.8) from expectation to a polyhedral risk measure ρ we obtain the problem $\min\{\rho[-z(x_1)] : x_1 \in X_1\}$ with $z(x_1) := \langle b, x_1\rangle + \Phi(h(\cdot) - T(\cdot)x_1)$ and $\Phi(t) := \inf\{\langle p, x_2\rangle + \langle \bar{p}, \bar{x}_2\rangle : x_2 \in X_2 \cap \mathbb{Z}^m,\ \bar{x}_2 \in \bar{X}_2,\ Wx_2 + \bar{W}\bar{x}_2 = t\}$. This problem is equivalent to

$$\min\left\{ \begin{array}{l} \langle c_1, y_1\rangle + \\ \mathbb{E}\left[\langle c_2, y_2(.)\rangle\right] \end{array} \;\middle|\; \begin{array}{l} x_1 \in X_1,\ x_2(\xi) \in X_2 \cap \mathbb{Z}^m,\ \bar{x}_2(\xi) \in \bar{X}_2, \\ y_1 \in Y_1,\ y_2(\xi) \in Y_2, \\ Wx_2(\xi) + \bar{W}\bar{x}_2(\xi) = h(\xi) - T(\xi)x_1, \\ \langle w_2, y_2(\xi)\rangle + \langle p, x_2(\xi)\rangle + \langle \bar{p}, \bar{x}_2(\xi)\rangle = \\ -\langle b, x_1\rangle - \langle w_1, y_1\rangle \end{array} \right\}. \tag{4.9}$$

The latter program has the same structure as (4.8) with
$\hat{x}_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$, $\hat{x}_2 = x_2$, $\hat{\bar{x}}_2 = \begin{pmatrix} \bar{x}_2 \\ y_2 \end{pmatrix}$, $\hat{X}_1 = X_1 \times Y_1$, $\hat{X}_2 = X_2$, $\hat{\bar{X}}_2 = \bar{X}_2 \times Y_2$,
$\hat{W} = \begin{pmatrix} W & 0 \\ p' & w_2' \end{pmatrix}$, $\hat{\bar{W}} = \begin{pmatrix} \bar{W} \\ \bar{p}' \end{pmatrix}$, $\hat{T}(\xi) = \begin{pmatrix} T(\xi) & 0 \\ b' & w_1' \end{pmatrix}$, $\hat{h}(\xi) = \begin{pmatrix} h(\xi) \\ 0 \end{pmatrix}$,
$\hat{b} = \begin{pmatrix} 0 \\ c_1 \end{pmatrix}$, $\hat{p} = \begin{pmatrix} 0 \\ c_2 \end{pmatrix}$, $\hat{\bar{p}} = 0$.

As in the previous paragraph, this combined program satisfies relatively complete recourse and dual feasibility if both (4.8) and ρ do so. To have the condition (B4) satisfied, one has to impose additionally that also p, $\bar{p}$, and w_2 have only rational coefficients. Then, however, the same stability (with respect to the metric ζ_{1,ph_k}) as for the original program is guaranteed if the (first stage) solution set $\bar{S} \subseteq X_1 \times Y_1$ of (4.9) is nonempty and bounded. Unfortunately, we cannot conclude as in Proposition 4.2 in the mixed-integer case since Φ is no longer continuous. However, a quantitative stability result is available for the expected loss and the Conditional-Value-at-Risk.

PROPOSITION 4.3. *Let ρ denote the expected loss or the Conditional-Value-at-Risk (see section* 2*). Then the first stage solution set $\bar{S} \subseteq X_1 \times Y_1$ of* (4.9) *is nonempty and bounded if the set $S_\rho := \{x_1 \in X_1 : \rho[-z(x_1)] = \inf_{\hat{x}_1 \in X_1} \rho[-z(\hat{x}_1)]\}$ is nonempty and bounded. Hence, the stochastic program* (4.9) *is stable at $\mathbb{P}$ with respect to ζ_{1,ph_k} if the conditions* (B1)–(B3), (B4)′ *W, $\bar{W}$, p, and $\bar{p}$ have rational coefficients only and are satisfied and S_ρ is nonempty and bounded.*

Proof. As in Proposition 4.2, boundedness of $\bar{S}$ is guaranteed if both the set S_ρ of X_1-solutions of (4.9) is nonempty and bounded and the subset $\bigcup_{x_1 \in S_\rho} S(\rho[-z(x_1)])$ of Y_1 is bounded, too. Clearly, the latter set is bounded if Y_1 is bounded which is the case for the expected loss. For the Conditional-Value-at-Risk, we argue as follows. The set of random variables $\{z(x_1) : x_1 \in S_\rho\}$ is bounded in $L_1(\Xi)$ since S_ρ is bounded and the estimate $|\Phi(t) - \Phi(\tilde{t})| \le a\|t - \tilde{t}\| + b$ holds for the second stage function Φ with some positive coefficients a and b (e.g., [30, Lemma 33]). This implies boundedness of the set of their probability distributions $\{D(z(x_1)) : x_1 \in S_\rho\}$ with respect to the Fortet–Mourier metric ζ_1. For real random variables z, $\hat{z}$ and their distributions $D(z)$, $D(\hat{z})$ the metric ζ_1 has the explicit representation (cf. [23, section 5.4])

$$\zeta_1(D(z), D(\hat{z})) = \int_{-\infty}^{\infty} |\mathbb{P}(z \le t) - \mathbb{P}(\hat{z} \le t)| dt.$$

For the $CVaR_\alpha$ we know that for any random variable z the first stage solution set is given by the interval of negative quantiles $S(\rho(z)) = [-\bar{q}_\alpha(z), -q_\alpha(z)]$ (cf. Example 2.10). Fix $\hat{x}_1 \in S_\rho$ and set $\hat{z} := z(\hat{x}_1)$. Let $\Psi_j : \mathbb{R}_+ \to \mathbb{R}_+$ $(j = 1, 2)$ be defined by

$$\Psi_1(r) := \int_{q_\alpha(\hat{z})-r}^{q_\alpha(\hat{z})} (\alpha - \mathbb{P}(\hat{z} \le t))\, dt \qquad \Psi_2(r) := \int_{\bar{q}_\alpha(\hat{z})}^{\bar{q}_\alpha(\hat{z})+r} (\mathbb{P}(\hat{z} \le t) - \alpha)\, dt.$$

Note that the functions Ψ_j $(j = 1, 2)$ are *strictly* increasing. Let z be a random variable. We show that the distances $|q_\alpha(\hat{z}) - q_\alpha(z)|$ and $|\bar{q}_\alpha(\hat{z}) - \bar{q}_\alpha(z)|$ are bounded in terms of $\zeta_1(D(z), D(\hat{z}))$. In case $q_\alpha(z) < q_\alpha(\hat{z})$ it holds that

$$\begin{aligned} \zeta_1(D(z), D(\hat{z})) &= \int_{-\infty}^{\infty} |\mathbb{P}(z \le t) - \mathbb{P}(\hat{z} \le t)| dt \quad \ge \int_{q_\alpha(z)}^{q_\alpha(\hat{z})} |\mathbb{P}(z \le t) - \mathbb{P}(\hat{z} \le t)| dt \\ &= \int_{q_\alpha(z)}^{q_\alpha(\hat{z})} (\mathbb{P}(z \le t) - \mathbb{P}(\hat{z} \le t))\, dt \ \ge \int_{q_\alpha(z)}^{q_\alpha(\hat{z})} (\alpha - \mathbb{P}(\hat{z} \le t))\, dt \\ &= \Psi_1(q_\alpha(\hat{z}) - q_\alpha(z)), \end{aligned}$$

hence $|q_\alpha(\hat{z}) - q_\alpha(z)| \le \Psi_1^{-1}(\zeta_1(D(z), D(\hat{z})))$. In case $\bar{q}_\alpha(z) > \bar{q}_\alpha(\hat{z})$ we get

$$\begin{aligned} \zeta_1(D(z), D(\hat{z})) &\ge \int_{\bar{q}_\alpha(\hat{z})}^{\bar{q}_\alpha(z)} |\mathbb{P}(z \le t) - \mathbb{P}(\hat{z} \le t)| dt \ = \int_{\bar{q}_\alpha(\hat{z})}^{\bar{q}_\alpha(z)} (\mathbb{P}(\hat{z} \le t) - \mathbb{P}(z \le t))\, dt \\ &\ge \int_{\bar{q}_\alpha(\hat{z})}^{\bar{q}_\alpha(z)} (\mathbb{P}(\hat{z} \le t) - \alpha)\, dt \qquad = \Psi_2(\bar{q}_\alpha(z) - \bar{q}_\alpha(\hat{z})), \end{aligned}$$

hence $|\bar{q}_\alpha(z) - \bar{q}_\alpha(\hat{z})| \le \Psi_2^{-1}(\zeta_1(D(z), D(\hat{z})))$. □

After this paper was submitted, the authors attention was called to the recent paper [39]. It contains a stability result for the Conditional-Value-at-Risk in mixed-integer two-stage stochastic programs, which is similar to the preceding proposition but proved without relying on Proposition 4.1.

4.2. Lagrangian relaxation and decomposition. We consider again the multistage stochastic program (4.1) and its risk averse alternative (4.2), which, according to Proposition 4.1, is of the form

$$\text{(4.10)} \qquad \min \left\{ \mathbb{E}\left[\sum_{t=1}^{T} \langle c_t, y_t \rangle\right] \;\middle|\; \begin{array}{l} x_t \in X_t,\ y_t \in Y_t, \\ H_t(x_t) = 0,\ H_t(y_t) = 0, \\ B_t(\xi_t) x_t \le d_t(\xi_t), \\ \sum_{\tau=0}^{t-1} A_{t,\tau}(\xi_t) x_{t-\tau} = h(\xi_t), \\ \sum_{\tau=0}^{t-1} (\langle w_{t,\tau}, y_{t-\tau} \rangle + \langle b_{\tau+1}(\xi_{\tau+1}), x_{\tau+1} \rangle) = 0 \end{array} \right\}.$$

Obviously, (4.10) has a similar structure as (4.1) but additionally with T vector valued random variables and T dynamic (equality) constraints. Thus, decomposition methods that work for (4.1) are likely to work similarly for (4.10), too. We exemplify this here by two important dual decomposition methods.

4.2.1. Scenario decomposition. When solving problems like (4.1) or (4.10) one usually has to approximate $\mathbb{P}$ or, equivalently, ξ by a finite number of scenarios (more precisely: by a finite scenario tree). This can be expressed by $\infty > \#\Omega =: S$ and one can assume without loss of generality $\Omega = \{\xi^1, \dots, \xi^S\}$ and $\mathcal{F} = \wp(\Omega)$. Then the problem is no longer infinite-dimensional and can be solved by standard mixed-integer linear programming techniques, but it is very large scale in most cases. Thus, specialized decomposition techniques are of great interest (cf. [8, 33, 31, 37, 34]).

Scenario decomposition means Lagrange-dualizing the nonanticipativity constraints of (4.10) and solving the dual scenario-wise. Setting $m_t := \dim x_t$ we obtain the dual problem

$$\max \left\{ D(\lambda_1, \lambda_2) : \lambda_{1t} \in L_1(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{R}^{m_t}),\ \lambda_{2t} \in L_1(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{R}^{k_t}) \right\},$$

where the dual function $D(\lambda_1, \lambda_2)$ is given by

$$D(\lambda_1, \lambda_2) = \min \left\{ L(\lambda_1, \lambda_2, x, y) \;\middle|\; \begin{array}{l} x_t \in X_t,\ y_t \in Y_t, \\ B_t(\xi_t) x_t \le d_t(\xi_t), \\ \sum_{\tau=0}^{t-1} A_{t,\tau}(\xi_t) x_{t-\tau} = h(\xi_t), \\ \sum_{\tau=0}^{t-1} (\langle w_{t,\tau}, y_{t-\tau} \rangle + \langle b_{\tau+1}(\xi_{\tau+1}), x_{\tau+1} \rangle) = 0 \end{array} \right\}$$

with $L(\lambda_1, \lambda_2, x, y) := \mathbb{E}[\sum_{t=1}^T (\langle c_t, y_t \rangle + \langle \lambda_{1t}, H_t(x_t) \rangle + \langle \lambda_{2t}, H_t(y_t) \rangle)]$ denoting the Lagrangian. Solving this problem is an iterative process: $D(\lambda_1, \lambda_2)$ has to be computed for a fixed pair (λ_1, λ_2) and then (λ_1, λ_2) has to be updated via subgradient-type methods and so on. If the sets X_t are nonconvex, this procedure only leads to lower bounds of the optimal value of (4.1) and suitable globalization techniques based on these lower bounds have to be used in addition.

Because both the restrictions and the Lagrangian are separable with respect to scenarios for a fixed pair (λ_1, λ_2), the calculation of the dual function can be carried out scenario-wise, i.e., $D(\lambda_1, \lambda_2) = \sum_{s=1}^S \mathbb{P}(\{\xi^s\}) D^s(\lambda_1, \lambda_2)$. To derive the separability of the Lagrangian the identities $\mathbb{E}[\langle \lambda_{1t}, H_t(x_t) \rangle] = \mathbb{E}[\langle H_t(\lambda_{1t}), x_t \rangle]$ and $\mathbb{E}[\langle \lambda_{2t}, H_t(y_t) \rangle] = \mathbb{E}[\langle H_t(\lambda_{2t}), y_t \rangle]$ were used.

Hence, instead of one problem with $S \cdot \sum_{t=1}^T (m_t + k_t)$ variables one only has to solve S subproblems each with $\sum_{t=1}^T (m_t + k_t)$ variables to update the multipliers. In comparison with the (dualized form of the) purely expectation-based problem (4.1) one has T additional equality constraints and $\sum_{t=1}^T k_t$ additional variables in each subproblem. Note that the dimensions k_t of y_t are typically small compared to the dimensions m_t of x_t.

4.2.2. Geographical decomposition. In many practical applications the stochastic program (4.1) shows the following kind of block separability $x_i = (x_{i1}, \dots, x_{iT})$, $i = 1, \dots, I$, of components of x:

$$\min \left\{ \mathbb{E}\left[\sum_{i=1}^I \sum_{t=1}^T \langle b_{it}(\xi_t), x_{it} \rangle \right] \;\middle|\; \begin{array}{l} x_{it} \in X_{it}, \\ H_t(x_{it}) = 0, \\ \sum_{i=1}^I B_{it}(\xi_t) x_{it} \le d_t(\xi_t), \\ \sum_{\tau=0}^{t-1} A_{it,\tau}(\xi_t) x_{i,t-\tau} = h_{it}(\xi_t) \end{array} \right\}. \tag{4.11}$$

Hence, the I blocks of x are only coupled by the sum in the third constraint in (4.11). For such programs, *Lagrange relaxation of coupling constraints*, also known as *geographical* or *component decomposition*, may lead to efficient algorithms for computing lower bounds (cf. [8, 31]).

By exchanging from $\mathbb{E}$ to a multiperiod polyhedral risk measure this property is maintained, but an additional block consisting of the y_t variables and T *additional (dynamic) coupling constraints* appear,

$$(4.12)\quad \min\left\{ \mathbb{E}\left[\sum_{t=1}^{T}\langle c_t, y_t\rangle\right] \;\middle|\; \begin{array}{l} x_{it}\in X_{it},\ y_t\in Y_t,\\ H_t(x_{it})=0,\ H_t(y_t)=0,\\ \sum_{i=1}^{I} B_{it}(\xi_t)x_{it}\le d_t(\xi_t),\\ \sum_{\tau=0}^{t-1} A_{it,\tau}(\xi_t)x_{i,t-\tau}=h_{it}(\xi_t),\\ \sum_{\tau=0}^{t-1}(\langle w_{t,\tau}, y_{t-\tau}\rangle+\sum_{i=1}^{I}\langle b_{i,\tau+1}(\xi_t), x_{i,\tau+1}\rangle)=0 \end{array}\right\}.$$

Here, Lagrange relaxation of coupling constraints means to assign $\mathcal{F}_t$-measurable Lagrange multipliers λ_{1t} and λ_{2t} to the third *and* fifth constraint in (4.12), respectively, and to arrive at the dual problem

$$\max\left\{D(\lambda_1,\lambda_2) : \lambda_{1t}\in L_{p'}(\Omega,\mathcal{F}_t,\mathbb{P};\mathbb{R}_+^{n_t}),\ \lambda_{2t}\in L_{p'}(\Omega,\mathcal{F}_t,\mathbb{P})\right\}.$$

The dual function $D(\lambda_1,\lambda_2)$ is given by

$$D(\lambda_1,\lambda_2)=\min\left\{ L(\lambda_1,\lambda_2,x_1,\dots,x_I,y) \;\middle|\; \begin{array}{l} x_{it}\in X_{it},\ y_t\in Y_t,\\ H_t(x_{it})=0,\ H_t(y_t)=0,\\ \sum_{\tau=0}^{t-1} A_{it,\tau}(\xi_t)x_{i,t-\tau}=h_{it}(\xi_t) \end{array}\right\}$$

and the Lagrangian $L(\lambda_1,\lambda_2,x_1,\dots,x_I,y)$ is defined by

$$\begin{aligned} &L(\lambda_1,\lambda_2,x_1,\dots,x_I,y)\\ =&\mathbb{E}\Big[\textstyle\sum_{t=1}^{T}\Big(\langle c_t,y_t\rangle+\Big\langle\lambda_{1t},\sum_{i=1}^{I}B_{it}(\xi_t)x_{it}-d_t(\xi_t)\Big\rangle\\ &\qquad+\lambda_{2t}\textstyle\sum_{\tau=0}^{t-1}\Big(\langle w_{t,\tau},y_{t-\tau}\rangle+\sum_{i=1}^{I}\langle b_{i,\tau+1}(\xi_{\tau+1}),x_{i,\tau+1}\rangle\Big)\Big)\Big]. \end{aligned}$$

By rearranging with respect to blocks in the objective, the dual function D decomposes into $I+1$ minimization subproblems and is then of the form

$$D(\lambda_1,\lambda_2)=\sum_{i=1}^{I}D_i(\lambda_1,\lambda_2)+D_R(\lambda_2)-\mathbb{E}\left[\sum_{t=1}^{T}\langle\lambda_{1t},d_t(\xi_t)\rangle\right].$$

The functions D_i correspond to I geographical subproblems

$$\begin{aligned} &D_i(\lambda_1,\lambda_2)\\ =&\min\left\{\mathbb{E}\left[\sum_{t=1}^{T}\Big\langle B_{it}(\xi_t)'\lambda_{1t}+b_{it}(\xi_t)\textstyle\sum_{\tau=t}^{T}\lambda_{2\tau},x_{it}\Big\rangle\right] \;\middle|\; \begin{array}{l} x_{it}\in X_{it},\\ H_t(x_{it})=0,\\ \sum_{\tau=0}^{t-1}A_{it,\tau}(\xi_t)x_{i,t-\tau}=h_{it}(\xi_t) \end{array}\right\} \end{aligned}$$

and D_R corresponds to the risk subproblem

$$D_R(\lambda_2)=\min\left\{\mathbb{E}\left[\sum_{t=1}^{T}\Big\langle c_t+\textstyle\sum_{\tau=t}^{T}\lambda_{2\tau}w_{\tau,\tau-t},y_t\Big\rangle\right] \;\middle|\; \begin{array}{l} y_t\in Y_t,\\ H_t(y_t)=0 \end{array}\right\}.$$

Compared to the (dualized form of the) purely expectation-based problem (4.11), the subproblems for the x_i-blocks have the same structure, therefore the same solution methods can be applied. The only change consists in the additional factors $\sum_{\tau=t}^{T} \lambda_{2\tau}$ of $b_{it}(\xi_t)$ in the objective. If Y_1 is a cone, the subproblem for the additional y-block represents a cone constrained linear stochastic program and can be solved explicitly, namely, it holds

$$D_R(\lambda_2) = \begin{cases} \quad 0 & \text{if } -\left(c_t + \sum_{\tau=t}^{T} \mathbb{E}[\lambda_{2\tau}|\mathcal{F}_t] w_{\tau,\tau-t}\right) \in Y_t^* \ (t = 1, \ldots, T), \\ -\infty & \text{otherwise.} \end{cases}$$

Hence, the dual problem reads

$$\max \left\{ \sum_{i=1}^{I} D_i(\lambda_1, \lambda_2) - \mathbb{E}\left[\sum_{t=1}^{T} \langle \lambda_{1t}, d_t(\xi_t) \rangle\right] \;\middle|\; \begin{array}{l} \lambda_{1t} \in L_{p'}(\Omega, \mathcal{F}_t, \mathbb{P}; \mathbb{R}_+^{n_t}), \\ \lambda_{2t} \in L_{p'}(\Omega, \mathcal{F}_t, \mathbb{P}), \\ c_t + \sum_{\tau=t}^{T} \mathbb{E}[\lambda_{2\tau}|\mathcal{F}_t] w_{\tau,\tau-t} \in -Y_t^* \\ (t = 1, \ldots, T) \end{array} \right\}$$

and the whole Lagrangian decomposition strategy has the same favorable features for the risk averse model (4.12) as for the expectation-based one (4.11). For example, the known Lagrangian relaxation based algorithms for electricity portfolio optimization (e.g., [4, 12, 16]) apply to risk aversive models after some modifications.

5. Conclusions. We have introduced the class of polyhedral risk measures. Polyhedral risk measures are defined as optimal values of certain linear stochastic programs with recourse where the arguments appear on the right-hand sides of the dynamic constraints. By means of convex duality, criteria for coherence and second order stochastic dominance consistency have been deduced. For the one-period case it has been shown that well-known risk measures are contained in this class: Conditional-Value-at-Risk / quantile dispersion, and expected loss. For the multiperiod case, five polyhedral (coherent) risk measures were suggested.

Stochastic programs with a polyhedral risk measure as objective (or, alternatively, with an objective consisting of a linear combination of an expectation and a polyhedral risk measure) can be easily transformed into expectation-based stochastic programs. This observation has been used to demonstrate that important dual decomposition techniques known for certain expectation-based stochastic programs can be applied to stochastic programs with polyhedral risk measures after some modicifactions. The same is true for stability properties of stochastic programs.

Hence, for large scale problems possibly including integer variables polyhedral risk measures are a reasonable and flexible means to control risk while keeping the problems tractable.

Acknowledgments. The authors wish to thank the referees as well as the associate editor for valuable remarks and suggestions. Further thanks are due to Stephan Tiedemann (University of Duisburg-Essen) for helpful discussions.

REFERENCES

[1] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, *Coherent measures of risk*, Math. Finance, 9 (1999), pp. 203–228.

[2] P. Artzner, F. Delbaen, J.-M. Eber, D. Heath, and H. Ku, *Coherent Multiperiod Risk Measurement*, Working Paper, 2002, downloadable from www.math.ethz.ch/~delbaen.

[3] P. Artzner, F. Delbaen, J.-M. Eber, D. Heath, and H. Ku, *Coherent Multiperiod Risk Adjusted Values and Bellman's Principle*, Working Paper, 2004, downloadable from www.math.ethz.ch/~delbaen.

[4] L. Bacaud, C. Lemaréchal, A. Renaud, and C. Sagastizábal, *Bundle methods in stochastic optimal power management: A disaggregated approach using preconditioners*, Comput. Optim. Appl., 20 (2001), pp. 227–244.

[5] J. F. Bonnans and A. Shapiro, *Perturbation Analysis of Optimization Problems*, Springer Ser. Oper. Res., Springer-Verlag, New York, 2000.

[6] F. Delbaen, *Coherent risk measures on general probability spaces*, in Advances in Finance and Stochastics, K. Sandmann and P. Schönbucher, eds., Springer, Berlin, 2002, pp. 1–37.

[7] M. A. H. Dempster, ed., *Risk Management: Value at Risk and Beyond*, Cambridge University Press, Cambridge, UK, 2002.

[8] D. Dentcheva and W. Römisch, *Duality gaps in nonconvex stochastic optimization*, Math. Program., Ser. A, 101 (2004), pp. 515–535.

[9] A. Eichhorn, W. Römisch, and I. Wegner, *Polyhedral risk measures in electricity portfolio optimization*, PAMM Proc. Appl. Math. Mech., 4 (2004), pp. 7–10.

[10] H. Föllmer and A. Schied, *Convex risk measures and trading constraints*, Finance Stoch., 6 (2002), pp. 429–447.

[11] H. Föllmer and A. Schied, *Stochastic Finance: An Introduction in Discrete Time*, Walter de Gruyter, Berlin, 2002.

[12] N. Gröwe-Kuska, K. C. Kiwiel, M. P. Nowak, W. Römisch, and I. Wegner, *Power management in a hydro-thermal system under uncertainty by Lagrangian relaxation*, in Decision Making under Uncertainty: Energy and Power, C. Greengard and A. Ruszczyński, eds., IMA Vol. Math. Appl. 128, Springer, New York, 2002, pp. 39–70.

[13] S. Kusuoka, *On law invariant coherent risk measures*, in Advances in Mathematical Economics, Vol. 3, S. Kusuoka et al., eds., Springer, Tokyo, 2001, pp. 83–95.

[14] H. Markowitz, *Portfolio selection*, J. Finance, 7 (1952), pp. 77–91.

[15] A. Müller and D. Stoyan, *Comparison Methods for Stochastic Models and Risks*, Wiley, Chichester, UK, 2002.

[16] M. P. Nowak and W. Römisch, *Stochastic Lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty*, Ann. Oper. Res., 100 (2000), pp. 251–272.

[17] W. Ogryczak and A. Ruszczyński, *From stochastic dominance to mean-risk models: Semideviations as risk measures*, European J. Oper. Res., 116 (1999), pp. 33–50.

[18] W. Ogryczak and A. Ruszczyński, *On consistency of stochastic dominance and mean-semideviation models*, Math. Program., Ser. B, 89 (2001), pp. 217–232.

[19] W. Ogryczak and A. Ruszczyński, *Dual stochastic dominance and related mean-risk models*, SIAM J. Optim., 13 (2002), pp. 60–78.

[20] G. C. Pflug, *Some remarks on the value-at-risk and the conditional value-at-risk*, in Probabilistic Constrained Optimization, S. P. Uryasev, ed., Kluwer, Dordrecht, The Netherlands, 2000, pp. 272–281.

[21] G. C. Pflug, *The value of perfect information as a risk measure*, in Dynamic Stochastic Optimization, K. Marti, Y. Ermoliev, and G. C. Pflug, eds., Lecture Notes in Econom. and Math. Systems 532, Springer, Berlin, 2004, pp. 275–291.

[22] G. C. Pflug and A. Ruszczyński, *A risk measure for income processes*, in Risk Measures for the 21st Century, G. Szegö, ed., Wiley, Chichester, UK, 2004, pp. 249–269.

[23] S. T. Rachev, *Probability Metrics and the Stability of Stochastic Models*, Wiley, Chichester, UK, 1991.

[24] S. T. Rachev and W. Römisch, *Quantitative stability in stochastic programming: The method of probability metrics*, Math. Oper. Res., 27 (2002), pp. 792–818.

[25] F. Riedel, *Dynamic coherent risk measures*, Stochastic Process. Appl., 112 (2004), pp. 185–200.

[26] R. T. Rockafellar, *Conjugate Duality and Optimization*, CBMS-NSF Regional Conf. Ser. in Appl. Math. 16, SIAM, Philadelphia, 1974.

[27] R. T. Rockafellar and S. Uryasev, *Conditional value-at-risk for general loss distributions*, Journal of Banking & Finance, 26 (2002), pp. 1443–1471.

[28] R. T. Rockafellar, S. Uryasev, and M. Zarabankin, *Deviation Measures in Risk Analysis and Optimization*, Research Report 2002-7, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, 2002.

[29] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, Springer, Berlin, 1998.

[30] W. Römisch, *Stability of stochastic programming problems*, in Stochastic Programming, A. Ruszczyński and A. Shapiro, eds., Handbooks Oper. Res. Management Sci. 10, Elsevier, Amsterdam, 2003, pp. 483–554.

[31] W. Römisch and R. Schultz, *Multistage stochastic integer programs: An introduction*, in Online Optimization of Large Scale Systems, M. Grötschel, S. O. Krumke, and J. Rambau, eds., Springer, Berlin, 2001, pp. 581–622.

[32] W. Römisch and R. J.-B. Wets, *Stability of ε-Approximate Solutions to Convex Stochastic*

Programs, in preparation.

[33] A. Ruszczyński, *Decomposition methods in stochastic programming*, Math. Program., Ser. B, 79 (1997), pp. 333–353.

[34] A. Ruszczyński and A. Shapiro, eds., *Stochastic Programming*, Handbooks Oper. Res. Management Sci. 10, Elsevier, Amsterdam, 2003.

[35] A. Ruszczyński and A. Shapiro, *Optimization of convex risk functions*, Stochastic Programming E-Print Series, 6 (2004), paper 8, downloadable from www.speps.info.

[36] G. Scandolo, *Risk Measures for Processes and Capital Requirement*, Working Paper, Universitá degli Studi di Milano, Milan, Italy, 2003.

[37] R. Schultz, *Stochastic programming with integer variables*, Math. Program., Ser. B, 97 (2003), pp. 285–309.

[38] R. Schultz and S. Tiedemann, *Risk aversion via excess probabilities in stochastic programs with mixed-integer recourse*, SIAM J. Optim., 14 (2003), pp. 115–138.

[39] R. Schultz and S. Tiedemann, *Conditional value-at-risk in stochastic programs with mixed-integer recourse*, Stochastic Programming E-Print Series, 6 (2004), paper 20, downloadable from www.speps.info.

[40] A. Shapiro and S. Ahmed, *On a class of minimax stochastic programs*, SIAM J. Optim., 14 (2004), pp. 1237–1249.

[41] G. P. Szegö, ed., *Statistical and Computational Problems in Risk Management: VaR and Beyond VaR*, special issue of Journal of Banking & Finance, 26 (7) (2002).

[42] S. Weber, *Distribution-Invariant Dynamic Risk Measures*, Working Paper, Humboldt-University, Berlin, Germany, 2003.

[43] R. J.-B. Wets, *Stochastic programs with fixed recourse: The equivalent deterministic program*, SIAM Rev., 16 (1974), pp. 309–339.

SIAM J. Optim.
Vol. 16, No. 1, pp. 96–119

STRONG LIPSCHITZ STABILITY OF STATIONARY SOLUTIONS FOR NONLINEAR PROGRAMS AND VARIATIONAL INEQUALITIES[*]

DIETHARD KLATTE[†] AND BERND KUMMER[‡]

Abstract. The stationary solution map X of a canonically perturbed nonlinear program or variational condition is studied. We primarily aim at characterizing X to be locally single-valued and Lipschitz near some stationary point x^0 of an initial problem, where our focus is on characterizations which are explicitly given in terms of the original functions and assigned quadratic problems. Since such criteria are closely related to a nonsingularity property of the strict graphical derivative of X, explicit formulas for this derivative are presented, too. It turns out that even for convex polynomial problems our stability (and the Aubin property) does not depend only on the derivatives, up to some fixed order, of the problem functions at x^0. This is in contrast to various other stability concepts. Further, we clarify completely the relations to Kojima's strong stability and present simplifications for linearly and certain linearly quadratically constrained problems, convex programs, and for the map of global minimizers as well.

Key words. perturbed stationary solutions, nonlinear programs, variational conditions, strong Lipschitz stability, Aubin property

AMS subject classifications. 90C31, 49J40, 49J52, 26E25

DOI. 10.1137/030601569

1. Introduction.

Our basic model is the canonically perturbed nonlinear program

$$\text{P}(a,b) : \min_{x \in G(b)} f(x) - \langle a, x \rangle, \text{ where } G(b) = \{x \,|\, g(x) \le b\} \tag{1}$$

with parameter $p = (a, b) \in R^n \times R^m$, where f and $g = (g_1, \ldots, g_m)$ are defined on R^n and have (at least) locally Lipschitzian derivatives, briefly $f, g \in C^{1,1}$. Our main topic is that of strong Lipschitz stability (s.L.s.) of stationary points, and this only under the (inherent) Mangasarian–Fromowitz constraint qualification (MFCQ). Extensions to programs with additional parametric equations and to variational inequalities with a nonpolyhedral constraint set will be discussed too. For describing the crucial singular situations only, MFCQ will be weakened.

Let $p \mapsto X(p)$ and $p \mapsto X_{\text{KKT}}(p)$ denote the maps of stationary solutions x and Karush–Kuhn–Tucker (KKT-) points (x, y) to (1), respectively. Further, let B denote the closed unit ball.

Notions of stability. Given a multifunction $\Gamma : R^k \rightrightarrows R^s$ and some $z^0 \in \Gamma(p^0)$, Γ is said to be *strongly Lipschitz stable* (s.L.s.) at (p^0, z^0) if

$$\begin{array}{l} \text{for certain positive reals } \varepsilon, \delta, L \text{ and all } p, p' \in p^0 + \delta B, \\ \text{the sets } \Gamma_\varepsilon(p) := \Gamma(p) \cap (z^0 + \varepsilon B) \text{ are singletons } \{z(p)\} \\ \text{and fulfill } \|z(p') - z(p)\| \le L\|p' - p\|; \end{array} \tag{2}$$

[*] Received by the editors October 31, 2003; accepted for publication (in revised form) August 27, 2004; published electronically August 17, 2005.
http://www.siam.org/journals/siopt/16-1/60156.html

[†] Institut für Operations Research, Universität Zürich, Moussonstrasse 15, CH-8044 Zürich, Switzerland (klatte@ior.unizh.ch).

[‡] Institut für Mathematik, Humboldt–Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany (kummer@mathematik.hu-berlin.de).

i.e., Γ is locally single-valued and Lipschitz near (p^0, z^0). We aim to characterize strong Lipschitz stability of the map $\Gamma = X$ at $(0, x^0)$ *in terms of the given data functions and model structure.*

The behavior of stationary solutions to (1) has been already precisely described by parametric optimization provided that both the involved functions are of type C^3 and (1) belongs to some generic class of problems; cf. [11, 12]. Here, we study less smooth problems in general. The assumption $f, g \in C^{1,1} \backslash C^2$ is typical for problems which involve extreme-value functions of other (sufficiently regular) optimization models like in design- or semi-infinite optimization or for multilevel problems. For comparison, we introduce a second stability notion and call Γ *upper Lipschitz* at $(p^0, z^0) \in \operatorname{gph} \Gamma$ if

(3)
$$\Gamma_\varepsilon(p) \subset z^0 + L\|p - p^0\|B \text{ holds for certain positive } \varepsilon, \delta, L \text{ and all } p \in p^0 + \delta B.$$

If both (3) and $\Gamma_\varepsilon(p) \neq \emptyset$ hold with appropriate constants, then we call Γ *upper Lipschitz stable* at (p^0, z^0). The notions concerning stability or regularity differ in the literature. So "s.L.s." and "strongly regular" often mean the same thing, and our "upper Lipschitz" is "locally upper Lipschitz" in [6], while "upper Lipschitz stable" is "upper regular" in [18]. We avoid speaking of "regularity," since this has been used in an alternative definition via local linearizations in [36].

For the inverse map $\Gamma = F^{-1}$ of a C^1 function $F : R^n \to R^n$ with $F(z^0) = p^0$, all these properties coincide and require $\det DF(z^0) \neq 0$. If F is only locally Lipschitz, they are quite different.

MFCQ. Throughout the paper, x^0 denotes a fixed stationary point for $p = 0$, and constraint qualifications concern just this pair of points. The upper Lipschitz stability (even more s.L.s.) of X at $(0, x^0)$ implies for the constraint sets $G(b)$ and small $\|b\|$ that

$$G(b) \neq \emptyset \text{ and } \operatorname{dist}(x^0, G(b)) \leq L\|b\|. \tag{4}$$

With $b = -\varepsilon(1, \ldots, 1)^T$, the latter is possible only under the Mangasarian–Fromovitz constraint qualification (MFCQ). Similarly, MFCQ follows if equations $\gamma(x) = c \in R^{m'}$ additionally appear in the description of $G(\cdot)$: To see this, change also $c = \varepsilon v$ separately for both $v \in R^{m'}$ and $v \in \ker D\gamma(x^0)$. Therefore, MFCQ is a canonical assumption for characterizing these stabilities.

However, then the stability behavior essentially differs from that under the constraint qualification LICQ which makes Lagrange multipliers $y(p, x)$ on $\operatorname{gph} X$ unique and Lipschitz near $(x^0, 0)$ and thus implies that the Lipschitz conditions coincide for X and X_{KKT}.

The current state of art concerning Lipschitz stability of stationary solutions for problem (1) under the given canonical perturbations, and similarly for the model (9) below, can be summarized as follows.

1. Under related smoothness, all known (explicit) conditions for stability are invariant w.r.t. replacing the involved functions by quadratic approximations near the reference point x^0 (for h in (5) or (9) take the linearization). This fact remains true for the generic stability theory in [11, 12] as well as for generalized equations of the type

$$p \in h(x) + \mathcal{N}(x), \tag{5}$$

where $\mathcal{N}$ is a closed multifunction and h a C^1 function, and is the red line through the whole stability theory (not only) in optimization or variational analysis even if a second parameter z appears in h and $D_x h(\cdot,\cdot)$ is continuous: linearize $h(\cdot, z^0)$. This invariance principle was basically shown, for (1), (5), and (9), by Robinson [36]; in his paper, $\mathcal{N}$ had a particular structure, but the main proofs hold for any closed $\mathcal{N}$, too.

Further, the principle simplifies stability conditions up to a certain level which, for (5), essentially depends on the ability to work with $\mathcal{N}$. If $\mathcal{N}$ has a concrete structure involving again some vector function, say ψ, the same invariance can be observed in the literature; cf. [37, 7, 32, 42, 6, 18, 40]: In conditions for strong or upper Lipschitz behavior of the solutions, ψ may be replaced by its quadratic approximation at the reference point.

We shall show that this principle is not valid for s.L.s. of the stationary point map $X = X(a, b)$ even if f and g_i are convex polynomials, and we will characterize this stability in an analytical manner as simply as possible.

2. The kind of constraint qualification is important since known sufficient conditions (as far as they are verifiable in original terms) for X being s.L.s. require, directly or indirectly, that LICQ is satisfied at x^0. After using another description of the problems in [39, 40], LICQ became a formally weaker *nondegeneracy property.* But it ensures again that assigned Lagrange multipliers $y(p, x)$ are unique and Lipschitz. So it implies that s.L.s. of X and X_{KKT} coincide.

3. Further, s.L.s. of X_{KKT}, for (1) or (9), implies necessarily LICQ at x^0 and already has been completely characterized. So LICQ and the map X_{KKT} itself are of less interest in the following. More information concerning X_{KKT} can be found in section 4.

4. Under MFCQ, there exists a nearly complete theory for X or X_{KKT} to be *upper Lipschitz*, where $p = (a, b)$ or only $p = a$ is regarded as the canonical parameter [37, 41, 31, 14, 27, 28, 42, 16, 2, 26, 17, 18]. The results include formulas for computing protoderivatives (being contingent derivatives with a particular property) of the mappings in question.

For the particular mapping $a \mapsto X(a, 0)$, s.L.s. at some local minimizer x^0 satisfying MFCQ is characterized in [2, section 5.1] by a second order growth condition which is uniform w.r.t. small perturbations of (x, a).

5. Most of the stability (or regularity) characterizations, e.g., those in [27, 28, 42, 24, 25], cannot be applied to (1) and (9) if $g \in C^{1,1}\backslash C^2$ since second derivatives are decisively applied. Some related statements for such problems along with a calculus for the subsequent derivatives are elaborated, e.g., in [18] and earlier work of the authors. Here, the $C^{1,1}$ difficulties are hidden in the problem of determining the sets $\Phi(y, u)$ in Remark 3.2. For a deeper discussion of the resulting questions and partial results we refer to [8].

6. The way of dealing with other parameterizations (some parameter z appears in f, g, or h) has been basically clear since [36] was published: Rewrite the new variations in terms of canonical perturbations and apply the knowledge concerning the latter. This brings results and difficulties (concerning necessity of the conditions and existence of solutions) as in the context of implicit functions; cf. [20, 37, 22, 34, 42, 17, 2, 24, 18]. So, a basic counterexample from convex quadratic optimization in [37] shows that s.L.s. of X under canonical perturbations does not imply the analogue property if z appears linearly in g and LICQ does not hold. The reason for these difficulties, under MFCQ, will be discussed in section 5.4.

7. Further, several abstract conditions on s.L.s. of X or of the solution map Γ to (5) (cf. [42, 24, 25, 18]) are given in terms of generalized derivatives or by interrelations to the Aubin property which can be characterized by generalized derivatives [29], too.

However, in contrast to Fréchet-derivatives, generalized derivatives are defined via more or less complicated sets of (double) limits, which make the calculus and direct applications quite hard. So, to mention only one difficulty, the derivatives $D = T$, $D = C$ below do not satisfy the addition rule $D(f + g) = Df + Dg$ for arbitrary real Lipschitz functions. The same problem occurs for Clarke's generalized Jacobians and coderivatives and even more for generalized derivatives of multifunctions. Therefore, the value of any stability characterization via generalized derivatives crucially depends on the ability to determine the latter ones for the problem in question. A typical derivative characterization is given by Theorem 2.3 below, and our main intent is just to show how to overcome its insufficiency and to exploit the problem structure.

The approach and content of the paper can be summarized as follows.

Negation of stability. For violating strong Lipschitz stability of X at $(0, x^0)$, there are exactly two (not disjoint) possibilities which arise from the negation of this property:

(6)

Local unsolvability: $\exists \varepsilon > 0$ such that $X(p^k) \cap (x^0 + \varepsilon B) = \emptyset$ for certain $p^k \to 0$.

Far solutions 1: $\exists x^k \in X(p^k)$, $\xi^k \in X(\pi^k)$ with x^k, $\xi^k \to x^0$ and p^k, $\pi^k \to 0$ such that $x^k \neq \xi^k$ and $\|\pi^k - p^k\| / \|\xi^k - x^k\| \to 0$.

Definition 1.1. *If the first situation does not happen, we will speak of* local solvability. *If the second situation happens, we will speak of* singularity.

Singularity means, in terms of a generalized derivative TX (cf. Definition 2.1 of the strict graphical derivative), that $u \in TX(0, x^0)(0)$ for some $u \neq 0$. This well-known fact, mentioned, e.g., in [21, 42, 24, 18], immediately follows with every cluster point u of $u^k = \frac{\xi^k - x^k}{\|\xi^k - x^k\|}$ and is nothing more than a motivation of the definition of TX. Nevertheless, one may try to characterize TX in terms of the original data in order to simplify the singularity condition. This is precisely both the main problem for applying any generalized derivative (which often describes some property only in another, more compact, form) and the kernel of our analysis in section 3, where we extend the calculus developed for $C^{1,1}$ problems (1) in [21, 22, 18]. For characterizing singularity, we weaken the MFCQ-supposition by requiring throughout the paper that

(7) for every sequence $(p^k, x^k) \in \operatorname{gph} X$ with $(p^k, x^k) \to (0, x^0)$, there is some subsequence such that certain assigned Lagrange multipliers y^k converge.

This covers (piecewise) linear constraints and several other constraint qualifications.

The crucial results of the paper. After preparations in section 2, we shall present and prove our main Theorem 3.1 in section 3. For C^2 problems (1), let $F_1(x, y) = Df(x) + \sum_i y_i Dg_i(x)$ denote the first derivative w.r.t. x of the Lagrangian and let Y^0 denote the set of Lagrange multipliers to x^0 at $p = 0$. Further, put $J = \{i | g_i(x^0) = 0\}$ and $Q(y) = D_x F_1(x^0, y)$. Then Theorem 3.1 characterizes singularity, under (7), as follows:

(8) *There exist* $u \neq 0$ *and* $y \in Y^0$ *such that* $y_i Dg_i(x^0)u = 0\ \forall i$ *and with certain sequences* $x^k \to x^0$ *and* $\alpha^k \in R^m$, *one has* $\alpha_i^k Dg_i(x^0)u \geq 0\ \forall i \in J$ *and* $Q(y)u + \sum_{i \in J} \alpha_i^k Dg_i(x^k) \to 0$.

By (8) and Corollary 5.3, strong Lipschitz stability at a local minimizer x^0 to $p = 0$ will be completely described. The key point concerning LICQ or assumption (7) at

x^0 can be seen as follows. Under LICQ, the sequence α^k in (8) cannot diverge. So one may select some cluster point α^0 to obtain a much simpler equivalent condition (namely, the negation of TF-injectivity w.r.t. u; cf. section 4) by setting $\alpha^k = \alpha^0$, $x^k = x^0$.

Theorem 3.1 also applies to the solution map X of parametric variational conditions

$$(9) \qquad a \in \ h(x) + N_{G(b)}(x), \ x \in G(b)$$

with parameter $p = (a, b)$, where $h \in C^1(R^n, R^n)$, $g \in C^2(R^n, R^m)$, and $N_{G(b)}(x)$ is the B-normal cone at x to the set $G(b)$ in (1). In this case, we speak of the C^2 *problem* (9).

Finally (cf. Remark 3.2), Theorem 3.1 similarly holds for $C^{1,1}$ problems (1) and for (9) if h is locally Lipschitz and $g \in C^{1,1}$. Then we also call (9) a $C^{1,1}$ problem.

As long as assumption (7) remains true, additional equations $g_i = 0$ may appear in the description of the feasible sets $G(b)$. For any related i, then also $Dg_i(x^0)u = 0$ must be required in (8), so α_i^k is not restricted by sign.

Condition (8) is something unsatisfactory due to the included sequences. Defining the cone $K(x^0, u) = \limsup_{x \to x^0} \{\sum_{i \in J} \alpha_i Dg_i(x) \mid \alpha_i Dg_i(x^0)u \geq 0 \, \forall i \in J\}$ via the upper Hausdorff (or Kuratovski–Painlevé) limit of sets, one can hide the limits in the equivalent condition that there are $u \neq 0$ and $y \in Y^0$ with $y_i Dg_i(x^0)u = 0 \ \forall i$ and $-\,Q(y) \in K(x^0, u)$. This is elegant but no simplification, of course.

So we derive *in section* 4 a second characterization of singularity by classical means of canonically perturbed quadratic problems (Theorem 4.2) and we ask whether the limit characterization can be simplified or replaced, under smoothness, by using only Fréchet-derivatives (up to some fixed order) of f, g, or h at x^0. We call such a condition a *finite derivative condition.*

Though various limit characterizations appear in nonsmooth analysis, the discussion of this question is not standard. Our Examples 4.7 and 4.8 demonstrate that, given any natural $q > 0$, there are two convex polynomial problems (1), uniquely solvable for all parameters, such that the derivatives up to order q of f and g coincide at x^0, though one problem is singular and the other is not.

Hence there is no singularity characterization by a finite derivative condition.

For other regularity or stability notions to problem (1) or (9), this phenomenon does not appear or has not been observed up to now. Because of unique solvability in our examples, also the Aubin property in section 4.3 permits no singularity characterization by a finite derivative condition for convex problems. It needs a limit condition although related (coderivative) criteria are sometimes called *point conditions.* We include the Aubin property in this paper since it yields local solvability; cf. Theorem 4.11.

In section 5, we discuss special cases in which a finite derivative condition still exists and compare (8) with criteria for upper Lipschitz stability and Kojima's strong stability [20]. In particular, the relations between *s.L.s.* and Kojima's stability will be completely clarified. Some results on *local solvability*, which are closely connected with these stabilities, will be mentioned in section 5.2. In this respect, there are differences for the problems under consideration. In particular for (9), or for (1) if $f, g \in C^{1,1} \backslash C^2$, such investigations need more effort and are not the focus of this paper. However, without having local solvability, the estimate in (2) still holds for all existing solutions, provided that singularity does not hold.

2. Preliminaries.

2.1. Kojima's function. For technical reasons we describe the KKT-points, following Kojima [20] as zeros of some function F given by the components

$$\begin{aligned} F_1(x,y) &= Df(x) + \sum_{i=1}^{m} y_i^+ Dg_i(x), \;\; y_i^+ = \max\{0, y_i\}, \\ F_{2,i}(x,y) &= g_i(x) - y_i^-, \;\; y_i^- = \min\{0, y_i\}. \end{aligned} \tag{10}$$

With the usual Lagrangian L, we have $F_1(x,y) = D_x L(x, y^+)$. For each zero of F, (x, y^+) is a usual KKT-point, and negative y_i coincide with $g_i(x)$. The KKT-points of (1) correspond to the equation $F(x,y) = p$; we put $X_{\text{KKT}}(p) = \{(x,y) \mid F(x,y) = p\}$. Throughout, let

$$Y(p,x) = \{y \mid F(x,y) = p\} \quad \text{and} \quad Y^0 = Y(0, x^0).$$

As is well known, MFCQ at a solution $(0, x^0)$ just ensures that Y^0 is nonempty and bounded, and hence the mapping $Y(\cdot,\cdot)$ is upper semicontinuous at $(0, x^0)$.

Additional equality constraints. For simplicity of presentation, we restrict ourselves to inequality constraints, though constraints $g_\nu(x) = b_\nu$, $\nu = m+1, \dots, \kappa$, can be included without affecting the results. The related Kojima function F then also contains the sum $\sum_\nu y_\nu Dg_\nu(x)$ in F_1 and new components $F_{3,\nu} = g_\nu(x)$. In every subsequent statement where the signs of y_i play any role, y_ν must be handled like positive y_i (even if $y_\nu \leq 0$). In the conditions of Theorem 3.1, additional components α_ν and β_ν will appear where α_ν is not restricted by sign and $\beta_\nu = 0$.

Analysis of variational inequalities. As in [18], our analysis of Lipschitz stability can be applied to the solution map X of variational conditions (9) of C^2 or $C^{1,1}$ type. Provided that $x \in G(b)$ satisfies MFCQ (or any weaker constraint qualification), it is known that x solves (9) if and only if $\widetilde{F}(x,y) = p$ holds for some $y \in R^m$, where $\widetilde{F}$ is defined by substituting $h(x)$ for $Df(x)$ in (10) only. As before, we call the related pairs (x,y) KKT-points and denote the corresponding solution map by X_{KKT}.

In the following, when characterizing the derivative TX and s.L.s., *we will never make use of the special form $h = D_x f$* if nothing else is written explicitly. Thus, this variation of F_1 allows us to deal with (1) and (9) in the same way (of course, if symmetry of Dh or the background of optimization is not needed). Again, $C^{1,1}$ equations may occur in $G(b)$ as long as MFCQ or (7), respectively, holds true.

For studying derivatives, it is essential that F in (10) has a separable product form

$$F = N(y)M(x), \quad \text{where } N(y) = (1, y^+, y^-) \in R^{1+2m} \tag{11}$$

has (simple) directional derivatives, while $M(x)$ is a matrix of size $(1+2m, n+m)$:

$$M(x) = \begin{pmatrix} Df(x)^{\mathsf{T}} & g_1(x) & \dots & g_m(x) \\ Dg_1(x)^{\mathsf{T}} & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ Dg_m(x)^{\mathsf{T}} & 0 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & -1 \end{pmatrix}.$$

Evidently, $M \in C^1$ holds for C^2 problems (1) and (9) while M is locally Lipschitz in the $C^{1,1}$ case. Note that additional equality constraints simply lead us to a larger matrix and to additional (smooth) components y_ν in N.

2.2. The derivatives $T\Gamma, C\Gamma$. It is well known that strong Lipschitz stability of a (multi)function $\Gamma : R^m \rightrightarrows R^n$ is closely related to nonsingularity of certain limits, e.g., [21, 22, 42, 24, 25, 18]. We recall the following definition.

DEFINITION 2.1 (strict graphical derivative $T\Gamma$). *Let $x^0 \in \Gamma(p^0)$ and $c \in \mathbb{R}^m$. Then the set $T\Gamma(p^0, x^0)(c)$ consists of all $u \in R^n$ such that $x^k \in \Gamma(p^k)$ and $x^k + t_k u_k \in \Gamma(p^k + t_k c^k)$ hold for certain $t_k \downarrow 0$ and related sequences $(x^k, u^k, p^k, c^k) \to (x^0, u, p^0, c)$.*

Sets of this form have been called *strict graphical derivatives* in [42]. For functions Γ, they were introduced by Thibault (to define other types of generalized derivatives) and denoted by *limit sets* in [44, 45], and they appeared, e.g., in [21, 16, 18] as Δ- or *T-derivatives*.

Setting, in the case of "far solutions 1" (6), $\xi^k = x^k + t_k u^k$, where $\|u^k\| = 1$ and $t_k > 0$, and selecting a subsequence such that $u^k \to u$, one obtains

$$\text{some } u \neq 0 \text{ belongs to } TX(0, x^0)(a, b) \text{ for } (a, b) = 0, \tag{12}$$

and vice versa. Hence, singularity and (12) coincide.

DEFINITION 2.2 (contingent derivative $C\Gamma$). *The set of all $u \in R^n$ such that $x^0 + t_k u_k \in \Gamma(p^0 + t_k c^k)$ holds for certain $t_k \downarrow 0$ and $(u^k, c^k) \to (u, c)$ forms the* contingent derivative, also called the Bouligand derivative $C\Gamma(p^0, x^0)(c)$; *cf.* [1].

Due to the symmetry w.r.t. images and preimages, the derivative $T\Gamma^{-1}$ or $C\Gamma^{-1}$ of the inverse Γ^{-1} is just the inverse of $T\Gamma$ or $C\Gamma$, respectively.

The negation of upper Lipschitz stability is described by the two (not disjoint) possibilities *local unsolvability as under* (6) and

$$\begin{array}{ll} \textit{Far solutions } 2: & \exists \xi^k \in X(\pi^k) \text{ with } \xi^k \to x^0 \text{ and } \pi^k \to 0 \\ & \text{such that } \xi^k \neq x^0 \;\text{ and }\; \|\pi^k\|/\|\xi^k - x^0\| \;\to\; 0. \end{array} \tag{13}$$

The singular situation (13) means, by taking a cluster point u of $u^k = \frac{\xi^k - x^0}{\|\xi^k - x^0\|}$, that

$$\text{some } u \neq 0 \text{ belongs to } CX(0, x^0)(a, b) \text{ for } (a, b) = 0. \tag{14}$$

This is just the negation of CF-injectivity w.r.t. u; cf. section 4. As mentioned in the introduction, the derivative CX has already been exactly determined. We shall recall related results for comparison and for obtaining a bridge to local solvability; cf. (18) and Corollary 5.3.

Here, we have $\Gamma(p) = X(p)$, $p = (a, b)$, and $p^0 = 0$ and we may put $u^k = u$ $(\forall k)$ in the definition of $TX(0, x^0)(c)$. Indeed,

$$x^k + t_k u^k \in X(p^k + t_k c^k) \text{ means } F(x^k + t_k u^k, y^k) = p^k + t_k c^k \text{ for some } y^k,$$

where, under (7), y^k may be seen as bounded for (p^k, x^k) near $(0, x^0)$ and small positive t_k (otherwise choose an appropriate subsequence). Further, F is locally Lipschitz. So, replacing u^k by u leads us to

$$F(x^k + t_k u, y^k) = p^k + t_k c^k + t_k r^k, \text{ where } r^k \to 0. \tag{15}$$

Thus, with $u^k \equiv u$ one obtains the same set $TX(0, x^0)(c)$. Concerning $CX(0, x^0)(c)$, where $x^k \equiv x^0, p^k \equiv 0$, the same arguments are valid.

Similarly, these derivatives can be applied to the function F and their factors M, N in (11). Then, the usual product rule of differentiation holds for these derivatives applied to F [21, 22, 18], and $TM(x^0)(u)$ consists of all limits of $\phi_k = t_k^{-1}[M(x^k + t_k u) - M(x^k)]$ where $t_k \downarrow 0$ and $x^k \to x^0$. Further, TN and CN have a simple form due to the structure of $c_i(y) = (y_i^+, y_i^-) = (y_i^+, y_i - y_i^+)$:

$$TN(y)(v) = \{(0, r, s) \,|\, r + s = v,\ r_i s_i \geq 0\ s_i y_i^+ = 0,\ r_i y_i^- = 0\ \forall i\},$$

$$CN(y)(v) = \{(0, r, s) \in TN(y)(v) \,|\, s_i = 0 \text{ if } y_i = 0 \leq v_i,\ \ r_i = 0 \text{ if } y_i = 0 \geq v_i\}.$$

The (full) images $T_N(y) := TN(y)(R^m)$ and $C_N(y) := CN(y)(R^m)$ of these maps are polyhedral cones in R^{1+2m}:

$$T_N(y) = \{(0, \alpha, \beta) \,|\, \alpha_i \beta_i \geq 0,\ \beta_i y_i^+ = 0,\ \alpha_i y_i^- = 0\ \forall i\}, \tag{16}$$

$$C_N(y) = \{(0, \alpha, \beta) \in T_N(y) \,|\, \alpha_i \geq 0 \geq \beta_i \text{ if } y_i = 0\}, \tag{17}$$

and $CN(y)(v)$ consists of the directional derivative $N'(y; v)$ of N at y in direction v. The set $T_N(y)$ coincides with the image of Clarke's [3, 4] generalized Jacobian $\partial N(y)$. *The reader who is less familiar with such derivatives may substitute these sets, via* (16) *and* (17) *by complementarity conditions in all subsequent formulas.*

2.3. First characterization of $\boldsymbol{TX}$. Before dealing with the double limits in TX, let us recall the simpler form of CX for $C^{1,1}$ problems (1) and (9). It holds that

$$\begin{array}{l} u \in CX(0, x^0)(a, b) \ \text{ iff } \ \text{there are } y \in Y^0 \text{ and } (\alpha, \beta) \in R^{2m} \text{ satisfying} \\ a \in CF_1(\cdot, y)(x^0)(u) + \sum_i \alpha_i Dg_i(x^0)\,, \quad Dg(x^0)u - \beta = b, \quad \text{and} \\ (0, \alpha, \beta) \in C_N(y). \end{array} \tag{18}$$

Now we give a first description of TX. For problems (1) under MFCQ, it was already shown in [18].

THEOREM 2.3. *For $C^{1,1}$ problems* (1) *or* (9) *satisfying* (7)*, it holds that $u \in TX(0, x^0)(a, b)$ if and only if there exist some sequence $t = t_k \downarrow 0$ and assigned converging points $x \to x^0$, $y \to y^0$, $y' \to y^1$ (all depending on $k \to \infty$) such that y^0, $y^1 \in Y^0$ and*

(i) *$t^{-1}[M(x + tu) - M(x)] \to M_\circ$ for some $M_\circ \in TM(x^0)(u)$, and*

(ii) *$N(y^1)M_\circ + t^{-1}[N(y') - N(y)]M(x) \to (a, b)$.*

The statement shows the kind of conditions that arise by exploiting the definition of TX and the structure (11) of F only, and it demonstrates the need for better reformulations in terms of the original data. Before proving the statement we will interpret the occurring terms and conditions. The terms $t^{-1}[N(y') - N(y)]M(x)$ depend on Dg only and have the form

$$\begin{array}{rcll} t^{-1} & \sum_{i=1}^m & (y'^+ - y^+)_i Dg_i(x) & \text{assigned to } F_1, \\ -t^{-1} & & (y'^- - y^-) & \text{assigned to } F_2. \end{array}$$

For $M \in C^1$, it clearly follows that

$$M_\circ = DM(x^0)u \ \text{ and } \ N(y^1)M_\circ = (D_x F_1(x^0, y^1)u, Dg(x^0)u) \in R^{n+m}. \tag{19}$$

If M is only locally Lipschitz, $M_\circ$ may depend on the sequences t_k, x^k under consideration.

The point (x, y^+) is a KKT-point for problems (1), (9) with perturbation $p = F(x, y)$, while $(x', y'^+) = (x + tu, y'^+)$ is a KKT-point for (1), (9) with $p' = F(x', y')$.

The singularity condition expresses the fact that $\|p' - p\| \ll \|x' - x\|$ provided that $t \downarrow 0$, where "$\ll$" is to read as *much smaller*. Condition (15) tells us that $p' - p = t(a, b) + o(t)$. These relations have to be true *for some discrete sequence* $t = t_k \downarrow 0$ *only.*

Proof of Theorem 2.3. ($\Rightarrow$) Let $u \in TX((0, 0), x^0)(a, b)$. It holds (at least for some subsequence), with $x' = x + tu$ and certain assigned y, y' having accumulation points $y \to y^0 \in Y^0$, $y' \to y^1 \in Y^0$, that

$$\begin{aligned} t(a,b) + o(t) &= F(x', y') - F(x, y) \\ &= N(y')M(x') - N(y)M(x) \\ &= N(y')[M(x') - M(x)] + [N(y') - N(y)]M(x). \end{aligned}$$

After division by t (and selecting an appropriate subsequence), one sees that

$$\begin{aligned} t^{-1}[M(x') - M(x)] &\to M_\circ \in TM(x^0)(u), \\ t^{-1}N(y')[M(x') - M(x)] &\to N(y^1)M_\circ. \end{aligned}$$

So the limit of

$$\Lambda = t^{-1}[N(y') - N(y)]M(x) \tag{20}$$

exists and satisfies $N(y^1)M_\circ + \Lambda \to (a, b)$, indeed.

($\Leftarrow$) Having sequences as in the theorem, it follows that

$$\begin{aligned} F(x', y') - F(x, y) &= N(y')[M(x') - M(x)] + [N(y') - N(y)]M(x) \\ &= N(y')[tM_\circ + o_1(t)] + t\Lambda \\ &= N(y')[tM_\circ + o_1(t)] + t[(a, b) - N(y^1)M_\circ] + o_2(t) \\ &= t(a, b) + o_3(t), \end{aligned}$$

where o_i denote different o-type functions. This finishes the proof. □

Comments.

1. Similarly, based on the product form of F, (simpler) formulas for $CX(0, x^0)(c)$, $TX_{\mathsf{KKT}}(0, (x^0, y^0))(c)$, and $CX_{\mathsf{KKT}}(0, (x^0, y^0))(c)$ can be derived.

2. If a further parameter z appears in f, g, or h one may ask for the derivative $TX((0, 0, 0), x^0)(a, b, \zeta)$. Now, one has $z' - z = t\ \zeta + o(t)$, and F depends on z only via $M = M(x, z)$. Having $M \in C^1$, one concludes as above that the term $N(y^1)M_\circ = N(y^1)DM(x^0)u$ must be replaced by $N(y^1)(D_xM(x^0, 0)u + D_zM(x^0, 0)\zeta)$ only.

3. Difficulties for simplifying the conditions of Theorem 2.3 arise from the fact that the speed of convergence of y, y', x and t may be different and that $y^0 \neq y^1$ is possible.

3. The main theorem. Here, we show that the singularity condition can be written in a way much simpler than what was done in Theorem 2.3.

Since we consider only points sufficiently close to $x^0 \in X(0)$, any constraint being not active at x^0 is also inactive at x near x^0 and *can be completely deleted* in our models. For this reason, we assume in this section (without loss of generality) that $g(x^0) = 0$. Then we have $y \geq 0\ \forall y \in Y^0$. Throughout, we also suppose (7) and put

$$A = Dg(x^0) \text{ with rows } A_i = Dg_i(x^0)^\mathsf{T} \quad \text{and} \quad Q(y) = D_xF_1(x^0, y). \tag{21}$$

Theorem 3.1. *For C^2 problem* (1) *or* (9), *singularity is equivalent to the following condition:*

$$\begin{array}{l}\textit{There exist } u \neq 0 \textit{ and } y \in Y^0 \textit{ with } y_i A_i u = 0\ \forall i \textit{ such that,}\\ \textit{for certain sequences } \ x^k \to x^0 \ \ \textit{and} \ \ \alpha^k \in R^m, \textit{ one has}\\ \alpha_i^k A_i u \geq 0\ \forall i \ \ \textit{and} \ \ Q(y)u + Dg(x^k)^{\mathsf{T}} \alpha^k \to 0.\end{array} \tag{22}$$

Moreover, if $b \in \operatorname{Im} A$, then $u \in TX(0, x^0)(a, b)$ means equivalently that

$$\begin{array}{l}\textit{there exist } \ y \in Y^0 \textit{ and sequences } x^k \to x^0, \alpha^k \in R^m \textit{ with } y_i\beta_i = 0\ \forall i,\\ \beta = Au - b, \ \ \alpha_i^k\ \beta_i \geq 0\ \forall i \ \ \textit{and} \ \ Q(y)u + Dg(x^k)^{\mathsf{T}} \alpha^k \to a.\end{array} \tag{23}$$

Remark 3.2. For M being locally Lipschitz, Theorem 3.1 remains true if one replaces $Q(y)u$ by any element $\phi \in \Phi(y, u) := TF_1(\cdot, y)(x^0)(u) \subset R^n$ and if the sequence of x^k is assigned to $t_k \downarrow 0$ such that, in accordance with Theorem 2.3, ϕ is just the limit of $t_k^{-1}[F_1(x^k + t_k u, y) - F_1(x^k, y)]$ as $k \to \infty$.

However, for general $C^{1,1}$ problems, computing or describing $\Phi(y, u)$ is a hard task even if Df, Dg, or h is continuous and piecewise linear. Taking $y \geq 0\ \forall y \in Y^0$ and formula (16) into account, we have, with $Au = \beta$,

$$(0, \alpha, \beta) \in T_N(y) \Leftrightarrow y_i A_i u = 0 \quad \text{and} \quad \alpha_i A_i u \geq 0\ \forall i. \tag{24}$$

So the singularity condition (22) means, in terms of T_N, that

$$\begin{array}{l}\text{there exist } u \neq 0,\ y \in Y^0 \text{ such that for certain sequences } x^k \to x^0 \text{ and}\\ \alpha^k \in R^m \text{ with } (0, \alpha^k, Au) \in T_N(y), \text{ one has } \ Q(y)u + Dg(x^k)^{\mathsf{T}} \alpha^k \to 0.\end{array} \tag{25}$$

Proof of Theorem 3.1 *and Remark* 3.2. In the remainder of this section, we verify both the second statement of Theorem 3.1 (the first one then follows from the equivalent singularity condition (12)) and Remark 3.2. This will be done by simplifying the condition in Theorem 2.3.

The key consists of studying the limit of (20) $\Lambda = t^{-1}[N(y') - N(y)]M(x)$ in condition (ii) of that theorem and in showing that one may put $y^0 = y^1$ in order to describe all limits in Theorem 2.3 (and to eliminate t_k for C^2 problems). In what follows we abbreviate

$$N^1 = N(y^1), \quad N^\circ = N(y^0).$$

Part 1: *The cone $N'(y^1;\ \mathbb{R}^m) - N'(y^0;\ \mathbb{R}^m)$.* Recall that $N'(y; v)$ denotes the directional derivative of N (11) at y in direction v. Setting $y' = y^1 + v',\ y = y^0 + v$, then, since N is piecewise linear, the formula

$$N(y') - N(y) = N^1 - N^\circ + N'(y^1; v') - N'(y^0; v)$$

holds for a small norm of v' and v. With $q = N'(y^1; v') - N'(y^0; v)$ we thus obtain

$$\Lambda = t^{-1}[N^1 - N^\circ + q]M(x). \tag{26}$$

So the differences q become important for fixed y^0, y^1. These elements have the form

$$q = (0, r, s) \in R^{1+2m} \ \text{ and } \ q \in D(y^1, y^0) := N'(y^1;\ \mathbb{R}^m) - N'(y^0;\ \mathbb{R}^m).$$

The set $\mathcal{D} = D(y^1, y^0)$ is a polyhedral cone. Since $y^0 \geq 0$, the images of the directional derivatives $c_i'(y_i^0;\ R)$ for the m functions $c_i(y_i) = (y_i^+, y_i^-)$ have the form

$$c_i'(y_i^0;\ R) = \begin{cases} (R, 0) & \text{if } y_i^0 > 0, \\ \{(\alpha, \beta) \in R^2 \mid \beta \leq 0 \leq \alpha,\ \alpha\beta = 0\} & \text{if } y_i^0 = 0. \end{cases}$$

The same holds for y^1. So the elements $q = (0, r, s)$ with $(r_i, s_i) \in D_i = D_i(y^1, y^0) := c_i'(y_i^1;\ R) - c_i'(y_i^0;\ R)$ build the set $\mathcal{D}$, where the sets $D_i \subset R^2$ are nothing but

Case 1 $\quad D_i = \{(r_i, s_i) \mid r_i s_i \geq 0\}$ if $y_i^1 = 0,\ y_i^0 = 0$,
Case 2 $\quad D_i = \{(r_i, s_i) \mid\ s_i \leq 0\}$ if $y_i^1 = 0,\ y_i^0 > 0$,
Case 3 $\quad D_i = \{(r_i, s_i) \mid\ s_i \geq 0\}$ if $y_i^1 > 0,\ y_i^0 = 0$,
Case 4 $\quad D_i = \{(r_i, s_i) \mid\ s_i = 0\}$ if $y_i^1 > 0,\ y_i^0 > 0$.

Now we can characterize the terms Λ of (26) in more detail, where again $A_i = Dg_i(x^0)^{\mathsf{T}}$:

$$qM(x) = \left(0,\ \sum_{i=1}^{m} r_i Dg_i(x)^{\mathsf{T}}, -s_1, \ldots, -s_m\right), \tag{27}$$

$$[N^1 - N^\circ]M(x) = \left(0,\ \sum_{i=1}^{m} (y_i^1 - y_i^0) Dg_i(x)^{\mathsf{T}}, 0, \ldots, 0\right). \tag{28}$$

Finally, recalling (19) one has $N^1 DM(x^0)u = D_x F(x^0, y^1)u = (Q(y^1)u, Au) \in R^{n+m}$. Knowing these terms, we obtain the following corollary from Theorem 2.3.

Corollary 3.3. *For C^2 problems, it holds that $u \in TX(0, x^0)(a, b)$ if and only if for certain y^0, $y^1 \in Y^0$ as well as some sequence $t = t_k \downarrow 0$ and assigned converging points $x \to x^0$ and $d = (d_1, \ldots, d_m) \to 0$ with $d_i = (r_i, s_i) \in D_i(y^1, y^0)$ ($\forall i$), one has*

$$t^{-1} \sum_{i=1}^{m} (y_i^1 - y_i^0 + r_i) Dg_i(x) \to a - Q(y^1)u, \tag{29}$$

$$-t^{-1} s \to b - Au. \tag{30}$$

For $C^{1,1}$ problems, one has to replace $Q(y^1)u$ by some $\phi \in \Phi(y, u)$, and the sequences of x and t have to correspond to each other as in Remark 3.2.

Part 2: *Simplifying s.* We specify s by requiring that

$$s_i = t\beta_i, \text{ where } \beta = Au - b \tag{31}$$

and show that this additional condition in Corollary 3.3 does not shrink the set of possible limits. Since s_i does not appear in (29), it is only important for defining the restrictions to r_i via $(r_i, s_i) \in D_i(y^1, y^0)$. These restrictions remain the same after replacing s_i by any $\lambda_i s_i$ ($\lambda_i > 0$).

If $\underline{\beta_i \neq 0}$, condition (30) yields $s_i = t\beta_i + o_i(t)$ and $\operatorname{sign} \beta_i = \operatorname{sign} s_i$. So, the setting (31) along with the requirement

$$(r_i, \beta_i) \in D_i(y^1, y^0) \tag{32}$$

leads us to the same (sign-) conditions for r_i.

If $\underline{\beta_i = 0}$ and $s_i = 0$ in the corollary, then, due to the particular form of D_i, the restrictions concerning the sign of r_i disappear. So the same variation of r_i as before is possible. Thus, the sequence of s_i may be specified by (31) for all i whereafter the conditions $(r_i, s_i) \in D_i(y^1, y^0)$ and (32) coincide.

Part 3: *Simplifying* $\gamma := y^1 - y^0 + r$ *dependent on* $\beta = Au - b$.

Lemma 3.4. *In the characterization of Corollary* 3.3, *both of the following hold:*

(i) $\beta_i \neq 0 \Rightarrow y_i^1 = y_i^0 = 0$, *provided that* $b \in \operatorname{Im} A$,

(ii) $\gamma_i \beta_i \geq 0 \ \forall i$ *if* $\|r\|$ *is small enough.*

Proof. For the purpose of this proof, the assumption $b \in \operatorname{Im} A$ is used. We may already suppose (31) and (32).

Let $\underline{\beta_i < 0}$. By the structure of D_i, the situation $\beta_i < 0$ is allowed in two cases. In the first case, $y_i^1 = 0$ and $y_i^0 > 0$ hold true. Then $y_i^1 - y_i^0 < 0$, and there is no sign restriction for r_i; so it follows that

$$(y_i^1 - y_i^0)\beta_i > 0 \ \text{ and } \gamma_i < 0 \text{ for } |r_i| \text{ small enough.}$$

In the second case, $y_i^1 = 0 = y_i^0$ holds true. Then $r_i \leq 0$ and hence $\gamma_i \leq 0$ have to hold. Thus,

$$\beta_i < 0 \ \Rightarrow y_i^1 = 0 \ \text{ and } \ \gamma_i \leq 0.$$

Let $\underline{\beta_i > 0}$. Again, the situation $\beta_i > 0$ is allowed in two cases. In the first one, $y_i^1 > 0$ and $y_i^0 = 0$ are valid. Then $y_i^1 - y_i^0 > 0$ and there is no sign restriction for r_i. So it follows that

$$(y_i^1 - y_i^0)\beta_i > 0 \ \text{ and } \gamma_i > 0 \text{ for } |r_i| \text{ small enough.}$$

In the second case, $y_i^1 = 0 = y_i^0$, it now follows that $r_i \geq 0$ and $\gamma_i \geq 0$. Thus,

$$\beta_i > 0 \Rightarrow y_i^0 = 0 \ \text{ and } \ \gamma_i \geq 0.$$

Other cases are not possible. Therefore, it holds in all cases that

$$\gamma_i \beta_i \geq 0 \ \text{ and } \ (y_i^1 - y_i^0)\beta_i \geq 0 \ \forall i. \tag{33}$$

Due to $y^0, y^1 \in Y^0$, we also have $A^{\mathsf{T}}(y^1 - y^0) = 0$. Since $b = Au - \beta \in \operatorname{Im} A$, there is some $w \in R^n$ such that $\beta = Aw$ and

$$(y^1 - y^0)^{\mathsf{T}} \beta = 0.$$

Therefore, the first case for both signs of $\beta_i \neq 0$ cannot occur. Indeed, otherwise one obtains from (33), $(y_{i_0}^1 - y_{i_0}^0)\beta_{i_0} > 0$ for some i_0 and $(y^1 - y^0)^{\mathsf{T}}\beta > 0$, a contradiction. So $\beta_i \neq 0$ is possible only for $y_i^1 = y_i^0 = 0$. This completes the proof. □

Part 4: *Setting* $y^0 = y^1$. We already know that, in Corollary 3.3, we may put $s = t\beta$ where $\beta = Au - b$ and that, due to the lemma, we may split the sum (29) into

$$S_1 + S_2 = t^{-1} \sum_{i:\beta_i \neq 0} r_i Dg_i(x) + t^{-1} \sum_{i:\beta_i = 0} \gamma_i Dg_i(x).$$

For $\beta_i \neq 0$, it follows that $y_i^1 = y_i^0 = 0$ and, by definition of D_i, r_i is restricted only by $r_i s_i \geq 0$. Thus the terms S_1 attain exactly all values of $\sum_{i:\beta_i \neq 0} \alpha_i Dg_i(x)$ where $\alpha_i \beta_i \geq 0$. For $\beta_i = 0$, the terms S_2 form a subset of the linear hull $H = \operatorname{lin}\{Dg_i(x), \beta_i = 0\}$.

However, if we change the related coordinates y_i^0 by setting $y_i^0 = y_i^1$ and take into account that thereafter $\gamma_i = r_i$ is not restricted by sign (in accordance with D_i), these terms form the whole space H. Hence, if at all, condition (29) can be fulfilled with the particular setting $y^0 = y^1$ and means: Some sequence of $\alpha = \alpha^k$ satisfies

$$\sum_{i:\beta_i \neq 0, \alpha_i \beta_i \geq 0} \alpha_i Dg_i(x) + \sum_{i:\beta_i = 0} \alpha_i Dg_i(x) \to a - Q(y^1)u, \ \text{ where } x = x^k \to x^0,\ t = t_k \downarrow 0.$$

Clearly, the interplay of the sequences x^k and t_k is important only in the $C^{1,1}$ case. Since $y_i^1 \beta_i = 0$ follows from Lemma 3.4, this proves Theorem 3.1 and Remark 3.2.

4. Discussion of the singularity condition (22). In this section, we demonstrate that the sequences in conditions (22) and (23) of Theorem 3.1 are really necessary and that there is no singularity characterization by a finite derivative condition. Further, we point out the role of quadratic approximations for s.L.s. of stationary points. The counterexamples are constructed in such a way that the same facts become clear for the so-called Aubin property (cf. section 4.3) of X as well.

4.1. Comparison of strong Lipschitz stability of X and X_{KKT}. Let us first compare strong Lipschitz stability of X and the map X_{KKT} of KKT-points assigned to $p = (a, b)$. The following characterization is well known and holds without supposing any constraint qualification. We apply T_N as in (16).

THEOREM 4.1. *For C^2 problems* (1), (9)*, the map X_{KKT} is strongly Lipschitz stable at $(0, (x^0, y^0))$ if and only if*

$$(34) \qquad Au = \beta,\ Q(y^0)u + A^{\mathsf{T}}\alpha = 0,\ \text{and } (0, \alpha, \beta) \in T_N(y^0) \text{ imply } (u, \alpha, \beta) = 0.$$

For details and equivalent conditions, we refer to [15], [2, Chapter 2], and [18, Chapter 8]. Nevertheless, some bibliographical notes seem to be appropriate. For problem (1) under LICQ, Theorem 4.1 was shown in [36] and [20]. An alternative approach to this result via Clarke's inverse function theorem [3] was given in [13]. For (1) without supposing LICQ, the result is known from [19] with C^2 functions and [21, 22] with $C^{1,1}$ functions. In the $C^{1,1}$ case, again $Q(y^0)u$ must be replaced by $\phi \in TF_1(\cdot, y^0)(x^0)(u)$ as in Remark 3.2. However, the proof has to sharpen Clarke's result [3] by using injectivity of $TF(z^0)$ and required chain rules for computing TF. In this way, Theorem 4.1 is also verified for the $C^{1,1}$ problem (9). For variational inequalities as in [36]

$$(35) \qquad p \in h(x) + N_C(x), \quad p \text{ a parameter},\ C \text{ is a polyhedron},\ h \in C^1;$$

thus also for C^2 problems (1), the theorem again appeared in [5]. The proof method in [5] is based on characterizations of the Aubin property [29, 30], normal maps [38], and alternative criteria for s.L.s. via coherent orientation [43, 35] of PC^1 functions. As a new topic, it has been shown [5] that, for (35), the weaker Aubin property and s.L.s. of X_{KKT} coincide.

The approach via the Aubin property does not apply to $C^{1,1}$ problems. A counterexample for *unconstrained* optimization ($X = X_{\text{KKT}}$) and an alternative approach to investigating this coincidence can be found in [23]. There, it also can be seen why the conditions in [5] and [21] are equivalent. The latter studies have been continued in [16, 17, 18], without requiring that appearing derivatives are piecewise smooth as in [43, 35], and in several models of the complementarity theory.

Needless to say, in all of these papers, the relations to noncanonical perturbations also have been pointed out.

Condition (34) means, in fact, injectivity of $TF(x^0, y^0)$. So, a similar but weaker condition is suggested which is called *TF-injectivity w.r.t. u* (cf. [18]):

$$(36) \qquad \begin{array}{l} \forall y \in Y^0, \\ Au = \beta,\ Q(y)u + A^{\mathsf{T}}\alpha = 0, \text{ and } (0, \alpha, \beta) \in T_N(y) \text{ imply } u = 0. \end{array}$$

In section 5 we need condition (36) also with C_N at the place of T_N, which we call *CF-injectivity w.r.t. u.*

Since $\alpha_i\beta_i \geq 0$, both TF- and CF-injectivity w.r.t. u are satisfied if $\langle u, Q(y^0)u\rangle > 0\ \forall y^0 \in Y^0,\ u \neq 0$. Recalling the descriptions (17), (16) of C_N and T_N and supposing again that $g(x^0) = 0$, without loss of generality, both conditions can be formulated for problem (1) by means of the quadratic programs

$$(37) \qquad P(p,y): \min_u \{Df(x^0)u + \tfrac{1}{2}\langle u, Q(y)u\rangle - \langle a,u\rangle \mid Au \le b\},\ y \in Y^0,\ p = (a,b),$$

and local uniqueness of their stationary solutions; cf. Remark 4.3. Notice that (34) just means that X_{KKT}, if considered for $P(p,y^0)$, is s.L.s. at $(0,0)$. To describe s.L.s. of X, one has to study a family of canonically perturbed quadratic programs instead of (37) only:

$$(38) \qquad P(p,y,x): \min_u \{Df(x^0)u + \tfrac{1}{2}\langle u, Q(y)u\rangle - \langle a,u\rangle \mid Dg(x)u \le b\},\ p = (a,b).$$

For problem (9), let $K_{b,x}(u)$ be the normal cone to the constraints in (38) at u and replace (38) analogously by the (polyhedral w.r.t. u) inclusion

$$(39) \qquad P(p,y,x): a \in h(x^0) + Q(y)u + K_{b,x}(u).$$

THEOREM 4.2. *For C^2 problems* (1), (9) *under* (7), *nonsingularity is the same as the following:*

(40) *For each $y \in Y^0$, there exist positive L and ε such that $\|u\| \le L\,\|p' - p\|$ whenever 0 and $u \in \varepsilon B$ are stationary points of $P(p,y,x)$ and $P(p',y,x)$, respectively, for $x \in x^0 + \varepsilon B$ and p,$p' \in \varepsilon B$.*

Proof. Let singularity hold, and let u, y, x^k, α^k be according to the singularity characterization (25). Setting $\beta = Au$ and $c = Q(y)u + Dg(x^k)^{\mathsf T}\alpha^k$, we have $(c, x^k) \to (0, x^0)$ and $(0, \alpha^k, \beta) \in T_N(y)$. Let t_k with $0 < t_k < (k + ||\alpha^k||)^{-1}$ be small enough such that $y_i - t_k|\alpha_i^k| > 0$ if $y_i > 0$. We write $x = x^k$, $t = t_k$, $\alpha = \alpha^k$ and put $a = (Dg(x) - A)^{\mathsf T}y - tDg(x)^{\mathsf T}\alpha^-$, $b = t\beta^+$, $p = (a,b)$. Now we show that for some p', the points $(0,\ y - t\alpha^-)$ and $(tu,\ y + t\alpha^+)$ are KKT-points for $P(p,y,x)$ and $P(p',y,x)$, respectively, where $p \to 0$, $p' - p = o(t)$, $t \downarrow 0$, and $x \to x^0$.

Discussing separately the cases of $y_i = 0$ and $y_i > 0$ for $(0,\alpha,\beta) \in T_N(y)$ (16), one obtains

$$(y - t\alpha^-)_i > 0 \Rightarrow \beta_i \le 0 = \beta_i^+ \quad\text{and}\quad (y + t\alpha^+)_i > 0 \Rightarrow 0 \le \beta_i = \beta_i^+.$$

Since, in addition,

$$Df(x^0) + Dg(x)^{\mathsf T}(y - t\alpha^-) = (Df(x^0) + A^{\mathsf T}y) + (Dg(x) - A)^{\mathsf T}y - tDg(x)^{\mathsf T}\alpha^- = a,$$

one easily sees that the origin along with the dual $y - t\alpha^-$ is stationary for problem $P(p,y,x)$. Notice that $p \to 0$ is ensured by the choice of t.

Next let $a' = a + tc$, $b' = b + t(Dg(x) - A)u$, and $p' = (a', b')$. Then $p' - p = o(t)$ and

$$\begin{aligned} &Df(x^0) + tQ(y)u + Dg(x)^{\mathsf T}(y + t\alpha^+) \\ &\quad = Df(x^0) + Dg(x)^{\mathsf T}(y - t\alpha^-) + t[Dg(x)^{\mathsf T}\alpha + Q(y)u] = a + tc = a'. \end{aligned}$$

If $(y+t\alpha^+)_i > 0$, we have $tDg_i(x)u = tA_iu + t(Dg_i(x) - A_i)u = t\beta_i^+ + t(Dg_i(x) - A_i)u = b'_i$. Hence constraint i is active. Due to $Au = \beta \le \beta^+$ (hence $Dg_i(x)(tu) \le b'_i\ \forall i$), it follows that $(tu,\ y + t\alpha^+)$ is a KKT-pair for $P(p',y,x)$ and (40) cannot hold.

Conversely, assume that (40) is violated, and let 0 and tu, with $\|u\| = 1$ and $t \downarrow 0$, be stationary for $P(p,y,x)$ and $P(p',y,x)$, respectively, where $y \in Y^0$ is fixed and $p' - p = o(t), x \to x^0, p \to 0$. With assigned duals λ, λ', one obtains $Df(x^0) + Dg(x)^{\mathsf{T}}\lambda^+ = a$, $Df(x^0) + tQ(y)u + Dg(x)^{\mathsf{T}}\lambda'^+ = a'$ and

$$o_1(t) = a' - a = Dg(x)^{\mathsf{T}}(\lambda'^+ - \lambda^+) + tQ(y)u,\ o_2(t) = b' - b = \ tDg(x)u + t(\lambda'^- - \lambda^-).$$

After selecting a cluster point u^* of the related u, now the convergence

$$t^{-1}Dg(x)^{\mathsf{T}}(\lambda'^+ - \lambda^+) \ \to \ -Q(y)u^* \quad \text{and} \quad (\lambda'^- - \lambda^-) \ \to -Dg(x^0)u^*$$

follows. This is just the singular situation studied in Theorem 2.3. □

Remark 4.3. The same formal condition (40), but with $x = x^0$ and $p' = p$, describes TF-injectivity w.r.t. u, whereas (40), with $x = x^0$ and $p' = p = 0$, characterizes just CF-injectivity w.r.t. u; see, e.g., [18, sections 8.2 and 8.3].

Theorem 4.2 and Remark 4.3 extend Robinson's approach of quadratic approximation [36] for analyzing KKT-points in a canonical way to the case of stationary points and establish the bridge to complementarity problems and second order conditions. Further, one directly obtains invariance of singularity w.r.t. quadratic approximation of the objective since the problems $P(p, y, x)$ remain the same under this operation.

COROLLARY 4.4. *Suppose* (7). *For C^2 problems* (1) *singularity at* $(0, x^0)$ *is invariant w.r.t. replacing f by its quadratic approximation at x^0. For C^2 problems* (9)*, the same holds w.r.t. replacing h by its linearization at x^0.*

Theorem 3.1 and Remark 4.3 also ensure (due to s.L.s. ⇒ MFCQ) the following implication which was already shown for C^2 programs (1) in [18] by alternative means.

COROLLARY 4.5. *For C^2 problems* (1), (9)*, TF-injectivity w.r.t. u is necessary for strong Lipschitz stability.*

Corollary 5.2 will ensure sufficiency if the constraint functions are linear-affine.

Fixing $u = 0$, (34) implies that all (active) gradients $Dg_i(x^0)$ are linearly independent, i.e., LICQ at x^0. In this case, strong Lipschitz stability of the map X (being now s.L.s. of X_{KKT}) depends on the first two derivatives of the data functions at x^0. The same is valid for all stability statements which (as in section 5) can be reduced to the above injectivity conditions. Now we show that this generally cannot be expected for s.L.s. of X.

4.2. Basic counterexamples.

Example 4.6. Consider the following problem, given for parameter $p = (a, b) = 0$ and with some constant r:

$$\min\ rx_1^2 + x_2 \ \text{ s.t. } \ g_1(x) = -x_2 \le 0,\ g_2(x) = x_1^2 - x_2 \le 0.$$

Then $Df = (2rx_1, 1)$, $Dg_1 = (0, -1)$, $Dg_2 = (2x_1, -1)$, and $x^0 = (0, 0)$ is a stationary point with $Y^0 = \{y \ge 0 \mid y_1 + y_2 = 1\}$ and $A_1 = A_2 = (0, -1)$. With $\gamma = 2r + 2y_2$, we have

$$Q(y) = \begin{pmatrix} \gamma & 0 \\ 0 & 0 \end{pmatrix}.$$

Hence $u\,Q(y) = (\gamma u_1, 0)$.

Since at least one y_i is positive for $y \in Y^0$, it follows that $u \perp A_i\ \forall i$ from $y_i A_i u = 0\ \forall i$. Hence all u of interest have the form $u = (u_1, 0),\ u_1 \ne 0$. Condition (22) now requires exactly that for some sequence of $(\alpha_1, \alpha_2) \in R^2$ and of converging $x \to x^0$, it holds that

$$(\gamma u_1, 0) + \alpha_1(0, -1) + \alpha_2(2x_1, -1) \to 0.$$

This condition cannot be satisfied *with* $x = x^0$ whenever $\gamma \neq 0$. Note that $\gamma \neq 0$ holds for all $y \in Y^0$ if $r \notin [-1, 0]$, so convexity of the problem plays no role.

On the other hand, it suffices to put $x = (\frac{1}{k},\ 0)$, $\alpha_2 = -\frac{1}{2}\ k\ \gamma u_1$, and $\alpha_1 = -\alpha_2$ in order to satisfy the singularity condition.

So, given any r, the problem is not s.L.s. However, if $r > 0$, then

$$\text{for each } y \in Y^0,\ Q(y) \text{ is positive definite on } K(y) = \{u \mid u \perp A_i \ \text{ if } y_i > 0\}, \tag{41}$$

since $u_1 \neq 0\ \forall u \in K(y) \setminus \{0\}$ and $y \in Y^0$. The latter is Kojima's condition [20] for *strong stability* at $(0, x^0)$; see section 5.3.

Example 4.7. Change Example 4.6, with some fixed $q \geq 2$, as follows:

$$\min\ x_1^2 + x_2 \ \text{ s.t. }\ g_1(x) = -x_2 \leq 0,\ g_2(x) = x_1^{q+1} - x_2 \leq 0.$$

We again obtain *singularity* at $(0, 0)$, since for any $u = (u_1, 0) \neq 0$, it holds that

$$(2u_1, 0) + \alpha_1(0, -1) + \alpha_2((q+1)x_1^q, -1) \to 0$$

for the sequences $x = (\frac{1}{k}, 0)$, $\alpha_2 = -\frac{2u_1}{(q+1)x_1^q}$, and $\alpha_1 = -\alpha_2$.

For odd q, we are still in the class of convex problems having unique and continuous solutions $x(p)$ for all parameters $p = (a, b)$. Nevertheless there is no possibility of identifying the singularity by using only the first q derivatives of f and g at x^0, since these derivatives are the same for the next, s.L.s. example with $r = 1$.

Example 4.8. Change only the second constraint in Example 4.6,

$$\min\ rx_1^2 + x_2 \ \text{ s.t. }\ g_1(x) = -x_2 \leq 0,\ g_2(x) = -x_2 \leq 0.$$

Now the map X is *s.L.s.* at $(0, x^0)$ for every $r \neq 0$. If $r < 0$, then the stationary points are never minimizers. This is remarkable since Kojima's strong stability holds, under (47), only if x^0 is a minimizer satisfying (41); cf. section 5.3.

4.3. Comparison with the Aubin property of X.

For verifying s.L.s. of a mapping, one may show that the image sets are (at most) single-valued and the so-called Aubin property is satisfied. Recall the definition in [1].

DEFINITION 4.9. *A map $\Gamma : P \rightrightarrows X$ (normed spaces) is said to be* pseudo-Lipschitz *at $(0, z^0) \in \operatorname{gph}\Gamma$ if there are positive ε, δ, and L such that $\Gamma(p) \cap (z^0 + \varepsilon B_X) \ \subset\ \Gamma(p') + L\|p' - p\| B_X \ \ \forall p, p' \in \delta B_P$.*

Other notations (or equivalent notions) for the same fact are that Γ^{-1} is metrically regular, respectively, pseudoregular, and that Γ *has the Aubin property* [42]. This property has several consequences and applications in stability theory. In particular, it implies local solvability due to $z^0 \in \Gamma(0)$.

Recalling (4), MFCQ at x^0 is necessary for the Aubin property of X at $(0, x^0)$; less obvious is that if X_{KKT} is pseudo-Lipschitz at $(0, (x^0, y^0))$, then LICQ holds at x^0 (cf. [23] and the more general [18, Lemma 7.1]).

For characterizations of the Aubin property and explicit results concerning X_{KKT}, we refer to [1, 29, 30]. An explicit limit condition for $X = X(a, b)$ having this property can be found in [18, Theorem 8.42]. In our context, the following particular statement of this theorem is of interest.

COROLLARY 4.10. *For C^2 problems* (1) *under linear constraints and MFCQ, TF-injectivity w.r.t. u is equivalent to the Aubin property of X at* $(0, x^0)$.

Further, both of the regularity notions coincide for the mapping $X_{glob}(a, b)$ *of global minimizers* to problem (1). The following statement (implicitly known from [18]) holds quite generally.

THEOREM 4.11. *For the mapping X_{glob} to programs* (1), *even without any constraint qualification or continuity of the involved functions, s.L.s. and the Aubin property at* $(0, x^0)$ *coincide.*

Proof. Under the Aubin property, $X_{glob}(a, b)$ remains locally single-valued. This is a consequence of Corollary 4.7 in [18] which states, for any b, that $X_{glob}(\cdot, b)$ is only pseudo-Lipschitz at (a, x), $x \in X_{glob}(a, b)$, if $X_{glob}(a', b)$ is single-valued for all a' near a. □

Comment. By Corollary 4.7 [18], the same holds for $X_{glob}(\cdot, 0)$ (fixed constraints), and $X_{glob}(a, b)$ may even denote the global solutions s.t. $G(b) \cap S$ for any set $S \subset R^n$.

5. Specializations and related stability concepts. In this section, we restrict ourselves to the C^2 case and compare s.L.s. with known conditions for several related stability notions. We derive a specialization of the singularity condition to some case of linear-quadratic constraints and discuss local solvability near $(0, x^0)$. Further, we help to clarify the interrelations between s.L.s. and Kojima's stability and discuss the case of general perturbations.

Throughout this section, we use the abbreviations $A = Dg(x^0)$ and $Q(y) = D_x F_1(x^0, y)$ as in (21) and (without loss of generality) $g(x^0) = 0$. Further, the fact that (1) and (9) are C^2 problems is essential.

5.1. Singularity for linear and quadratic constraints. In this subsection, we additionally suppose that (7) is fulfilled.

If all constraint functions g_i are quadratic, then s.L.s. depends on the first two derivatives of the problem functions *at* x^0 *only*; cf. Theorem 4.2 and note that g is completely described by these derivatives. Nevertheless, we cannot present, at this moment, a verifiable algebraic condition in these terms.

For linear constraints, one may replace the sequence (α^k, x^k) by (α^0, x^0) in Theorem 3.1 to obtain a simpler condition (cf. Corollary 5.2 below). A similar condition (for fixed y, still equivalent to a linear complementarity system) can be obtained if only one constraint is quadratic and the others are linear (where, under presence of equations, it does not matter which constraint is quadratic). To show this, one mainly has to apply that, for any system of linear inequalities, the right-hand sides that permit solvability form a closed set (closed range argument). Given u, define the polyhedral cone $K_u = \{\alpha \mid \alpha_i A_i u \geq 0 \ \forall i\}$.

THEOREM 5.1. *Let* $g_1(x) = \gamma + c^{\mathsf T} x + \frac{1}{2} x^{\mathsf T} C x$ *(with given* γ, c, *and quadratic symmetric matrix* C*) and let all* g_i, $i > 1$, *be linear-affine for problem* (1) *or* (9)*. Then singularity holds true if and only if there exist* $u \neq 0$ *and* $y \in Y^0$ *with* $y_i A_i u = 0$ $\forall i$ *such that*

(i) $Q(y)u + A^{\mathsf T}\alpha = 0$ *for some* $\alpha \in K_u$ *or*

(ii) *there are* $z, \zeta \in K_u, w \in R^n$ *with* $Q(y)u + A^{\mathsf T} z + Cw = 0$, $A^{\mathsf T}\zeta = 0$, *and* $|\zeta_1| = 1$.

Proof. Given α^k and $x^k \to x^0$ according to the singularity condition of Theorem 3.1, take $Dg_1(x^k) = A_1^{\mathsf T} + C(x^k - x^0)$ into account, put $w^k = \alpha_1^k(x^k - x^0)$, and notice that

$$Dg(x^k)^{\mathsf T}\alpha^k = Cw^k + A^T\alpha^k = b^k, \ b^k \to -Q(y)u, \ \alpha^k \in K_u. \tag{42}$$

Case 1. If $Cw^k \to 0$ (for some subsequence), the closed range argument yields that the singularity condition holds in the form $A^T\alpha = -Q(y)u$, $\alpha \in K_u$, and (i) follows.

Case 2. Otherwise, $\lambda_k := |\alpha_1^k|$ diverges, and division in (42) shows that $A^T\alpha^k/\lambda_k \to 0$. Hence, again by the closed range argument, the equation $A^T\zeta = 0$ holds for some $\zeta \in K_u$ with $|\zeta_1| = 1$. So, (ii) is satisfied since (42) guarantees the solvability of

$$Cw + A^T z = -Q(y)u,\ z \in K_u.$$

It remains to show that (i) and (ii) imply singularity. Concerning (i) this is trivial; we assume (ii) and define $\alpha^k = k\zeta + z$ and $x^k = x^0 + t_k w$ with $k \to \infty$ and $t_k = (k\zeta_1 + z_1)^{-1}$. This yields

$$\alpha^k \in K_u \text{ and } Dg(x^k)^{\mathsf{T}}\alpha^k = A^{\mathsf{T}}z + (k\zeta_1 + z_1)\ t_k\ Cw = A^{\mathsf{T}}z + Cw = -Q(y)u.$$

Since now $x^k \to x^0$, the singularity condition is satisfied. □

Note that g_1 may describe a complementarity condition $\langle x^1, x^2\rangle = 0, x = (x^1, x^2) \in R^{2n}$ since (7) holds true. If all constraints are linear, then $Q(y)$ is constant and one obtains the following corollary.

Corollary 5.2. *Let all g_i be linear-affine for* (1) *or* (9)*. Then singularity holds true if and only if TF-injectivity w.r.t. u* (36) *is violated.*

The case of linear constraints does not coincide with the analysis of the polyhedral inclusion (35) due to variation of a and b and $X \neq X_{\text{KKT}}$.

For the remainder of the paper, hypothesis (7) is too weak and will be replaced by MFCQ.

5.2. The upper Lipschitz property and local solvability. As already mentioned, CF-injectivity w.r.t. u characterizes exactly the upper Lipschitz property of X for canonical perturbations $p = (a, b)$.

The upper Lipschitz condition becomes more complicated for the stationary point mapping $a \mapsto \hat{X}(a) := X(a, 0)$ (i.e., constraints are not perturbed). The next statement originally has been shown via the analysis of proto-derivatives for subgradients of composed maps; see [41, 31, 27, 42]. For deriving, as in [18, Corollary 9.10], the form (43) given in the terms introduced above, let $\|Df(x^0)\|$ be sufficiently small (otherwise multiply f with some small $\lambda > 0$) and suppose MFCQ:

(43)

$\hat{X}$ is not locally upper Lipschitz at $(0, x^0)$ iff CF-injectivity w.r.t. u can
be violated with some $\alpha \geq 0$ and $y^0 \in Y^0$ where, in addition, y^0 solves the LP
$\max_y u^{\mathsf{T}} D_x^2 L(x^0, y)u$ s.t. $D_x L(x^0, y) = 0$ and $y_i \geq 0$ if $A_i u = \alpha_i = 0$,
with optimal value 0.

Again, only the first and second derivatives of f and g at x^0 are crucial.

If x^0 is a local minimizer of P(0) satisfying MFCQ for a C^2 problem (1), then by [18, Theorem 8.36],

CF-injectivity w.r.t. u implies that X is upper Lipschitz stable
at $(0, x^0)$ and all $x \in X(p) \cap (x^0 + \varepsilon B)$ are local minimizers for
sufficiently small ε and $\|p\| < \delta(\varepsilon)$.

Further, if the singularity condition of Theorem 3.1 is not satisfied, then TF-injectivity w.r.t. u holds true. This ensures the weaker CF-injectivity and, consequently, we have the following corollary.

Corollary 5.3. *For programs* (1) *under MFCQ and a local minimizer* x^0 *of* $P(0)$*, the upper Lipschitz property (hence also nonsingularity) implies local solvability.*

Theorem 5.4. *For linearly constrained programs* (1) *under MFCQ,* TF*-injectivity w.r.t.* u (36) *coincides with the Aubin property and s.L.s. of* X *at* $(0, x^0)$*.*

Proof. By Corollary 5.2, the singularity condition of Theorem 3.1 is the negation of (36). Hence, under local solvability, the equivalence between s.L.s. of X and (36) is valid. On the other hand, (36) coincides with the Aubin property by Corollary 4.10. So local solvability is satisfied, too. □

By Theorem 4.11 and its comment, nonsingularity implies local solvability if the map of stationary points is replaced by the map X_{glob} of minimizers to problem (1) (w.r.t. some fixed neighborhood of x^0). Clearly, under the generality of Theorem 4.11, we cannot present any verifiable condition for nonsingularity. On the other hand, second order (and growth) conditions can be imposed to ensure that all stationary points near x^0 are local minimizers, see, e.g., [37, 9, 10, 2]. However, many conditions of this type directly imply the upper Lipschitz property; then local solvability follows from Corollary 5.3, too.

Recently [24, 25], the implication "nonsingularity ⇒ local solvability" has been studied for maps $\Gamma : R^n \rightrightarrows R^n$, based on the nonsingularity condition (12) only, i.e.,

$$\{0\} = T\Gamma(0, z^0)(0). \tag{44}$$

Though our map $X = X(a, b)$ acts between spaces having different dimension, let us give the main topics of this approach in view of our models:

(i) It is supposed (described by being *kernel inverting*) that, near $(0, z^0)$, Γ satisfies

$$\Gamma(p) = U^{-1}\sigma^{-1}(Vp) + Wp \tag{45}$$

with locally Lipschitzian $\sigma : R^n \to R^n$, regular linear transformations U, V of the image, and preimage space and linear W. This hypothesis yields that

$$(44) \Leftrightarrow \{0\} = T(\sigma^{-1})(0, Uz^0)(0) \Leftrightarrow \sigma^{-1} \text{ is s.L.s. at } (0, Uz^0).$$

Here, the equivalence on the right is just Theorem 1.1 in [21] and ensures via chain rules the *inverse mapping theorem* [24, 25]: Γ is s.L.s. at $(0, z^0) \Leftrightarrow$ (44) holds true.

(ii) Property (45) holds for mappings $\Gamma = H$ and $\Gamma = H^{-1}$ if H is *max-hypomonotone*, i.e., $H + \mu\,\mathrm{id}$ (id = identity) is maximal-monotone for some $\mu > 0$, locally around the (by μ transformed) reference point.

(iii) Under MFCQ at x^0 and $g \in C^2$, the map of normals to $G(0)$

$$H(x) = \{z \mid z = Dg(x)^{\mathsf{T}}y,\ \ y_i g_i(x) = 0,\ y \geq 0\} \text{ if } x \in G(0) \text{ else } H(x) = \emptyset \tag{46}$$

is max-hypomonotone at $(x^0, z^0) \in \operatorname{gph} H$.

For (ii) and (iii), basic results of [33] have been used. Applying (i), (ii), and (iii) to H^{-1}, one may now conclude that nonsingularity of the mapping $X(\cdot, 0)$ (fixed constraints) implies local solvability, too. However, one does not know whether $X(0, b)$ is locally empty or not, and checking (44) in original terms remained open if Γ differs from the solution map $\Gamma = \Gamma(p)$ of (35).

5.3. Strong stability in Kojima's sense. For a C^2 problem (1), Kojima's [20] notion of strong stability requires that, w.r.t. small C^2 perturbations of the objective and small C^2 perturbations of the constraints, the map X of stationary points is locally unique and *continuous* near $(0, x^0)$. *Under LICQ*, the equivalence of s.L.s. and Kojima's strong stability for X is a standard fact; see, e.g., [19]. Thus let us consider the remaining case of

$$x^0 \in X(0) \ \text{ satisfies MFCQ, but not LICQ.} \tag{47}$$

For this situation, we can combine Theorem 5.4 with the following well-known statement of Kojima [20, Theorem 7.2]:

> Under (47), strong stability in Kojima's sense holds true iff
> condition (41) is satisfied; we recall: $Q(y)$ is positive definite
> on $K(y) = \{u \mid u \perp A_i \ \text{ if } y_i > 0\} \ \forall y \in Y^0$.

Obviously, this implies that x^0 is a local minimizer for (1) at $p = 0$ and, in addition, that one cannot find $y \in Y^0$ and (u, α) with $u \neq 0$ such that

$$Q(y)u + A^{\mathsf{T}}\alpha = 0, \quad y_i A_i u = 0, \ \text{and } \alpha_i A_i u \geq 0 \ (\forall i)$$

(otherwise multiply the equation $Q(y)u + A^{\mathsf{T}}\alpha = 0$ with u). The condition just given is once more TF-injectivity w.r.t. u. So we arrive at the following interrelations which seem to be new.

COROLLARY 5.5. *Let $x^0 \in X(0)$ satisfy* (47).

(i) *Under linear constraints, Kojima's strong stability implies s.L.s.*

(ii) *If x^0 is a local minimizer for $p = 0$, then s.L.s. implies Kojima's strong stability.*

Proof. Statement (i) now follows from Theorem 5.4, and statement (ii) follows from [18, Corollary 8.37], which just asserts that Kojima's condition is satisfied. □

Due to Example 4.8, statement (i) cannot be reversed. Due to Example 4.6, Kojima's strong stability does not imply s.L.s. for convex problems even if $X = X(\cdot, 0)$ is s.L.s.

5.4. Consequences for nonlinear perturbations. Let $(z, p) \mapsto \widetilde{X}(z, p)$ be the stationary point map of the parametric program

$$\text{(P)}(z,p): \quad \min_x \ f(x,z) - a^{\mathsf{T}}x \ \text{ s.t. } \ g_i(x,z) \leq b_i \, (i = 1, \ldots, m), \, p = (a, b), \tag{48}$$

and suppose that $f, g_i \in C^2(R^{n+\kappa}, R)$ and $D^2_{xz}f$ and $D^2_{xz}g_i$ are locally Lipschitz.

Then it is well known that strong stability in Kojima's sense and the upper Lipschitz property of the stationary solution set map X under canonical perturbations carry over to the mapping $\widetilde{X}$. On the other hand, Robinson [37, p. 219] gave an example, namely, to minimize $\sum_{i=1}^m x_i^2$ under nonlinearly (by z) perturbed linear constraints satisfying MFCQ, with two important properties: Under canonical perturbations of the problem to $z = 0$, X is s.L.s. at $(0, x^0)$. In contrast, $\widetilde{X}$ is not s.L.s. So there is a gap between canonical and nonlinear perturbations.

Below we explain this gap in the context of a parametric variational inequality. Our construction is standard in linking an inverse function theorem with an implicit function theorem. Under MFCQ and suitable second order conditions, it also has been successfully used w.r.t. upper Lipschitz behavior, Kojima's strong stability, and calculation of $C\widetilde{X}$ in many papers, see, e.g., [20, 37, 42, 2, 17, 18]. However, if g depends on z, s.L.s. of $\widetilde{X}$ requires additional assumptions such as a constant rank constraint qualification [34] even if s.L.s. holds for canonical perturbations.

Consider, again under MFCQ, the problem

$$\text{Find } x \in G_0(z)\text{: } \ 0 \in h(x,z) + N_{G_0(z)}(x), \qquad z \text{ a parameter,}$$

where $G_0(z) = \{x | g(x,z) \le 0\}$, $g \in C^2(R^{n+\kappa}, R^m)$, and $h \in C^1(R^{n+\kappa}, R^n)$ with locally Lipschitz $D_{xz}g$, $D_z h$, and z varies near $0 \in R^\kappa$. Denote here by $X(z)$ the problem's solution set at given z and by $Y(z,x)$ the associated multiplier set. Due to existence and continuity of the related derivatives, we have

$$\begin{aligned} h(x,z) &= h(x,0) + D_z h(x,0)z + o_1(x,z),\\ g(x,z) &= g(x,0) + D_z g(x,0)z + o_2(x,z),\\ D_x g(x,z) &= D_x g(x,0) + D_{zx} g(x,0)z + o_3(x,z),\end{aligned}$$

and obtain with $x \in X(z)$, $y \in Y(z,x)$ from $h(x,z) + y^{+T} D_x g(x,z) = 0$ and $g(x,z) = y^-$ that

$$h(x,0) + D_z h(x,0)z + o_1 + y^{+T}(D_x g(x,0) + D_{zx} g(x,0)z + o_3) = 0,$$

$$g(x,0) + D_z g(x,0)z + o_2 = y^-.$$

Setting

$$a = D_z h(x,0)z + o_1 + y^{+T}(D_{zx} g(x,0)z + o_3) \ \text{ and } \ b = -(D_z g(x,0)z + o_2), \tag{49}$$

one sees that x is nothing but a stationary point for the canonically perturbed problem

$$a \in h(x,0) + N_{G(b)}(x), \quad \text{where } G(b) \text{ is given by } g(x,0) \le b.$$

Here, a depends on y if g depends on z. For (z,x) near $(0,x^0)$, the duals y are bounded (MFCQ) and close to $Y^0 = Y(0,x^0)$. So the linear maps in (49), $D_z h(x,0)$, $y^{+T} D_{zx} g(x,0)$, and $D_z g(x,0)$, are uniformly Lipschitz functions (rank L') of z. The same holds true for the functions o_k above. This ensures, for $x \in X(z)$ near x^0, a (pointwise) Lipschitz estimate

$$\|x - x^0\| \le L(\|a\| + \|b\|) \le LL'\|z\|.$$

However, for a second pair of points $x' \in X(z')$, $y' \in Y(z',x')$, the crucial terms

$$H := y^{+T} D_{zx} g(x,0) \ \text{ and } \ H' := y'^{+T} D_{zx} g(x',0) \tag{50}$$

of (49) are (in general) only close to each other if there are multipliers y, y' such that $\|y' - y\|$ is small enough or if $(\text{lin } Y^0)^T D_{zx} g(x^0,0) = 0$. Only in these cases may one estimate, with certain constants $0 < c < c'$ and $W = H - H' \to 0$,

$$\begin{aligned} ||a - a'|| &\le \ c \,||z - z'|| + ||Hz - H'z'||\\ &= c\,||z - z'|| + ||H(z - z') + (H - H')z')|| \le \ c'\ ||z - z'|| + ||W||\ ||z'||,\end{aligned}$$

which guarantees, with new L' and due to s.L.s. for canonical perturbations,

$$\|x' - x\| \le \ L(\|a' - a\| + \|b' - b\|) \le \ L'\|z' - z\| + L'\ ||W||\ ||z'||.$$

Under the *strict MFCQ* which just ensures that Y^0 is single-valued, one obtains that $||W|| \le L'' \max\{||z||, ||z'||\}$. Under the *constant rank condition*, one finds $y \in Y(z,x), y' \in Y(z',x')$ such that $||y' - y|| \le L''||z' - z||$. The latter yields the desired Lipschitz estimate, indeed. Under MFCQ alone, the terms (50) are out of control.

6. Concluding remarks. The main results of this paper (cf. Theorems 3.1 and 4.2) indicate that (and how) s.L.s. of the stationary point mapping X for problem (1) or (9) depends on limits of data at a sequence $x^k \to x^0$. Some counterexamples show that our condition is not reducible to a condition which uses derivatives of the involved functions up to a fixed order *at the reference points* x^0 *only.* Further, a standard invariance principle fails to hold: Strong Lipschitz stability of X is not invariant w.r.t. replacing the problem functions by quadratic approximations at x^0.

Nevertheless, s.L.s. depends on the stationary solutions of canonically perturbed quadratic problems which, however, have different constraint-matrices $A(x) = Dg(x)$ for x near x^0 in general.

These properties of the studied stability notion are in contrast to s.L.s. of the primal-dual solution mapping X_{KKT} and of other popular stability properties (cf. sections 4.1 and 5), where—w.r.t. some multiplier associated with x^0—a fixed quadratic problem has to be analyzed.

Since under MFCQ the multiplier set is not a singleton, it is difficult to computationally verify the stability criteria already if the mentioned invariance principle holds true. Our analysis has shown that this is even more difficult in the case of s.L.s. of X and that this cannot be avoided. However, even for obtaining this negative result, we had to show how the abstract characterization of Theorem 2.3 via (seemingly hopeless) double sequences of multipliers can be transformed into the simpler analytical criterion of Theorem 3.1. Only the latter permitted the comparison with other stability notions and (more understandable) reformulations via quadratic problems.

For future research, simplifications of the conditions for particular problems such as in the case of linear-quadratic constraints (see section 5.1), as well as the study of generic classes of s.L.s. problems, seem to be most useful.

Acknowledgments. We are indebted to the referees for their detailed and constructive comments. Also, we would like to thank Stefan Scholtes for his suggestions and editorial help.

REFERENCES

[1] J.-P. Aubin and I. Ekeland, *Applied Nonlinear Analysis*, Wiley, New York, 1984.

[2] J. F. Bonnans and A. Shapiro, *Perturbation Analysis of Optimization Problems*, Springer-Verlag, New York, 2000.

[3] F. H. Clarke, *On the inverse function theorem*, Pacific J. Math., 64 (1976), pp. 97–102.

[4] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Wiley, New York, 1983.

[5] A. L. Dontchev and R. T. Rockafellar, *Characterizations of strong regularity for variational inequalities over polyhedral convex sets*, SIAM J. Optim., 6 (1996), pp. 1087–1105.

[6] A. Dontchev and R. T. Rockafellar, *Characterizations of Lipschitz stability in nonlinear programming*, in Mathematical Programming with Data Perturbations, A. V. Fiacco, ed., Marcel Dekker, New York, 1998, pp. 65–82.

[7] A. V. Fiacco, *Introduction to Sensitivity and Stability Analysis*, Academic Press, New York, 1983.

[8] P. Fusek, D. Klatte, and B. Kummer, *Examples and counterexamples in Lipschitz analysis*, Control Cybernet., 31 (2002), pp. 471–492.

[9] H. Gfrerer, *Hölder continuity of solutions of perturbed optimization problems under Mangasarian-Fromovitz constraint qualification*, in Parametric Optimization and Related Topics, Math. Res. 35, J. Guddat et al., eds., Akademie-Verlag, Berlin, 1987, pp. 113–124.

[10] A. Ioffe, *On sensitivity analysis of nonlinear programs in Banach spaces: The approach via composite unconstraint optimization*, SIAM J. Optim., 4 (1994), pp. 1–43.

[11] H. Th. Jongen, P. Jonker, and F. Twilt, *Nonlinear Optimization in* R^n, I: *Morse Theory, Chebyshev Approximation*, Methoden Verfahren Math. Phys. 29, Verlag Peter D. Lang, Frankfurt am Main, Germany, 1983.

[12] H. Th. Jongen, P. Jonker, and F. Twilt, *Nonlinear Optimization in R^n*, II: *Transversality, Flows, Parametric Aspects*, Methoden Verfahren Math. Phys. 32, Verlag Peter D. Lang, Frankfurt am Main, Germany, 1983.

[13] H. Th. Jongen, D. Klatte, and K. Tammer, *Implicit functions and sensitivity of stationary points*, Math. Programming, 49 (1990), pp. 123–138.

[14] D. Klatte, *Nonlinear optimization under data perturbations*, in Modern Methods of Optimization, W. Krabs and J. Zowe, eds., Springer-Verlag, New York, 1992, pp. 204–235.

[15] D. Klatte and B. Kummer, *Strong stability in nonlinear programming revisited*, J. Austral. Math. Soc. Ser. B, 40 (1999), pp. 336–352.

[16] D. Klatte and B. Kummer, *Generalized Kojima functions and Lipschitz stability of critical points*, Comput. Optim. Appl., 13 (1999), pp. 61–85.

[17] D. Klatte and B. Kummer, *Contingent derivatives of implicit (multi-) functions and stationary points*, Ann. Oper. Res., 101 (2001), pp. 313–331.

[18] D. Klatte and B. Kummer, *Nonsmooth Equations in Optimization—Regularity, Calculus, Methods and Applications*, Nonconvex Optimization and Its Applications 60, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.

[19] D. Klatte and K. Tammer, *Strong stability of stationary solutions and Karush-Kuhn-Tucker points in nonlinear optimization*, Ann. Oper. Res., 27 (1990), pp. 285–307.

[20] M. Kojima, *Strongly stable stationary solutions in nonlinear programs*, in Analysis and Computation of Fixed Points, S.M. Robinson, ed., Academic Press, New York, 1980, pp. 93–138.

[21] B. Kummer, *Lipschitzian inverse functions, directional derivatives and application in $C^{1,1}$ optimization*, J. Optim. Theory Appl., 70 (1991), pp. 559–580.

[22] B. Kummer, *An implicit function theorem for $C^{0,1}$-equations and parametric $C^{1,1}$-optimization*, J. Math. Anal. Appl., 158 (1991), pp. 35–46.

[23] B. Kummer, *Lipschitzian and pseudo-Lipschitzian inverse functions and applications to nonlinear programming*, in Mathematical Programming with Data Perturbations, A. V. Fiacco, ed., Marcel Dekker, New York, 1998, pp. 201–222.

[24] A. B. Levy, *Solution sensitivity from general principles*, SIAM J. Control Optim., 40 (2001), pp. 1–38.

[25] A. B. Levy, *Lipschitzian multifunctions and a Lipschitzian inverse mapping theorem*, Math. Oper. Res., 26 (2001), pp. 105–118.

[26] A. B. Levy, R. A. Poliquin and R. T. Rockafellar, *Stability of locally optimal solutions*, SIAM J. Optim., 10 (2000), pp. 580–604.

[27] A. B. Levy and R. T. Rockafellar, *Sensitivity of solutions in nonlinear programs with nonunique multipliers*, in Recent Advances in Nonsmooth Optimization, D.-Z. Du, L. Qi, and R. S. Womersley, eds., World Scientific Press, Singapore, 1995, pp. 215–223.

[28] A. B. Levy and R. T. Rockafellar, *Variational conditions and the proto-differentiation of partial subgradient mappings*, Nonlinear Anal., 26 (1996), pp. 1951–1964.

[29] B. S. Mordukhovich, *Complete characterization of openness, metric regularity and Lipschitzian properties of multifunctions*, Trans. Amer. Math. Soc., 340 (1993), pp. 1–35.

[30] B. S. Mordukhovich, *Stability theory for parametric generalized equations and variational inequalities via nonsmooth analysis*, Trans. Amer. Math. Soc., 343 (1994), pp. 609–657.

[31] R. A. Poliquin, *Proto-differentiation of subgradient set-valued mappings*, Canad. J. Math, 42 (1990), pp. 520–532.

[32] R. A. Poliquin and R. T. Rockafellar, *Proto-derivative formulas for basic subgradient mappings in mathematical programming*, Set-Valued Anal., 2 (1994), pp. 275–290.

[33] R. A. Poliquin and R. T. Rockafellar, *Prox-regular functions in variational analysis*, Trans. Amer. Math. Soc., 348 (1995), pp. 1805–1838.

[34] D. Ralph and S. Dempe, *Directional derivatives of the solution of a parametric nonlinear program*, Math. Programming, 70 (1995), pp. 159–172.

[35] D. Ralph and S. Scholtes, *Sensitivity analysis of composite piecewise smooth equations*, Math. Programming, 76 (1997), pp. 593–612.

[36] S. M. Robinson, *Strongly regular generalized equations*, Math. Oper. Res., 5 (1980), pp. 43–62.

[37] S. M. Robinson, *Generalized equations and their solutions.* II: *Applications to nonlinear programming*, Math. Programming Stud., 19 (1982), pp. 200–221.

[38] S. M. Robinson, *Normal maps induced by linear transformations*, Math. Oper. Res., 17 (1992), pp. 691–714.

[39] S. M. Robinson, *Constraint nondegeneracy in variational analysis*, Math. Oper. Res., 28 (2003), pp. 201–232.

[40] S. M. Robinson, *Variational conditions with smooth constraints: Structure and analysis*, Math. Program., 97 (2003), pp. 245–265.

[41] R. T. Rockafellar, *First and second order epi-differentiability in nonlinear programming*, Trans. Amer. Math. Soc., 207 (1988), pp. 75–108.

[42] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, Springer-Verlag, Berlin, 1998.

[43] S. Scholtes, *Introduction to Piecewise Differentiable Equations*, Preprint 53, Institut für Statistik und Mathematische Wirtschaftstheorie, Universität Karlsruhe, Karlsruhe, Germany, 1994.

[44] L. Thibault, *Subdifferentials of compactly Lipschitzian vector-valued functions*, Ann. Mat. Pura Appl. (4), 125 (1980), pp. 157–192.

[45] L. Thibault, *On generalized differentials and subdifferentials of Lipschitz vector-valued functions*, Nonlinear Anal., 6 (1982), pp. 1037–1053.

SIAM J. Optim.
Vol. 16, No. 1, pp. 120–145

GLOBAL CONVERGENCE OF AN ELASTIC MODE APPROACH FOR A CLASS OF MATHEMATICAL PROGRAMS WITH COMPLEMENTARITY CONSTRAINTS[*]

MIHAI ANITESCU[†]

Abstract. We prove that any accumulation point of an elastic mode approach, that approximately solves the relaxed subproblems, is a C-stationary point of the problem of optimizing a parametric mixed P variational inequality. If, in addition, the accumulation point satisfies the MPCC-LICQ constraint qualification, and if the solutions of the subproblem satisfy approximate second-order sufficient conditions, then the limiting point is an M-stationary point. Moreover, if the accumulation point satisfies the upper-level strict complementarity condition, the accumulation point will be a strongly stationary point. If we assume that the penalty function associated with the feasible set of the mathematical program with complementarity constraints has bounded level sets, and if the objective function is bounded below, we show that the algorithm will produce bounded iterates and will therefore have at least one accumulation point. We prove that the obstacle problem satisfies our assumptions for both a rigid and a deformable obstacle. The theoretical conclusions are validated by several numerical examples.

Key words. MPCC, global convergence, complementarity constraints, nonlinear programming, obstacle problem

AMS subject classifications. 65K10, 90C33

DOI. 10.1137/040606855

1. Introduction.

Complementarity constraints are used to model numerous economics and engineering applications [18, 22]. Solving optimization problems with complementarity constraints may prove difficult in classical nonlinear optimization; however, given that, at a solution x^* such problems cannot satisfy a constraint qualification [18]. Nevertheless, recently there has been substantial interest in solving mathematical programs with complementarity constraints (MPCCs) by using classical nonlinear programming techniques. It has been shown that an SQP elastic mode approach can be expected to locally solve the generic case of MPCC [1]. Several positive results in the same direction have been proved for FilterSQP [10]. These results have been validated by an extensive numerical investigation of SNOPT and of FilterSQP [9]. SNOPT is an algorithm that implements a version of the elastic mode considered here [15]. The success of the SQP elastic mode approach has also been empirically extended to interior point approaches coupled with a relaxation strategy much like the elastic mode approach [2].

Classical nonlinear programming techniques are not the only ones developed for solving MPCC. Other types of techniques have been developed, most notably bundle nonsmooth trust region methods for implicit programming [22] and disjunctive

[*] Received by the editors April 16, 2004; accepted for publication (in revised form) December 7, 2004; published electronically September 8, 2005. This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under contract W-31-109-ENG-38. This work was performed by an employee of the U.S. Government or under U.S. Government contract. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.

http://www.siam.org/journals/siopt/16-1/60685.html

[†] Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439 (anitescu@mcs.anl.gov).

programming [18]. We believe that further investigation into the behavior of classical nonlinear programming techniques is warranted, however, given their success in solving large-scale problems, as well as the level of maturity of their software implementations.

Most of the convergence results presented so far in the literature are of a local nature [1, 10, 24]. In this work, we investigate the global convergence of an elastic-mode approach for a special class of MPCCs: optimization of parametric variational inequalities that satisfy the mixed P property [18, Def. 6.1.4]. To accommodate the fact that the penalty parameter may need to be driven to infinity, and to avoid the possibility that an insufficiently penalized relaxation will require an infinite number of steps to solve, we consider the possibility that a subproblem is solved only inexactly for a given penalty parameter.

Some of our techniques are related to those of [18, sect. 6.1]. That reference was the original inspiration for the class of problems considered here. A major difficulty with that work, however, is that unless lower-level strict complementarity is satisfied at the point toward which the algorithm defined in [18, sect. 6.1] converges, nothing can be said about the quality of that point [18, p. 312]. In this work we refine the range of outcomes that was provided in [18]. We show that under assumptions similar to those of [18, sect. 6], the outcome of an elastic-mode approach can be connected to weaker, though broadly employed in the literature [16, 23, 24], stationarity concepts, without requiring lower-level strict complementarity.

We note that global convergence to a B-stationary point was proved for an active set method that assumes that the MPCC linear independence constraint qualification (MPCC-LICQ) holds everywhere [14]. The assumptions of this work neither imply nor are implied from the assumptions of [14]. In particular, we do not assume uniform MPCC-LICQ, but we do assume that the problem has a structure that is more restrictive than that of [14].

2. Accumulation points are C-stationary points.

In this section, we discuss the mixed P property, state our model problem, and present and prove our global convergence result.

2.1. The mixed P property.

The key notion used in this section is the *mixed P partition* [18, Def. 6.1.4].

DEFINITION (mixed P partition). *Let $n_c \geq 1$ and $l \geq 0$. Let $A \in \mathbb{R}^{(n_c+l)\times n_c}$, $B \in \mathbb{R}^{(n_c+l)\times n_c}$, and $C \in \mathbb{R}^{(n_c+l)\times l}$. We say that the partition $[A\ B\ C]$ is mixed P partition if*

$$0 \neq (y,w,z) \in \mathbb{R}^{2n_c+l},\ Ay + Bw + Cz = 0 \Rightarrow \exists i,\ 1 \leqslant i \leqslant n_c,\ \textit{such that}\ y_i w_i > 0. \tag{2.1}$$

When $l = 0$, the C block is empty, and $[A\ B]$ is called a *P partition*. When defining our MPCC, the variables y_i and w_i will be required to be complementary to each other for $i = 1, 2, \ldots, n_c$. Therefore, n_c will denote the number of complementarity pairs, and thus its subscript.

LEMMA 2.1. *Assume $[A\ B\ C]$ is a mixed P partition. Let $D \in \mathbb{R}^{n_c \times n_c}$ be a diagonal matrix such that all its diagonal entries satisfy $d_i \neq 0$, $i = 1, 2, \ldots, n_c$. Then $[AD\ BD\ C]$ is also a mixed P partition.*

Proof. Let $0 \neq (y,w,z) \in \mathbb{R}^{2n_c+l}$ such that $ADy + BDw + Cz = 0$. Let $\tilde{y} = Dy$ and $\tilde{w} = Dw$. We then have that $0 \neq (\tilde{y}, \tilde{w}, z)$ and $A\tilde{y} + B\tilde{w} + Cz = 0$. From (2.1) we obtain that $\exists i$, $1 \leqslant i \leqslant n_c$, such that $0 < \tilde{y}_i\, \tilde{w}_i = d_i^2 y_i w_i$, which in turns implies that $y_i w_i > 0$. The proof is complete. □

Theorem 2.2. *Assume that $[A\ B\ C]$ is a mixed P partition. The system of linear constraints*

$$A^T\theta \leqslant 0, \quad B^T\theta \leqslant 0, \quad C^T\theta = 0$$

has the unique feasible point $\theta = 0$.

Proof. Let $0 \neq y \in R^{n_c}$. An immediate consequence of the fact that $[A\ B\ C]$ is a mixed P partition is that the matrix $[B\ C]$ is invertible [18]. We define $w \in \mathbb{R}^{n_c}$ and $z \in \mathbb{R}^l$ by

$$\begin{array}{rcl} w & = & -[I_{n_c}\ 0][B\ C]^{-1}Ay, \\ z & = & -[0\ I_l][B\ C]^{-1}Ay. \end{array}$$

Here we denote by I_k the $k \times k$ identity block. One can immediately see that (y, w, z) satisfies $Ay + Bw + Cz = 0$. Using that $[A\ B\ C]$ is a mixed P partition, we obtain that $\exists i$, $1 \leqslant i \leqslant n_c$, such that $y_i w_i > 0$. Let $Q = -[I_{n_c}\ 0][B\ C]^{-1}A$. Since $w = Qy$, this means that $\forall y \neq 0$, $\exists i$, $1 \leqslant i \leqslant n_c$, such that $y_i\,(Qy)_i > 0$, and thus Q is a P matrix [6]. Therefore, Q^T is also a P matrix [6], where

$$Q^T = -A^T \begin{bmatrix} B^T \\ C^T \end{bmatrix}^{-1} \begin{bmatrix} I_{n_c} \\ 0 \end{bmatrix}.$$

Now let θ be a feasible point of the linear constraints in the statement of the theorem. There exist $\eta_1, \eta_2 \in \mathbb{R}^{n_c}$, $\eta_1 \geqslant 0$, $\eta_2 \geqslant 0$, such that

$$A^T\theta + \eta_1 = 0, \quad B^T\theta + \eta_2 = 0, \quad C^T\theta = 0.$$

We solve for θ from the last two equations to obtain that

$$\theta = -\begin{bmatrix} B^T \\ C^T \end{bmatrix}^{-1} \begin{bmatrix} I_{n_c} \\ 0 \end{bmatrix} \eta_2. \tag{2.2}$$

Substituting in the remaining equation, we get that

$$0 = \eta_1 - A^T \begin{bmatrix} B^T \\ C^T \end{bmatrix}^{-1} \begin{bmatrix} I_{n_c} \\ 0 \end{bmatrix} \eta_2,$$

which, using our definitions for Q and Q^T, can be rewritten as

$$-\eta_1 = Q^T\eta_2.$$

From the definition of a P matrix, it follows that, if $\eta_2 \neq 0$, there exists i, where $1 \leqslant i \leqslant n_c$ such that $-\eta_{1,i}\eta_{2,i} > 0$ or $\eta_{1,i}\eta_{2,i} < 0$. This would contradict the fact that $\eta_1 \geqslant 0$ and $\eta_2 \geqslant 0$. The only alternative remaining is $\eta_1 = \eta_2 = 0$. From (2.2) this results in $\theta = 0$, which proves our claim. □

2.2. Optimization of parameterized mixed P variational inequalities. We now define the following MPCC together with its relaxed version:

$$\begin{array}{rlll} & \text{(OMPV)} & & \\ \min\limits_{x,y,w,z} & f(x,y,w,z) & & \\ \text{s.t.} & g(x) & \leqslant 0 & (\mu) \\ & h(x) & = 0 & (\lambda) \\ & F(x,y,w,z) & = 0 & (\theta) \\ & y,w & \leqslant 0 (\eta_{y,w}) & \\ & y^T w & \leqslant 0 & (\alpha_c) \end{array}$$

$$\begin{array}{rlll} & \text{(OMPV(c))} & & \\ \min\limits_{x,y,w,z,\zeta_1,\zeta_2} & f(x,y,w,z)+ & c(\zeta_1+\zeta_2) & \\ \text{s.t.} & g(x) & \leqslant 0 & (\mu) \\ & h(x) & = 0 & (\lambda) \\ -\zeta_1 e_{n_c+l} \leqslant & F(x,y,w,z) & \leqslant \zeta_1 e_{n_c+l} & (\theta^{-,+}) \\ & y,w & \leqslant 0 & (\eta_{y,w}) \\ & y^T w & \leqslant \zeta_2 & (\alpha_c) \\ & \zeta_1,\zeta_2 & \geqslant 0 & (\alpha_{1,2}) \end{array}$$

Here we show in parentheses the symbols we will use for the Lagrange multipliers. We denote by e_{n_c+l} a vector of all ones of dimension n_c+l.

The last constraint, which together with the bound constraints $y, w \leqslant 0$ forms the complementarity constraints of (OMPV), can be formulated as either an equality or an inequality constraint without altering the feasible set [10]. This constraint makes the problem a member of the class of MPCCs. Many of the issues and properties we will discuss are relevant to MPCCs and we will use the MPCC acronym to identify them. We note that there exists a more general class of related problems called mathematical programs with equilibrium constraints (MPEC) [18]. However, when the convex set over which the equilibrium constraints are defined can be represented by a finite number of inequalities, which is the case in most applications, an MPEC problem can be reduced to an MPCC problem.

Here $x \in \mathbb{R}^n$, $y, w \in \mathbb{R}^{n_c}$, $z \in \mathbb{R}^l$, $f : \mathbb{R}^{n+2n_c+l} \rightarrow \mathbb{R}$, $h : \mathbb{R}^n \rightarrow \mathbb{R}^{n_e}$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^{n_i}$, $F : \mathbb{R}^{n+2n_c+l} \rightarrow \mathbb{R}^{n_c+l}$. In (OMPV(c)) we relax the complementarity constraints $y^T w \leqslant 0$ as well as the nonlinear equation $F(x,y,w,z) = 0$. A large variety of other relaxations, connected to various nondifferentiable exact penalty functions, lead to similar results. For example, all nonlinear constraints can be relaxed [1]. We do not pursue that avenue further.

2.2.1. Stationary points of (OMPV). The difficult nature of MPCC has led to the definition of stationary points of (OMPV) different from those that correspond to the nonlinear programming interpretation of (OMPV). Formally, we consider multipliers of (OMPV) that satisfy $\alpha_c = 0$, though we relax the requirement that $\eta_y, \eta_w \geqslant 0$, and we denote the new multipliers by $\widehat{\eta}_y, \widehat{\eta}_w$. We call such multipliers $(\lambda, \mu, \theta, \widehat{\eta}_y, \widehat{\eta}_w)$ MPCC multipliers. The corresponding stationary points (x, y, w, z), together with the MPCC multipliers, satisfy the following relations:

$$\begin{array}{rcl} \nabla_x f(x,y,w,z)^T + \nabla_x h(x)^T \lambda & & \\ + \nabla_x g(x)^T \mu + \nabla_x F(x,y,w,z)^T \theta & = & 0, \\ \nabla_y f(x,y,w,z)^T + \widehat{\eta}_y + \nabla_y F(x,y,w,z)^T \theta & = & 0, \\ \nabla_w f(x,y,w,z)^T + \widehat{\eta}_w + \nabla_w F(x,y,w,z)^T \theta & = & 0, \\ \nabla_z f(x,y,w,z)^T + \nabla_z F(x,y,w,z)^T \theta & = & 0, \\ g(x) \leqslant 0, \mu \geqslant 0, h(x) = 0, g(x)^T \mu = 0, & & \\ F(x,y,w,z) = 0, y \leqslant 0, \ w \leqslant 0, \ y^T w = 0, & & \\ \sum_{k=1}^{n_c} y_k |\widehat{\eta}_{y,k}| = 0, \ \sum_{k=1}^{n_c} w_k |\widehat{\eta}_{w,k}| = 0. & & \end{array} \tag{2.3}$$

We distinguish the following types of stationarity [16, 23]:

- Weakly stationary points, where no sign requirements are made on $\widehat{\eta}_y, \widehat{\eta}_w$.
- C-stationary points, where we require that $\widehat{\eta}_{y,k} \widehat{\eta}_{w,k} \geqslant 0, \ k = 1, 2, \ldots, n_c$.

- M-stationary points, which are C-stationary points with the additional requirement that either $\widehat{\eta}_{y,k} \geqslant 0$ or $\widehat{\eta}_{w,k} \geqslant 0,\ k = 1, 2, \ldots, n_c$.
- B-stationary points, for which $d = 0$ is a solution of the problem obtained by linearizing all the data of (OMPV) with the exception of the complementarity constraint $y^T w \leqslant 0$.
- Strongly stationary points, which satisfy

$$y_k = 0,\ w_k = 0 \Rightarrow \widehat{\eta}_{y,k} \geqslant 0 \text{ and } \widehat{\eta}_{w,k} \geqslant 0, \quad k = 1, 2, \ldots, n_c.$$

It is immediate that such points are also KKT points in the nonlinear programming sense of (OMPV) [23].

If a point is strongly stationary, then it is also a stationary point of any other type [23]. Also, a stationary point of any type is a weakly stationary point [23]. In addition, an M-stationary point is also a C-stationary point. No other relation holds in general between these stationarity concepts. For an approach that uses the linearization of the data, B-stationary points seem to be the desirable outcome. However, the amount of work necessary to recognize B-stationary points may be exponential in the dimension of the problem [22].

Definition (ULSC. See [13, 24]). *A weakly stationary point (x, y, w, z) of (OMPV) satisfies the upper-level strict complementarity (ULSC) property if there exists an MPCC multiplier that satisfies*

$$y_k + w_k = 0 \ \Rightarrow \widehat{\eta}_{y,k}\widehat{\eta}_{w,k} \neq 0,\ k = 1, 2, \ldots, n_c.$$

2.2.2. Parametric mixed P variational inequalities. For fixed x, the system of constraints

$$F(x, y, w, z) = 0,\ y \leqslant 0,\ w \leqslant 0,\ w^T y = 0 \tag{2.4}$$

defines a mixed nonlinear complementarity problem, that is, an instance of a variational inequality. We can therefore interpret y, w, z as the state variables and x as the parameters of the parameterized variational inequality (2.4).

For the remainder of this work we make the following assumptions:

A1. The mappings f, g, h, F are twice continuously differentiable.

A2. The constraints involving only the parameters x satisfy, for any x, that
(i) $\nabla_x h(x)^T$ has full column rank.
(ii) $\exists p \in \mathbb{R}^n$ such that $\nabla_x h(x)p = 0$ and $\nabla g_i(x)p < 0$ whenever $g_i(x) \geqslant 0$.
(iii) the linearization $h(x) + \nabla_x h(x)d = 0,\ g(x) + \nabla_x g(x)d \leqslant 0$ is feasible.

A3. The partition $[\nabla_y F, \nabla_w F, \nabla_z F]$ is a mixed P partition (2.1).

Note that **A2(i)** and **A2(ii)** do not imply **A2(iii)**, since we allow for the point x to be infeasible for the constraints $g(x) \leqslant 0,\ h(x) = 0$. Assumption **A2** holds when $g \leqslant 0, h = 0$ represent a polyhedral set in a minimal representation.

Under **A1**–**A3** we call the problem (OMPV) containing (2.4) as a constraint *optimization of parameterized mixed P variational inequalities.* Note that (OMPV) under **A1**–**A3** was also studied in [18] but with a different algorithmic approach and outcome.

Theorem 2.3. *The nonlinear program (OMPV(c)) has a feasible linearization and satisfies the Mangasarian–Fromovitz constraint qualification (MFCQ)* [4, p. 441], [19] *at any point $(x, y, w, z, \zeta_1, \zeta_2)$.*

Proof. Consider the constraints $g(x) \leqslant 0,\ h(x) = 0,\ y \leqslant 0,\ w \leqslant 0$. Using **A2**, we obtain that there exist d_x, d_y, d_w such that

$$\begin{array}{lcl}
g(x) + \nabla_x g(x) d_x & < & 0,\\
h(x) + \nabla_x h(x) d_x & = & 0,\\
y + d_y & < & 0,\\
w + d_w & < & 0.
\end{array}$$

Choose also $d_w = 0_{n_c}$, $d_y = 0_{n_c}$, $d_z = 0_l$, and consider the linearization of the feasible set of the program (OMPV(c)) at the point $(x, y, w, z, \zeta_1, \zeta_2)$:

$$\begin{array}{rcl}
g(x) + \nabla_x g(x) d_x & \leqslant & 0,\\
h(x) + \nabla_x h(x) d_x & = & 0,\\
-\zeta_1 e_{n_c+l} - d_{\zeta_1} e_{n_c+l} \ \leqslant F(x, y, w, z) & & \\
+\nabla_{(x,y,w,z)} F(x, y, w, z)\,(d_x, d_y, d_w, d_z)^T & \leqslant & \zeta_1 e_{n_c+l} + d_{\zeta_1} e_{n_c+l},\\
y + d_y, w + d_w & \leqslant & 0,\\
y^T w + w^T d_y + y^T d_w & \leqslant & \zeta_2 + d_{\zeta_2},\\
\zeta_1 + d_{\zeta_1} & \geqslant & 0,\\
\zeta_2 + d_{\zeta_2} & \geqslant & 0.
\end{array}$$

If we choose d_{ζ_1}, d_{ζ_2} sufficiently large, then we obtain $d = (d_x, d_y,\ d_w, d_z,\ d_{\zeta_1}, d_{\zeta_2})$ such that the inequalities in the above system are strictly satisfied, which shows that (OMPV(c)) has a feasible linearization and satisfies the MFCQ everywhere. □

In the following, we introduce the notion of an ε stationary point. A variant of it has been used for Lagrange multiplier algorithms [3, Prop. 4.2.2]. This will help define an algorithm that needs to spend only a finite number of steps to approximately solve (OMPV(c)) before having to increase the penalty parameter. This property is unlike the regularization approach in [24], where the relaxed problem must be solved exactly.

Definition (ε stationary point). *We say that $(x, y, w, z, \zeta_1, \zeta_2)$ is an ε stationary point of (OMPV(c)) if there exists $(\lambda, \mu, \theta, \eta_y, \eta_w, \alpha_c, \alpha_1, \alpha_2)$ such that the following conditions are satisfied*:

$$\textit{Dual}\quad \left\{\begin{array}{l}
\nabla_x f(x, y, w, z)^T + \nabla_x h(x)^T \lambda\\
\qquad\qquad +\nabla_x g(x)^T \mu + \nabla_x F(x, y, w, z)^T (\theta^+ - \theta^-) \ = \ t_x,\\
\nabla_y f(x, y, w, z)^T + \eta_y + \alpha_c w + \nabla_y F(x, y, w, z)^T (\theta^+ - \theta^-) \ = \ t_y,\\
\nabla_w f(x, y, w, z)^T + \eta_w + \alpha_c y + \nabla_w F(x, y, w, z)^T (\theta^+ - \theta^-) \ = \ t_w,\\
\qquad \nabla_z f(x, y, w, z)^T + \nabla_z F(x, y, w, z)^T (\theta^+ - \theta^-) \ = \ t_z,\\
\qquad \|\theta^+\|_1 + \|\theta^-\|_1 + \alpha_1 = c + t_{\alpha 1};\ \alpha_c + \alpha_2 = c + t_{\alpha 2},\\
\qquad\quad \mu \geqslant 0;\ \eta_y, \eta_w \geqslant 0; \theta^+, \theta^- \geqslant 0;\ \alpha_c, \alpha_1, \alpha_2 \geqslant 0,
\end{array}\right.$$

$$\textit{Primal}\quad \left\{\begin{array}{rcl}
g(x) & \leqslant & t_g,\\
h(x) & = & t_h,\\
-\zeta_1 e_{n_c+l} - t_{1F} & \leqslant & F(x, y, w, z) \leqslant \zeta_1 e_{n_c+l} + t_{2F},\\
y, w & \leqslant & 0,\\
y^T w & \leqslant & \zeta_2 + t_c,\\
\zeta_1, \zeta_2 & \geqslant & 0,
\end{array}\right.$$

$$\textit{Compl.}\quad \left\{\begin{array}{l}
(\zeta_1 e_{n_c+l} - F)^T \theta^+ + (F + \zeta_1 e_{n_c+l})^T \theta^- = t_{cF},\\
\alpha_c(\zeta_2 - w^T y) = t_{cc};\ g(x)^T \mu = t_{cg},\\
|\alpha_2 \zeta_2| \leqslant t_{cp};\ |\alpha_1 \zeta_1| \leqslant t_{cp};\ \left|y^T \eta_y\right| \leqslant t_{cp};\ \left|w^T \eta_w\right| \leqslant t_{cp},
\end{array}\right.$$

where the size of the inexactness is bounded above by ε, that is,

$$\left\|t_g, t_h, t_{1F}, t_{2F}, t_c, t_x, t_y, t_w, t_z, t_{\alpha 1,} t_{\alpha 2}, t_{cc}, t_{cF}, t_{cg}, t_{cp}\right\|_\infty \leqslant \varepsilon.$$

When defining our approximate stationary point, we assumed that bound constraints that involve a zero bound and the sign condition for the resulting multiplier can be exactly enforced. For interior point methods this assumption is readily satisfied (even in finite-precision arithmetic).

Definition (global convergence safeguard). *We say that a nonlinear programming algorithm has a global convergence safeguard if any accumulation point of the algorithm is one of the following*:

1. *An infeasible point of the nonlinear program at which the linearization of the constraints is infeasible.*
2. *A feasible point of the nonlinear program that does not satisfy the MFCQ.*
3. *A feasible point of the nonlinear program that satisfies the MFCQ and that is a KKT point of the nonlinear program.*

An example of such an algorithm is FilterSQP [11]. We use the following assumption.

Alg1. The nonlinear programming algorithm has a global convergence safeguard.

Lemma 2.4. *Any accumulation point of a nonlinear programming algorithm that satisfies **Alg1** and is applied to (OMPV(c)) is a KKT point.*

Proof. Since the nonlinear programming algorithm has a global convergence safeguard, it cannot end in case 1 or case 2, following Theorem 2.3. The conclusion follows, and the proof is complete. □

Based on this lemma, we will assume in the rest of this work that we have access to an algorithm that satisfies **Alg1** and that can provide, for any $\varepsilon > 0$, an ϵ stationary point of (OMPV(c)). We can now state the following theorem, which will be our main convergence tool.

Theorem 2.5. *Assume that (OMPV) satisfies **A1**, **A2**, and **A3** and that the relaxed problems (OMPV(c)) are solved with an algorithm that satisfies **Alg1**. Let $(x^n, y^n, w^n, z^n, \zeta_1^n, \zeta_2^n)$ be an ε^n stationary point of (OMPV(c^n)). Assume that $\lim_{n\to\infty} c^n = \infty$, $\lim_{n\to\infty} \varepsilon^n = 0$, and $\lim_{n\to\infty} c^n\varepsilon^n = 0$. Then any accumulation point $(x^*, y^*, w^*, z^*, \zeta_1^*, \zeta_2^*)$ of $(x^n, y^n, w^n, z^n, \zeta_1^n, \zeta_2^n)$ must satisfy $\zeta_1^* = 0$, $\zeta_2^* = 0$, and (x^*, y^*, w^*, z^*) is a feasible C-stationary point of (OMPV).*

Proof (feasibility). From our assumption, $(x^n, y^n, w^n, z^n, \zeta_1^n, \zeta_2^n)$ is an ε^n stationary point of (OMPV(c^n)), which, by Theorem 2.3, satisfies the MFCQ everywhere. There must exist the Lagrange multipliers $\lambda^n \in \mathbb{R}^{n_e}$, $\mu^n \in \mathbb{R}_+^{n_i}$, $\theta^{+n}, \theta^{-n} \in \mathbb{R}_+^{n_c+l}$, $\eta_y^n \in \mathbb{R}_+^{n_c}$, $\eta_w^n \in \mathbb{R}_+^{n_c}$, $\alpha_1^n, \alpha_2^n, \alpha_c^n \in \mathbb{R}_+$ that, together with $(x^n, y^n, w^n, z^n, \zeta_1^n, \zeta_2^n)$, satisfy the ε^n approximate KKT conditions, which include the following equations:

(2.5)
$$\begin{array}{rcl}
\nabla_x f(x^n, y^n, w^n, z^n)^T + \nabla_x h(x^n)^T \lambda^n & & \\
+\nabla_x g(x^n)^T \mu^n + \nabla_x F(x^n, y^n, w^n, z^n)^T (\theta^{+n} - \theta^{-n}) & = & t_x^n, \\
\nabla_y f(x^n, y^n, w^n, z^n)^T + \alpha_c^n w^n + \eta_y^n + \nabla_y F(x^n, y^n, w^n, z^n)^T (\theta^{+n} - \theta^{-n}) & = & t_y^n, \\
\nabla_w f(x^n, y^n, w^n, z^n)^T + \alpha_c^n y^n + \eta_w^n + \nabla_w F(x^n, y^n, w^n, z^n)^T (\theta^{+n} - \theta^{-n}) & = & t_w^n, \\
\nabla_z f(x^n, y^n, w^n, z^n)^T + \nabla_z F(x^n, y^n, w^n, z^n)^T (\theta^{+n} - \theta^{-n}) & = & t_z^n, \\
(\zeta_1^n e_{n_c+l} - F(x^n, y^n, w^n, z^n))^T \theta^{+n} + (F(x^n, y^n, w^n, z^n) + \zeta_1^n e_{n_c+l})^T \theta^{-n} & = & t_{cF}, \\
\alpha_1^n + ||\theta^{+n}||_1 + ||\theta^{-n}||_1 = c^n + t_{\alpha_1}^n; g(x^n) \leqslant t_g^n, \ \ y^n \leqslant 0, \ \ w^n & \leqslant & 0, \\
\alpha_c^n (w^{nT} y^n - \zeta_2^n) = t_{cc}^n, \ \ \alpha_2^n + \alpha_c^n = c^n + t_{\alpha_2}^n; \ (w^{nT} y^n - \zeta_2^n) \leqslant t_c^n, \ \ \zeta_2^n & \geqslant & 0, \\
g(x_n)^T \mu^n = t_{cg}^n, \ \ |\alpha_1^n \zeta_1^n| \leqslant t_{cp}; \ \ \left|y_n^T \eta_y^n\right| \leqslant t_{cp}^n, \ \ \left|w^{nT} \eta_w^n\right| \leqslant t_{cp}^n, \ \ |\alpha_2^n \zeta_2^n| & \leqslant & t_{cp}^n,
\end{array}$$

$$\left\|t_x^n, t_y^n, t_w^n, t_z^n, t_{cc}^n, t_{cF}^n, t_{cg}^n, t_{cy}^n, t_{cw}^n, t_{\alpha_1}^n, t_{\alpha_2}^n, t_{cp}^n\right\|_\infty \leqslant \varepsilon^n.$$

We ignore for the time being the effect of the variable ζ_1 on the optimality conditions. We also denote $\theta^n = \theta^{+n} - \theta^{-n}$ and $\widetilde{\lambda}^n = \left(\lambda^n, \mu^n, \theta^n, \eta_y^n, \eta_w^n, \alpha_c^n, \alpha_2^n\right)$.

Since $\alpha_c^n + \alpha_2^n = c^n + t_{\alpha_2}^n$, $c^n \to \infty$ and $\varepsilon^n \to 0$, we must have that $\|\tilde{\lambda}^n\|_\infty \to \infty$ as $n \to \infty$. Therefore, the sequence $\frac{\widetilde{\lambda}^n}{\|\widetilde{\lambda}^n\|_\infty}$, admits an accumulation point

$$\widetilde{\lambda}^* = \left(\lambda^*, \mu^*, \theta^*, \eta_y^*, \eta_w^*, \alpha_c^*, \alpha_2^*\right)$$

that satisfies $\|\tilde{\lambda}^*\|_\infty = 1$ and $\mu^* \geqslant 0$, $\eta_y^* \geqslant 0$, $\eta_w^* \geqslant 0$, $\alpha_c^* \geqslant 0$, and $\alpha_2^* \geqslant 0$. We can assume without loss of generality (after eventually restricting the respective sequences to subsequences) that

$$\frac{\widetilde{\lambda}^n}{\left\|\widetilde{\lambda}^n\right\|_\infty} \to \widetilde{\lambda}^* \text{ and } (x^n, y^n, w^n, z^n, \zeta_2^n) \to (x^*, y^*, w^*, z^*, \zeta_2^*).$$

We now divide (2.5) by $\|\tilde{\lambda}^n\|_\infty$ and take the limit as $n \to \infty$ to obtain

$$\begin{array}{rcl}\nabla_x h(x^*)^T\lambda^* + \nabla_x g(x^*)^T\mu^* + \nabla_x F(x^*, y^*, w^*, z^*)^T\theta^* & = & 0,\\ \alpha_c^* w^* + \eta_y^* + \nabla_y F(x^*, y^*, w^*, z^*)^T\theta^* & = & 0,\\ \alpha_c^* y^* + \eta_w^* + \nabla_w F(x^*, y^*, w^*, z^*)^T\theta^* & = & 0,\\ \nabla_z F(x^*, y^*, w^*, z^*)^T\theta^* & = & 0,\\ g(x^*) \leqslant 0,\;\; y^* \leqslant 0,\;\; w^* \leqslant 0,\;\; ({w^*}^T y^* - \zeta_2^*) \leqslant 0,\;\; \zeta_2^* & \geqslant & 0,\\ g(x^*)^T\mu^* = 0,\;\; {y^*}^T\eta_y^* = 0,\;\; {w^*}^T\eta_w^* = 0,\;\; \alpha_c^*({w^*}^T y^* - \zeta_2^*) = 0,\;\; \alpha_2^*\zeta_2^* & = & 0.\end{array} \tag{2.6}$$

Now take an index k such that $1 \leqslant k \leqslant n_c$. Since $\alpha_c^* \geqslant 0$, $w_k^* \leqslant 0$, $y_k^* \leqslant 0$, $\eta_{w,k}^* \geqslant 0$, and $\eta_{y,k}^* \geqslant 0$, we must have that

$$\eta_{y,k}^* + \alpha_c^* w_k^* > 0 \Rightarrow \eta_{y,k}^* > 0 \overset{(2.6)}{\Rightarrow} y_k^* = 0 \Rightarrow \eta_{w,k}^* + \alpha_c^* y_k^* \geqslant 0.$$

Similarly, we have that

$$\eta_{w,k}^* + \alpha_c^* y_k^* > 0 \Rightarrow \eta_{w,k}^* > 0 \overset{(2.6)}{\Rightarrow} w_k^* = 0 \Rightarrow \eta_{y,k}^* + \alpha_c^* w_k^* \geqslant 0.$$

We therefore conclude that, for $k = 1, 2, \ldots, n_c$, we must have that

$$(\eta_{w,k}^* + \alpha_c^* y_k^*)(\eta_{y,k}^* + \alpha_c^* w_k^*) \geqslant 0.$$

We can therefore define for $k = 1, 2, \ldots, n_c$ the quantities

$$d_k = \begin{cases} 1 & \text{if} \quad (\eta_{w,k}^* + \alpha_c^* y_k^*) > 0 \text{ or } (\eta_{y,k}^* + \alpha_c^* w_k^*) > 0,\\ -1 & \text{if} \quad (\eta_{w,k}^* + \alpha_c^* y_k^*) < 0 \text{ or } (\eta_{y,k}^* + \alpha_c^* w_k^*) < 0,\\ 1 & \text{if} \quad (\eta_{w,k}^* + \alpha_c^* y_k^*) = \ (\eta_{y,k}^* + \alpha_c^* w_k^*) = 0.\end{cases}$$

From our observation and the definition of $d_k,\ k = 1, 2, \ldots, n_c$, we must have that

$$d_k(\eta_{w,k}^* + \alpha_c^* y_k^*) \geqslant 0 \quad \text{and} \quad d_k(\eta_{y,k}^* + \alpha_c^* w_k^*) \geqslant 0,\ k = 1, 2, \ldots, n_c.$$

We denote by $D \in \mathbb{R}^{n_c \times n_c}$ the matrix whose diagonal elements are d_k, $k = 1, 2, \ldots, n_c$. The middle equations from (2.6) and our definition of D imply that

$$\begin{array}{lcl} D\nabla_y F(x^*, y^*, w^*, z^*)^T\theta^* & \leqslant & 0,\\ D\nabla_w F(x^*, y^*, w^*, z^*)^T\theta^* & \leqslant & 0,\\ \nabla_z F(x^*, y^*, w^*, z^*)^T\theta^* & = & 0.\end{array}$$

From **A3**, Lemma 2.1, and Theorem 2.2, the preceding equation implies that $\theta^* = 0$. Replacing this in (2.6), we obtain that

$$\nabla_x h(x^*)^T \lambda^* + \sum_{i \in A(x^*)} \nabla_x g_i(x^*)^T \mu_i^* = 0,$$

which, from **A2**, implies that $\lambda^* = 0$ and $\mu^* = 0$. The fact that $\theta^* = 0$ also implies from (2.6) that

$$\eta_y^* + \alpha_c^* w^* = 0, \quad \eta_w^* + \alpha_c^* y^* = 0. \tag{2.7}$$

Multiplying the first relation by ${y^*}^T$, the second by ${w^*}^T$, and using the complementarity relations ${y^*}^T \eta_y^* = 0$ and ${w^*}^T \eta_w^* = 0$ from (2.6), we obtain that

$$\alpha_c^* {y^*}^T w^* = 0. \tag{2.8}$$

We have the following cases:

1. $\alpha_c^* > 0$. Then (2.8) implies that ${y^*}^T w^* = 0$. From $\alpha_c^*({w^*}^T y^* - \zeta_2^*) = 0$ of (2.6) we get that $\zeta_2^* = {y^*}^T w^* = 0$.
2. $\alpha_c^* = 0$. Then from (2.7) we get that $\eta_y^* = \eta_w^* = 0$. It then follows that the only nonzero component of $\widetilde{\lambda}^*$ is α_2^*, which must then satisfy $\alpha_2^* = \|\widetilde{\lambda}^*\|_\infty = 1$. The complementarity condition $\alpha_2^* \zeta_2^* = 0$ from (2.6) now implies $\zeta_2^* = 0$.

In either case we obtain $\zeta_2^* = 0$.

Assume now that $\zeta_1^n \xrightarrow{n \to \infty} \zeta_1^* > 0$. Define $\widetilde{\lambda}^n = (\lambda^n ,\mu^n, \theta^{+n}, \theta^{-n}, \eta_y^n, \eta_w^n, \alpha_c^n, \alpha_1^n, \alpha_2^n)$ that, by an observation similar to that of the previous paragraphs, must satisfy $\|\widetilde{\lambda}^n\|_\infty \to \infty$ as $n \to \infty$. Consider an accumulation point $\widetilde{\lambda}^* = \left(\lambda^*, \mu^*, \theta^{+*}, \theta^{-*}, \eta_y^*, \eta_w^*, \alpha_c^*, \alpha_1^*, \alpha_2^*\right)$ of the sequence $\frac{\widetilde{\lambda}^n}{||\widetilde{\lambda}^n||_\infty}$. Using the complementarity relationships from (2.5), we obtain that ${\theta^{+*}}^T \theta^{-*} = 0$. Indeed, if we assume that $\theta_k^{+*} > 0$, for some $k = 1, 2, \ldots, n_c$, the complementarity relations from (2.5), in which we take the limit after dividing with $\|\widetilde{\lambda}^n\|_\infty \to \infty$, result in $0 < \zeta_1^* = F_k(x^*, y^*, w^*, z^*)$, which in turn implies that $\theta_k^{-*} = 0$.

Repeating the argument that led to the conclusion that $\zeta_2^* = 0$, we obtain that $0 = \theta^* = \theta^{+*} - \theta^{-*}$ which, in conjunction with ${\theta^{+*}}^T \theta^{-*} = 0$, implies $\theta^{+*} = \theta^{-*} = 0$ and $(\lambda^*, \mu^*, \theta^{+*}, \theta^{-*}) = 0$ as well as $\eta_y^* + \alpha_c^* w^* = 0,$ and $\eta_w^* + \alpha_c^* y^* = 0$. The last two equations imply that $\eta_y^n, \eta_w^n = O(\alpha_c^n)$. Using the definition of $\widetilde{\lambda}$ we obtain that $\|\widetilde{\lambda}^n\|_\infty \leqslant \Gamma \|(\alpha_c^n, \alpha_1^n, \alpha_2^n)\|_\infty \leqslant \Gamma c^n \, \forall \, n$ sufficiently large, for some positive Γ, which in turn implies that $\frac{\theta^{+n}}{c^n} \xrightarrow{n \to \infty} 0, \frac{\theta^{-n}}{c^n} \xrightarrow{n \to \infty} 0$. Using these relations, together with $\alpha_1^n + ||\theta^{+n}||_1 + ||\theta^{-n}||_1 = c^n + t_{\alpha_1}^n$, we obtain that $\alpha_1^* > 0$. However, from the limit of complementarity relationships from (2.5), which are obtained after dividing with $\|\widetilde{\lambda}^n\|_\infty \to \infty$, we obtain that $\alpha_1^* \zeta_1^* = 0$. This is a contradiction to the initial assumption that $\zeta_1^* > 0$. We must therefore have that $\zeta_1^* = 0$ in addition to $\zeta_2^* = 0$, which shows that the limit point (x^*, y^*, w^*, z^*) must be feasible. □

Proof (C-stationarity). We return to (2.5). We define

$$\widehat{\eta}_y^n = \eta_y^n + \alpha_c^n w^n, \qquad \widehat{\eta}_w^n = \eta_w^n + \alpha_c^n y^n. \tag{2.9}$$

Following our definition of an ε^n stationary point, we have that, $\forall \; k = 1, 2, \ldots, n_c$, the following relations hold:

$$\begin{aligned} \eta^n_{y,k},\, \eta^n_{w,k} \geq 0; \quad y^n_k,\, w^n_k \leq 0,& \\ \alpha^n_c \leq c^n + \varepsilon^n,& \\ \left|\eta^n_{y,k} y^n_k\right| + \left|\eta^n_{w,k} w^n_k\right| \leq 2\varepsilon^n.& \end{aligned}$$

In turn, this implies that, $\forall k = 1, 2, \ldots, n_c$,

$$\alpha^n_c \left(\eta^n_{y,k} y^n_k + \eta^n_{w,k} w^n_k\right) \geqslant -2c^n \varepsilon^n - 2\left(\varepsilon^n\right)^2,$$

and, thus, that

(2.10)

$$\widehat{\eta}^n_{y,k}\, \widehat{\eta}^n_{w,k} = \eta^n_{y,k}\eta^n_{w,k} + (\alpha^n_c)^2 y^n_k w^n_k + \alpha^n_c \left(\eta^n_{y,k} y^n_k + \eta^n_{w,k} w^n_k\right) \geqslant -2c^n\varepsilon^n - 2(\varepsilon^n)^2 \overset{n\to\infty}{\longrightarrow} 0.$$

Define

$$\widehat{\lambda}^n = \left(\lambda^n, \mu^n, \theta^n, \widehat{\eta}^n_y, \widehat{\eta}^n_w\right).$$

The components of $\widehat{\lambda}^n$ satisfy the following set of equations derived from (2.5):

$$\begin{array}{rcl} \nabla_x f(x_n, y_n, w_n, z_n)^T + \nabla_x h(x_n)^T \lambda^n & & \\ +\nabla_x g(x_n)^T \mu^n + \nabla_x F(x_n, y_n, w_n, z_n)^T \theta^n & = & t^n_x, \\ \nabla_y f(x_n, y_n, w_n, z_n)^T + \widehat{\eta}^n_y + \nabla_y F(x_n, y_n, w_n, z_n)^T \theta^n & = & t^n_y, \\ \nabla_w f(x_n, y_n, w_n, z_n)^T + \widehat{\eta}^n_w + \nabla_w F(x_n, y_n, w_n, z_n)^T \theta^n & = & t^n_w, \\ \nabla_z f(x_n, y_n, w_n, z_n)^T + \nabla_z F(x_n, y_n, w_n, z_n)^T \theta^n & = & t^n_z, \\ h(x_n) = t^n_h,\ g(x_n) \leqslant t^n_g,\ \ y_n \leqslant 0,\ \ w_n \leqslant 0, & & \end{array} \tag{2.11}$$

where $\left\|t^n_g, t^n_h, t^n_x, t^n_y, t^n_w, t^n_z, t^n_y\right\|_\infty \leqslant \varepsilon^n$. Assume that $\widehat{\lambda}^n$ admits a subsequence that diverges to ∞. We can assume without loss of generality that the entire sequence itself diverges to ∞. Define the sequence

$$\widetilde{\widehat{\lambda}}^n = \frac{\widehat{\lambda}^n}{\left\|\widehat{\lambda}^n\right\|_\infty},$$

which, being bounded, must admit a convergent subsequence. We assume, again without loss of generality, that the sequence $\widetilde{\widehat{\lambda}}^n$ is itself convergent to

$$\widetilde{\widehat{\lambda}}^* = \left(\widetilde{\lambda}^*, \widetilde{\mu}^*, \widetilde{\theta}^*, \widetilde{\eta}^*_y, \widetilde{\eta}^*_w\right),$$

with $\|\widetilde{\widehat{\lambda}}^*\|_\infty = 1$. From the construction of $\widehat{\lambda}^n$ we must have that $\widetilde{\mu}^* \geqslant 0$, whereas from (2.10) we must have that

$$\widetilde{\eta}^*_{y,k}\, \widetilde{\eta}^*_{w,k} \geqslant 0, \qquad k = 1, 2, \ldots, n_c. \tag{2.12}$$

Now, dividing all equations involving multipliers of (2.11) by $\|\widehat{\lambda}^n\|_\infty$ and taking the limit as $n \to \infty$, we obtain that

$$\begin{array}{lcl} \nabla_x h(x^*)^T \widetilde{\lambda}^* + \nabla_x g(x^*)^T \widetilde{\mu}^* + \nabla_x F(x^*, y^*, w^*, z^*)^T \widetilde{\theta}^* & = & 0 \\ \widetilde{\eta}^*_y + \nabla_y F(x^*, y^*, w^*, z^*)^T \widetilde{\theta}^* & = & 0 \\ \widetilde{\eta}^*_w + \nabla_w F(x^*, y^*, w^*, z^*)^T \widetilde{\theta}^* & = & 0 \\ \nabla_z F(x^*, y^*, w^*, z^*)^T \widetilde{\theta}^* & = & 0 \\ h(x^*) = 0\ g(x^*) \leqslant 0,\ \ g(x^*)^T \widetilde{\mu}^* = 0,\ \ y^* \leqslant 0,\ \ w^* \leqslant 0. & & \end{array} \tag{2.13}$$

Using the same argument that we applied to (2.6) and that led to the conclusion $\theta^* = 0$ and, subsequently, $\zeta^* = 0$, we get that (2.13), (2.12), and **A3** imply that $\widetilde{\theta}^* = 0$. In turn, this implies that $\widetilde{\eta}_y^* = \widetilde{\eta}_w^* = 0$ and, from **A2** and from using the complementarity relation on the last line of (2.13), that $\widetilde{\lambda}^* = 0$, $\widetilde{\mu}^* = 0$, and thus $\widetilde{\widetilde{\lambda}}^* = 0$, which is a contradiction to $\|\widetilde{\widetilde{\lambda}}^*\|_\infty = 1$. This implies that the sequence $\widehat{\lambda}^n$ must be bounded. Let

$$\widehat{\lambda}^* = \left(\lambda^*, \mu^*, \theta^*, \widehat{\eta}_y^*, \widehat{\eta}_w^*\right)$$

be a limit point of this sequence. We assume without loss of generality that it is the unique limit point. From (2.10) we must have that

$$\widehat{\eta}_{y,k}^* \, \widehat{\eta}_{w,k}^* \geqslant 0, \qquad k = 1, 2, \ldots, n_c. \tag{2.14}$$

From our definition of $\widehat{\eta}_w^n$ and $\widehat{\eta}_y^n$ (see (2.9)), it does not immediately follow that the corresponding limit point satisfies a complementarity relation with w^* and, respectively, y^*. Although we have that $\eta_{w,k}^n w_k^n \to 0$ and $\eta_{y,k}^n y_k^n \to 0$ for $k = 1, 2, \ldots, n_c$ from (2.5), the additional terms $\alpha_c y_k^n$ and $\alpha_c w_k^n$ may potentially prevent a corresponding complementarity relation from holding for $\widehat{\eta}_w^n$ and $\widehat{\eta}_y^n$, or the respective limits, since α_c^n may diverge into ∞.

In the following we show that this is not the case. We prove that $\widehat{\eta}_{y,k}^* y_k^* = 0$. Since $\widehat{\lambda}^n$ is bounded we must have that

$$O(1) = \widehat{\eta}_{y,k}^n = \eta_{y,k}^n + \alpha_c^n w_k^n, \quad O(1) = \widehat{\eta}_{w,k}^n = \eta_{w,k}^n + \alpha_c^n y_k^n, \tag{2.15}$$

and that $y_k^* = 0 \Rightarrow y_k^n \overset{n\to\infty}{\longrightarrow} 0 \Rightarrow \widehat{\eta}_{y,k}^* y_k^* = 0$, which would complete the proof.

Assume then that $y_k^* < 0$. Since the limit point is feasible for (OMPV), we must have that $w_k^* = 0$, and therefore that $w_k^n \overset{n\to\infty}{\longrightarrow} 0$. Multiplying the second equation in (2.15) by w_k^n, we obtain that $\lim_{n\to\infty} \eta_{w,k}^n w_k^n + \alpha_c^n y_k^n w_k^n = 0$.

Since, from the definition of ε^n stationary points, we have that $\|\eta_{w,k}^n w_k^n\| \leq \varepsilon^n$, this implies that $\alpha_c^n y_k^n w_k^n \overset{n\to\infty}{\longrightarrow} 0$. Using the first equation of (2.15) and the fact that $(x^n, y^n, w^n, z^n, \zeta_1^n, \zeta_2^n)$ is an ε^n stationary point, we obtain that

$$\widehat{\eta}_{y,k}^n y_k^n = \eta_{y,k}^n y_k^n + \alpha_c^n y_k^n w_k^n \overset{n\to\infty}{\longrightarrow} 0.$$

The last equation proves that $\widehat{\eta}_{y,k}^* y_k^* = 0$, $k = 1, 2, \ldots, n_c$. Similarly, it also follows that $\widehat{\eta}_{w,k}^* w_k^* = 0$, $k = 1, 2, \ldots, n_c$.

Now, taking the limit in (2.11) as $n \to \infty$ and using (2.14), and $\widehat{\lambda}^n \overset{n\to\infty}{\longrightarrow} \widehat{\lambda}^*$, we obtain that (x^*, y^*, w^*, z^*) is a C-stationary point with MPCC multiplier $\widehat{\lambda}^* = (\lambda^*, \mu^*, \theta^*, \widehat{\eta}_y^*, \widehat{\eta}_w^*)$. The proof is complete. □

Note that, in order to obtain a similar result, MPCC-LICQ was needed in [24]. The preceding result also allows us to characterize all local solutions of (OMPV).

Corollary 2.6. *Assume that (OMPV) satisfies* ***A1****,* ***A2****, and* ***A3*** *everywhere and that* (x^*, y^*, w^*, z^*) *is a strict local minimum of (OMPV). Then* (x^*, y^*, w^*, z^*) *is a C-stationary point of (OMPV).*

Proof. It is immediate from the definition of (OMPV(c)) that $(x^c, y^c, w^c, z^c, \zeta^c)$ is a local solution of (OMPV(c)) if and only if (x^c, y^c, w^c, z^c) is a local solution of

$$(OMPV1(c)) \quad \begin{array}{cl} \displaystyle\min_{x,y,w,z} & f(x, y, w, z) + c y^T w + c \, \|F(x, y, w, z)\|_\infty \\ \text{s.t.} & g(x) \leqslant 0, \; h(x) = 0, \; y, w \leqslant 0. \end{array}$$

If $\widehat{x} = (x^*, y^*, w^*, z^*)$ is a strict local minimum of (OMPV), then there exist $\delta > 0$ and a ball $B(\widehat{x}, \delta)$, whose boundary we denote by Γ, such that for any $(x, y, w, z) \in \Gamma$, which is a feasible point of (OMPV1(c)), we must have that

$$\max\{f(x, y, w, z) - f(x^*, y^*, w^*, z^*),\ y^T w + \|F(x, y, w, z)\|_\infty\} > 0.$$

This implies that there exists $\widehat{c}$ such that, $\forall\, \gamma > \widehat{c}$, we have that for any (x, y, w, z), which is a feasible point of (OMPV1(c)) on the boundary Γ of $B(\widehat{x}, \delta)$, we must have that

$$f(x, y, w, z) - f(x^*, y^*, w^*, z^*) + \gamma \left(y^T w + \|F(x, y, w, z)\|_\infty\right) > 0.$$

If this is not true, then for any n there exists $\gamma_n > n$ such that, for some $(x^n$, y^n, w^n, $z^n) \in \Gamma$, which is a feasible point of (OMPV1(c)), we have that

$$f(x^n, y^n, w^n, z^n) - f(x^*, y^*, w^*, z^*) + \gamma_n \left({y^n}^T w^n + \|F(x^n, y^n, w^n, z^n)\|_\infty\right) \leqslant 0. \tag{2.16}$$

Since Γ is compact, the sequence (x^n, y^n, w^n, z^n) has an accumulation point $(x^\circ$, y°, w°, $z^\circ) \in \Gamma$ that must be feasible for (OMPV1(c)). Dividing (2.16) by γ_n and taking the limit as $n \to \infty$, we get that ${y^\circ}^T w^\circ = 0$ and that $F(x^\circ, y^\circ, w^\circ, z^\circ) = 0$, that is, that $(x^\circ, y^\circ, w^\circ, z^\circ)$ is, in effect, feasible for (OMPV). But (2.16) also implies that, for all n,

$$f(x^n, y^n, w^n, z^n) - f(x^*, y^*, w^*, z^*) \leqslant 0.$$

Taking the limit in the last inequality, we obtain that

$$f(x^\circ, y^\circ, w^\circ, z^\circ) - f(x^*, y^*, w^*, z^*) \leqslant 0,$$

which contradicts our choice of δ.

Therefore, $\widehat{c}$ with the properties specified above must exist. This shows that, for $c > \widehat{c}$, (OMPV1(c)) will have a local solution inside of $B(\widehat{x}, \delta)$. For all $n > \widehat{c}$ let (x^n, y^n, w^n, z^n) be the local solution of (OMPV1(n)) in $B(\widehat{x}, \delta)$ with the lowest value. By an argument similar to that which led to the existence of $\widehat{c}$, it follows that $(x^n, y^n, w^n, z^n) \to (x^*, y^*, w^*, z^*)$. It also follows from the observation at the beginning of the proof that $(x^n, y^n, w^n, z^n, F(x^n, y^n, w^n, z^n), {y^n}^T w^n)$ is a local solution, and thus a stationary point, of (OMPV(n)). From Theorem 2.5 it thus follows that (x^*, y^*, w^*, z^*) is a C-stationary point of (OMPV). The proof is complete. □

We note that the above result could also be proven using [23, Thm. 2], after one proves that, under **A1**, **A2**, and **A3**, (OMPV) satisfies MPCC–MFCQ at any solution. The proof follows once we use the dual form of MPCC–MFCQ in conjunction with Theorem 2.2. However, the resulting proof is not shorter than the one we just provided.

2.3. A globally convergent modified elastic mode for the optimization of parameterized mixed P variational inequalities. We now describe our adaptive elastic-mode approach. Although the global convergence result we have presented in the preceding subsection deals with the situation in which $c^n \to \infty$, we are interested in also allowing the penalty parameter to remain bounded because, in that case, we can recover a strongly stationary point [1]. This is a major advantage over

Table 2.1
Elastic-mode algorithm.

```
Choose  $c_0 > 0$, $n = 0$, $K > 1$, an integer $q \geqslant 1$ and a sequence $\varepsilon^n \to 0$.
for $n = 1, 2, \ldots$
        Find  an $\varepsilon^n$ solution $\left(x^{c^n}, y^{c^n}, w^{c^n}, z^{c^n}, \zeta_1^{c^n}, \zeta_2^{c^n}\right)$ of (OMPV($c^n$)).
        If $\zeta_1^{c^n} + \zeta_2^{c^n} > (\varepsilon^n)^{\frac{1}{q}}$,
              update $c$ :  $c^{n+1} = Kc^n$ and $n$ :  $n = n + 1$.
        end if
end for
```

regularization and smoothing methods which, even under the strongest assumptions, recover a solution of the original problem only in the limit of the range of the smoothing/regularization parameter [13, 24]. An important issue in that case is how the penalty parameter c^n should be chosen. Since MPCC does not have bounded Lagrange multipliers at a solution, one cannot apply the update that takes into account the local size of the Lagrange multipliers [3]. Here, we select c^n based on a comparison with $(\varepsilon^n)^{1/q}$, where $q \geq 2$ is an integer. In fact, when testing for the size of ζ_1^n, ζ_2^n, one may want to compare it with the size of the solution of the quadratic subproblem of an SQP method [1]. The equivalent test in that case would require $q = 2$, and one can show that it does not locally interfere with superlinear convergence for a method that uses exact second-order derivatives when it converges to a strongly stationary point, as in [1]. That argument is tenuous and beyond the scope of this paper. Here we simply assume that the parameter q is provided by the user, and the test takes the form shown in Table 2.1.

Theorem 2.7. *Consider the algorithm described in Table* 2.1. *Assume that the problem (OMPV) satisfies* ***A*1**, ***A*2**, *and* ***A*3**. *Assume that, for a fixed* c^n, *the subproblem (OMPV(c^n)) is solved with a nonlinear programming algorithm that satisfies* ***Alg*1**. *Assume that* $\lim_{n\to\infty} c^n\varepsilon^n = 0$. *Assume that the algorithm does not diverge to* ∞ *and produces* $\left(x^{c^n}, y^{c^n}, w^{c^n}, z^{c^n}, \zeta_1^{c^n}, \zeta_2^{c^n}\right)$. *Then either*

1. *the penalty parameter update rule is activated a finite number of times, and then any accumulation point of* $\left(x^{c^n}, y^{c^n}, w^{c^n}, z^{c^n}\right)$ *is a strongly stationary point of (OMPV), or*
2. *the penalty parameter update rule is activated an infinite number of times, and then any accumulation point of* $\left(x^{c^n}, y^{c^n}, w^{c^n}, z^{c^n}\right)$ *is a C-stationary point of (OMPV).*

Proof. Part 1. Since the penalty parameter update rule is activated only a finite number of times, it follows that there exist a $c^* > 0$ and an n_0 such that the penalty parameter satisfies $c^n = c^* \, \forall n \geqslant n_0$. Therefore any accumulation point $(x^*, y^*, w^*, z^*, \zeta_1^*, \zeta_2^*)$ of $\left(x^{c^n}, y^{c^n}, w^{c^n}, z^{c^n}, \zeta_1^{c^n}, \zeta_2^{c^n}\right)$ is a stationary point of (OMPV(c^*)) that verifies, from the test of the update rule, $\zeta_1^* = \zeta_2^* = 0$. It can be immediately verified that such points are strongly stationary KKT points of (OMPV), much as in [1, 3]. Since such verification is fairly straightforward from the above references, it is omitted here.

Part 2. If the penalty parameter is updated an infinite number of times, it follows that c^n is increased to ∞, and, by applying Theorem 2.5, we get that any accumulation point (x^*, y^*, w^*, z^*) of $\left(x^{c^n}, y^{c^n}, w^{c^n}, z^{c^n}\right)$ is a C-stationary point of (OMPV). □

3. When MPCC-LICQ holds, accumulation points are M-stationary points. The convergence result can be improved when we consider stronger stationary conditions. In the rest of this section, we assume that the linear independence constraint qualification in an MPCC sense (MPCC-LICQ) holds at the solution of the MPCC under consideration.

Definition (MPCC-LICQ). *We say that the MPCC-LICQ holds at a feasible (x, y, w, z) point of (OMPV) if the gradients of all active constraints of (OMPV) at (x, y, w, z), with the exception of the complementarity constraint $y^T w \leqslant 0$, are linearly independent.*

To accommodate the fact that a solution of a nonlinear program is never exactly determined, we will work again with approximate optimality conditions.

Definition (χ active constraints). *We say that a constraint $\tilde{g}(\tilde{x}) \leqslant 0$ $(= 0)$ of a nonlinear program is χ active at a point $\tilde{x}^*$ if we have that $|\tilde{g}(\tilde{x}^*)| \leqslant \chi$.*

Definition ((ϵ, χ) second-order stationary point). *We say that the point $\tilde{x} = (x, y, w, z, \zeta_1, \zeta_2)$, together with a Lagrange multiplier $\widetilde{\lambda} = (\lambda\ ,\ \mu,\ \theta^{+n},\ \theta^{-n},\ \eta_y,\ \eta_w,\ \alpha_c,\ \alpha_1,\ \alpha_2)$ is an (ε, χ) second-order point of (OMPV(c)) if*

1. *$\tilde{x} = (x, y, w, z, \zeta_1, \zeta_2)$, is an ε stationary point of (OMPV(c)) that satisfies* exactly *the primal-dual complementarity involving the slack variables $\eta_y^T y = 0$, $\eta_w^T w = 0$.*
2. *$u^T \Lambda_{xx}^c(\tilde{x}, \tilde{\lambda}) u > 0$ for any u that is at the same time in the null space of the gradients of the active bound constraints of (OMPV(c)) and null space of a subset of the χ-active nonbound constraints of (OMPV(c)).*

Here, $\Lambda^c(\tilde{x}, \tilde{\lambda})$ is the Lagrangian function of (OMPV(c)). For a general nonlinear program that satisfies linear independence and strict complementarity at a solution, the above condition is equivalent to second-order sufficient conditions. In most other cases, it is weaker than second-order sufficient conditions.

We need the two sequences ε, χ to define our approximate second-order sufficient point. If one uses ε to define the almost active constraints, and if all constraints that are active at $\tilde{x}^* = (x^*, y^*, w^*, z^*, \zeta_1^*, \zeta_2^*)$ are infeasible at $\tilde{x} = (x, y, w, z, \zeta_1, \zeta_2)$, and if ε is too small, then no constraint will be ε active and the second condition of our definition becomes too strong.

We denote by $A(\chi, \tilde{x})$ the matrix formed by the gradients of the active bound constraints and χ active nonbound constraints of (OMPV(c)). The second condition of the above definition can be verified in the process—commonly encountered in active-set methods—of solving a linear system with the following matrix:

$$\begin{bmatrix} \Lambda_{xx}(\tilde{x}, \tilde{\lambda}) & A^T(\chi, \tilde{x}) \\ A(\chi, \tilde{x}) & 0 \end{bmatrix}.$$

Special versions of the symmetric indefinite factorization will reveal whether this matrix has the inertia that is compatible with the second-order condition [5].

It thus seems possible to define an active-set approach that, for given $\epsilon > 0$ and $\chi > 0$, determines an $(\epsilon,\ \chi)$ second-order stationary point. Nevertheless, we are not aware at this time of any software package that is guaranteed to provide such points. Moreover, our definition is unlikely to work for algorithms such as interior point approaches. We are currently investigating (1) ways of defining alternate approximate second-order stationary points that can accommodate approaches other than active-set approaches, and (2) whether currently used active-set approaches satisfy our assumptions.

Theorem 3.1. *Assume that the problem (OMPV) satisfies **A1**, **A2**, and **A3** and that every instance of the problem (OMPV(c^n)) is solved with an algorithm that satisfies **Alg1**. Assume that $\tilde{x}^n = (x^n, y^n, w^n, z^n, \zeta_1^n, \zeta_2^n)$ is an (ε^n, χ^n) second-order stationary point of (OMPV(c^n)) $\forall\, n = 1, 2, \ldots, \infty$ and for sequences $\{c^n\}, \{\varepsilon^n\}, \{\chi^n\}$ that satisfy $\lim_{n\to\infty} c^n = \infty, \lim_{n\to\infty} \varepsilon^n = 0, \lim_{n\to\infty} \chi^n = 0$, and $\lim_{n\to\infty} c^n \varepsilon^n = 0$.*

Let $(x^*, y^*, w^*, z^*, \zeta_1^*, \zeta_2^*)$ *be an accumulation point of this sequence. If* (x^*, y^*, z^*, w^*) *satisfies MPCC-LICQ, then* (x^*, y^*, w^*, z^*) *must be an M-stationary point of (OMPV).*

Proof. The argument from [24, Thm. 3.3] that was used to prove the similar result for the case when $\epsilon^n = 0$, $\chi^n = 0$, applies here. We outline the main elements and the way the argument continues to apply for approximate stationary points.

The limit point of the sequence $\tilde{x}^n = (x^n, y^n, w^n, z^n, \zeta_1^n, \zeta_2^n)$ is $(x^*, y^*, w^*, z^*, \zeta_1^*, \zeta_2^*)$. Using Theorem 2.5, we obtain that $0 = \zeta_1^* = \zeta_2^*$ and that (x^*, y^*, w^*, z^*), with the associated MPCC multiplier, is a C-stationary point. Note that since MPCC-LICQ holds, the MPCC multiplier must be unique. Assume now that the limit point $(x^*, y^*, w^*, z^*, 0, 0)$ is not an M-stationary point. This means that, for the unique MPCC multiplier $\left(\lambda^*, \mu^*, \theta^*, \widehat{\eta}_y^*, \widehat{\eta}_w^*\right)$, there exists an index k^C, $k^C \in \{1, 2, \ldots, n_c\}$ such that $\widehat{\eta}^*_{w,k^C} < 0$ and $\widehat{\eta}^*_{y,k^C} < 0$.

If we follow the logic we went through to obtain (2.10), starting from (2.5), as well as the definition of an ε stationary point, this means that for any n there exists a multiplier $\widetilde{\lambda}^n = \left(\lambda^n, \mu^n, \theta^{+n}, \theta^{-n}, \eta_y^n, \eta_w^n, \alpha_c^n, \alpha_1^n, \alpha_2^n\right)$ of (OMPV(c^n)) such that $y^n_{k^C} < 0, w^n_{k^C} < 0$, $\eta^n_{y,k^C} = 0$, $\eta^n_{w,k^C} = 0$, $\zeta_2^n > 0$, $\alpha_2^n = 0$. Therefore, the bound constraints $y_{k^C} \leqslant 0, w_{k^C} \leqslant 0$ *must be inactive* at the current n, and we must also have that $\alpha_c^n = c^n + t^n_{\alpha 2} = c^n + O(\varepsilon^n)$. Moreover, any nonbound constraint of (OMPV(c^n)) that is χ^n active must be a relaxed active constraint of (OMPV) at (x^*, y^*, w^*, z^*). In addition, following again the proof of Theorem 2.5, used to obtain (2.10), starting from (2.5), we obtain that the MPCC multipliers corresponding to the index k^C must satisfy $\widehat{\eta}^*_{y,k^C} = \lim_{n\to\infty} \alpha_c^n w^n_{k^c} < 0, \widehat{\eta}^*_{w,k^C} = \lim_{n\to\infty} \alpha_c^n y^n_{k^c} < 0$. In particular, this means that the sequence $\frac{w^n_{k^c}}{y^n_{k^c}}$ is lower bounded away from 0.

Now choose a vector $d^n = (d_x^n, d_y^n, d_w^n, d_z^n)$ that satisfies the following constraints:

$$\begin{array}{lcll}
\nabla_x g_i(x^n) d_x^n & = & 0, & i : g_i(x^*) = 0, \\
\nabla_x h(x^n) d_x^n & = & 0, & \\
\nabla_{(x,y,w,z)} F(x^n, y^n, w^n, z^n) \left(d_x^n, d_y^n, d_w^n, d_z^n\right)^T & = & 0, & \\
d^n_{y,k} & = & 0, & k : y_k^* = 0,\ k \neq k^C, \\
d^n_{w,k} & = & 0, & k : w_k^* = 0,\ k \neq k^C, \\
d^n_{y,k^C} & = & 1, & \\
d^n_{w,k^C} & = & -\frac{w^n_{k^C}}{y^n_{k^C}}. &
\end{array}$$

Since the limit point (x^*, y^*, w^*, z^*) satisfies MPCC-LICQ, a vector $d^n = (d_x^n, d_y^n, d_w^n, d_z^n)$ must exist for n sufficiently large. In addition, from (a) our observation that any nonbound constraint of (OMPV(c^n)) that is χ^n active must be a relaxed active constraint of (OMPV) at (x^*, y^*, w^*, z^*), and (b) since the constraints $y_{k^C} \leqslant 0$, $w_{k^C} \leqslant 0$ are inactive at $\tilde{x}^n = (x^n, y^n, w^n, z^n, \zeta_1^n, \zeta_2^n)$, it follows that $\tilde{d}^n = (d_x^n, d_y^n, d_w^n, d_z^n, 0, 0)$ is in the null space of the gradients of the active bound constraints and χ^n active nonbound constraints. Consider the Lagrangian of (OMPV(c)),

$$\begin{array}{rcl}
\Lambda^c(\tilde{x}, \tilde{\lambda}) & = & f(x, y, w, z) + g(x)^T \lambda + h(x)^T \mu + F(x, y, w, z)^T (\theta^+ - \theta^-) + y^T \eta_y \\
& + & w^T \eta_w + \alpha_c y^T w + (c - e^T_{n_c+l} \theta^+ - e^T_{n_c+l} \theta^- - \alpha_1)\zeta_1 + (c - \alpha_c - \alpha_2)\zeta_2.
\end{array}$$

Following the assumption that $\tilde{x}^n = (x^n, y^n, w^n, z^n, \zeta_1^n, \zeta_2^n)$, together with the multiplier $\widetilde{\lambda}^n = \left(\lambda^n, \mu^n, \theta^{+n}, \theta^{-n}, \eta_y^n, \eta_w^n, \alpha_c^n, \alpha_1^n, \alpha_2^n\right)$, is an (ε^n, χ^n) second-order stationary point, we must have that $\tilde{d}^{n,T} \nabla_{\tilde{x}\tilde{x}} \Lambda^{c^n}(\tilde{x}^n, \tilde{\lambda}^n) \tilde{d}^n \geqslant 0$.

Following the expression of the Lagrangian of (OMPV(c)) and the definition of d^n, and since $\lambda^n, \mu^n, \theta^{+,n}, \theta^{-,n}$ are bounded, we obtain that

$$\tilde{d}^{n,T} \nabla_{\tilde{x}\tilde{x}} \Lambda^{c^n}(\tilde{x}^n, \tilde{\lambda}^n) \tilde{d}^n = -\alpha_c^n \frac{w_{k^c}^n}{y_{k^c}^n} + O(1).$$

However, as we argued in the beginning of this proof, the fraction in the previous equation is lower bounded away from 0, whereas we have that $\alpha_c^n \to \infty$, which means that $\limsup \tilde{d}^{n,T} \nabla_{\tilde{x}\tilde{x}} \Lambda^{c^n}(\tilde{x}^n, \tilde{\lambda}^n) \tilde{d}^n < 0$.

This contradicts the assumption that $\tilde{x}^n$ is an (ε^n, χ^n) second-order stationary point, which in turn contradicts our assumption that (x^*, y^*, w^*, z^*) is not an M-stationary point. Therefore, (x^*, y^*, w^*, z^*) is an M-stationary point, and the proof is complete. □

An important question is whether the result can be improved to show convergence to a strongly stationary point when MPCC-LICQ holds. The M-stationarity result can be enhanced when ULSC holds at the convergence point. The following theorem goes back to a result from [13] for a smoothing method and has also been stated in [24] for the regularization method, in which each subproblem is solved exactly (though, in practice, this may take an infinite number of steps). We formally state this result in the following.

THEOREM 3.2. *If, in addition to the assumptions of Theorem* 3.1*, we have that ULSC holds at the accumulation point* (x^*, y^*, w^*, z^*)*, then* (x^*, y^*, w^*, z^*) *is a strongly stationary point and, as a result, a B-stationary point.*

Proof. Since MPCC-LICQ holds at the solution, there exists a unique MPCC multiplier that satisfies the M-stationarity conditions. From the uniqueness property, the same multiplier must also satisfy ULSC, which, together with the M-stationarity conditions, implies that (x^*, y^*, w^*, z^*) is a strongly stationary point of (OMPV). Following [23, Thm. 4], we get that the point must also be a B-stationary point. □

3.1. M-stationary points in finite arithmetic.

3.1. M-stationary points in finite arithmetic. In this subsection we discuss whether M-stationary points and strongly stationary points can be distinguished in finite arithmetic.

The following result is in the vein of backward error analysis, and it shows that in any neighborhood of an M-stationary point there is a strongly stationary point of a perturbed problem. The result is stated for (OMPV), though it can be immediately extended to the entire MPCC class. Note that we use no other property of (OMPV) except the twice continuous differentiability of the data.

THEOREM 3.3. *Assume that* (x^*, y^*, w^*, z^*) *is an M-stationary point of (OMPV). Then, for any* $\delta > 0$*, the following exist:*

1. *A perturbation* $f^\delta(x, y, w, z)$*of the objective function* $f(x, y, w, z)$ *that satisfies* $\left\|\nabla_{\tilde{x}} f^\delta(x, y, w, z) - \nabla_{\tilde{x}} f(x, y, w, z)\right\| \leqslant \delta \, \forall \, \tilde{x} = (x, y, w, z)$ *in a neighborhood of* (x^*, y^*, w^*, z^*)*.*
2. *A vector* l_F^δ *that satisfies* $\left\|l_F^\delta\right\| \leqslant \delta$*.*
3. *A point* $(x^\delta, y^\delta, w^\delta, z^\delta)$ *that satisfies* $\left\|(x^\delta, y^\delta, w^\delta, z^\delta) - (x^*, y^*, w^*, z^*)\right\| \leqslant \delta$ *and that is a strongly stationary point for the perturbed problem*

$$(\delta OMPV) \qquad \begin{array}{lll} \displaystyle\min_{x,y,w,z} & f^\delta(x, y, w, z) & \\ s.t. & g(x) & \leqslant 0, \\ & h(x) & = 0, \\ & F(x, y, w, z) & = l_F^\delta, \\ & y, w & \leqslant 0, \\ & (y^T w = 0) \;\; y^T w & \leqslant 0. \end{array}$$

Note. A pure backward error result would require that the M-stationary point at hand be the strongly stationary point of a nearby problem, which is generally not true because of the structure of the problem.

Proof. By the definition of an M-stationary point, it follows that there exists an MPCC multiplier $(\lambda, \mu \geqslant 0, \theta, \widehat{\eta}_y, \widehat{\eta}_w)$ at $\tilde{x} = (x^*, y^*, w^*, z^*)$ that satisfies

$$\begin{array}{lcl}
\nabla_x f(x^*, y^*, w^*, z^*)^T + \nabla_x h(x^*)^T \lambda & & \\
\qquad + \nabla_x g(x^*)^T \mu + \nabla_x F(x^*, y^*, w^*, z^*)^T \theta & = & 0, \\
\nabla_y f(x^*, y^*, w^*, z^*)^T + \hat{\eta}_y + \nabla_y F(x^*, y^*, w^*, z^*)^T \theta & = & 0, \\
\nabla_w f(x^*, y^*, w^*, z^*)^T + \hat{\eta}_w + \nabla_w F(x^*, y^*, w^*, z^*)^T \theta & = & 0, \\
\nabla_z f(x^*, y^*, w^*, z^*)^T + \nabla_z F(x^*, y^*, w^*, z^*)^T \theta & = & 0, \\
g(x^*) \leqslant 0, \ h(x^*) = 0, \ F(x, y, w, z) & = & 0, \\
\quad \mu \geqslant 0, \ y \leqslant 0, \ w & \leqslant & 0, \\
y^T w = 0, \ g(x)^T \mu = 0, \sum_{k=1}^{n_c} y_k \, |\widehat{\eta}_{y,k}| = 0, \ \ \sum_{k=1}^{n_c} w_k \, |\widehat{\eta}_{w,k}| & = & 0,
\end{array}$$

in addition to the sign condition on the multipliers $\widehat{\eta}_y, \widehat{\eta}_w$ associated with the variables involved in the complementarity constraints. These conditions are

$$\forall k = 1, 2, \ldots, n_C, \quad \widehat{\eta}_{y,k} \widehat{\eta}_{w,k} \geqslant 0 \text{ and } \left\{ \begin{array}{l} \widehat{\eta}_{w,k} < 0 \Rightarrow \widehat{\eta}_{y,k} = 0, \\ \widehat{\eta}_{y,k} < 0 \Rightarrow \widehat{\eta}_{w,k} = 0. \end{array} \right.$$

To simplify the notation, we assume without loss of generality that the negative multipliers appear only in the y variables. In turn, this assumption implies that there is a partition $\tilde{K} \cup \tilde{K}^c = 1, 2, \ldots, n_C$ that satisfies

$$\begin{array}{l}
k \in \tilde{K} \Rightarrow \widehat{\eta}_{y,k} < 0, \widehat{\eta}_{w,k} = 0, \\
k \in \tilde{K}^c \Rightarrow \widehat{\eta}_{y,k} \geqslant 0, \widehat{\eta}_{w,k} \geqslant 0.
\end{array}$$

We now construct the family of points

$$\begin{array}{l}
\tilde{x}(t) = (x(t), y(t), w(t), z(t)), \\
x(t) = x^*, y(t) = y^*, z(t) = z^*, \\
w_k(t) = w_k^*, \ k \in \tilde{K}^c, \\
w_k(t) = w_k^* - t, \ k \in \tilde{K}.
\end{array}$$

We have that the point $\tilde{x}(t)$ satisfies

$$\begin{array}{rcl}
\nabla_x f(x^*, y^*, w(t), z^*)^T + \nabla_x h(x^*)^T \lambda \qquad\qquad & & \\
+ \nabla_x g(x^*)^T \mu + \nabla_x F(x^*, y^*, w(t), z^*)^T \theta & = & l_x(t), \\
\nabla_y f(x^*, y^*, w(t), z^*)^T + \widehat{\eta}_y + \nabla_y F(x^*, y^*, w(t), z^*)^T \theta & = & l_y(t), \\
\nabla_w f(x^*, y^*, w(t), z^*)^T + \widehat{\eta}_w + \nabla_w F(x^*, y^*, w(t), z^*)^T \theta & = & l_w(t), \\
\nabla_z f(x^*, y^*, w(t), z^*)^T + \nabla_z F(x^*, y^*, w(t), z^*)^T \theta & = & l_z(t), \\
g(x^*) \leqslant 0, h(x^*) = 0, \ F(x^*, y^*, w(t), z^*) \qquad\qquad\qquad & = & l_F(t), \\
\mu \geqslant 0, y^* \leqslant 0, \ \ w(t) \leqslant 0, \ y^{*,T} w(t) = 0, & & \\
g(x^*)^T \mu = 0, \sum_{k=1}^{n_c} y_k^* \, |\widehat{\eta}_{y,k}| = 0, \sum_{k=1}^{n_c} w_k^* \, |\widehat{\eta}_{w,k}| = 0. & &
\end{array}$$

Since from **A1** the data of (OMPV) are twice continuously differentiable, we have that there exists $c_l > 0$ depending only on the point (x^*, y^*, w^*, z^*) such that the residuals satisfy $\|l_x(t), l_y(t), l_w(t), l_z(t), l_F(t)\| \leqslant c_l t \, \forall \, t$ sufficiently small. There exists a t^δ such that, $\forall \, t \leqslant t^\delta$ we have, at the same time,

$$\|(x(t), y(t), w(t), z(t)) - (x^*, y^*, w^*, z^*)\| \leqslant \delta \text{ and } \|l_x(t), l_y(t), l_w(t), l_z(t), l_F(t)\| \leqslant \delta.$$

After defining $l_F^\delta = l_F(t^\delta)$, $f^\delta(x,y,w,z) = f(x,y,w,z) - x^T l_x(t^\delta) - y^T l_y(t^\delta) - w^T l_w(t^\delta) - z^T l_z(t^\delta)$, and $(x^\delta, y^\delta, w^\delta, z^\delta) = (x(t^\delta),\ y(t^\delta),\ w(t^\delta),\ z(t^\delta))$, the conclusion of the theorem follows. □

If the point toward which we are converging is an M-stationary point that satisfies MPCC-LICQ and that is not a strongly stationary point, then a descent direction can be found [14]. In that sense, Theorem 3.1 is still weaker than the ideal result, which is that if MPCC-LICQ holds at the point toward which we are converging, then that point is strongly stationary and a B-stationary [14, 23].

However, the preceding theorem shows that, in finite arithmetic, one may not be able use only the signs of the multipliers to predict whether we are converging to a strongly stationary point $(x^\delta, y^\delta, w^\delta, z^\delta)$ or to a proper M-stationary point (x^*, y^*, w^*,z^*), where descent is still possible. This point will be demonstrated later with a numerical example. To guarantee that one can escape a proper M-stationary point, at least when MPCC-LICQ holds, one has to combine a nonlinear programming algorithm with an active-set method of the type studied in [14]. How to robustly switch between the two is the subject of future research.

4. Conditions for global convergence. To obtain a global convergence result, we need to ensure that the iterates do not drift away to ∞. To achieve such a result, we make two more assumptions about the problem and one about the algorithm used to solve the relaxed problem (OMPV(c)).

A4. The penalty function $\psi(x,y,w,z) = ||F(x,y,w,z)||_\infty + y^T w$ has bounded level sets over the set defined by the constraints $g(x) \le 0,\ h(x) = 0,\ y \le 0,\ w \le 0$.

A5. The objective function $f(x,y,w,z)$ is bounded below over the same set.

Alg2. For any fixed value of c, the algorithm applied for solving the problem (OMPV(c)) decreases the merit function $f(x,y,w,z) + c\psi(x,y,w,z)$.

Assumption **Alg2** is quite natural in connection with the subproblem (OMPV(c)). If the constraints $g(x)$ and $h(x)$ are linear, and if the algorithm applied is an SQP algorithm that uses a positive definite matrix in the quadratic program (from a BFGS-type approximation, for example), then one can show that for a fixed c the SQP algorithm produces a sequence of decreasing values of $f(x,y,w,z) + c\psi(x,y,w,z)$, much as in the case of the L_∞ penalty function [3]. Assumption **A5** is standard in most global convergence results [8]. Assumption **A4**, however, seems to be quite restrictive for general nonlinear programming, unless the feasible set is compact. Nevertheless, we will show that, for the obstacle problem presented in section 5, the assumption does hold. A similar condition was used to enforce boundedness of the iterates in [18], for a different merit function, that did not enjoy the exactness property and could not lead to the outcome of part 1 of Theorem 2.7 and can be expected to have a bounded feasible set, which is the first prerequisite for **A4** to hold. So for the case of MPCC, **A4** is not overly restrictive.

THEOREM 4.1. *Assume that (OMPV) satisfies* ***A*1–*A*5** *and that the algorithm used to solve the subproblems satisfies* ***Alg*1** *and* ***Alg*2**. *Then the solution sequence produced by the algorithm in Table* 2.1 *is bounded, and any accumulation point is a C-stationary point.*

Proof. Let B_f denote the lower bound of the objective function $f(x,y,w,z)$, which exists from **A5**. It then follows that the merit function $\Psi(x,y,w,z,c) = \frac{1}{c}\left(f(x,y,w,z) - B_f\right) + \psi(x,y,w,z)$ is decreased at any step of the algorithm in Table 2.1. When c is fixed, the decrease follows from **Alg2**. When c is increased, $\Psi(x,y,w,z,c)$ must decrease since, at that point, $f(x,y,w,z) - B_f > 0$. Therefore, all the iterates (x^n, y^n, w^n, z^n) will satisfy $\psi(x^n, y^n, w^n, z^n) < \Psi(x^n,y^n,w^n,\ z^n,c^n) \le \Psi(x^0, y^0, w^0, z^0, c^0)$. The conclusion follows from **A4** and Theorem 2.5. □

We emphasize that the value of the lower bound B_f does not need to be known for the decrease in the (unknown) merit function $\Psi(x, y, w, z)$ to occur.

We note that global convergence results for methods that use a penalty term are generally restricted to the case when the global solution of the subproblem is obtained [8, Thm. 12.1.1]. In our case, local solutions of the relaxed/penalized subproblems under the assumptions described here are sufficient.

5. The obstacle problem. As examples of problems that satisfy these conclusions, we present several instances of the obstacle problem from [22]. This problem concerns the optimization of an elastic membrane in contact with a rigid or elastic obstacle. The design parameters quantify the shape of support of the membrane. The discretized problem is the following:

$$\text{(OBST)}\qquad \begin{array}{lll} \displaystyle\min_{x,y,w,z} & f(x,z) & \\ \text{s.t.} & g(x) & \leqslant 0, \\ & -A(x)z + \phi(x) & = y, \\ & k(\phi(x) - A(x)z) + \chi(x) - z & = w, \\ & y, w & \leqslant 0, \\ & (y^T w = 0)\;\; y^T w & \leqslant 0. \end{array}$$

Here all functions are differentiable with respect to their parameters. In addition, for any x we have that the matrix $A(x)$, which originates in the discretization of an elliptic operator, is positive definite. The parameter k satisfies $k \geqslant 0$ and, if $k = 0$, then the obstacle is rigid. The case $k > 0$ models the situation in which the obstacle is flexible. The inequality constraints $g(x) \leqslant 0$ are box constraints on the design parameters x.

LEMMA 5.1. *The problem (OBST) satisfies* ***A1****,* ***A2****, and* ***A3****.*

Proof. Assumption **A1** is satisfied immediately from the differentiability properties of all functions involved in the definition of the obstacle problem. The parameters x satisfy box constraints, and therefore the functions $g(x)$ are linear and always satisfy **A2**.

To verify **A3** for (OBST) in the (OMPV) framework, we have

$$\begin{array}{rcl} F(x,y,w,z) & = & \left(\begin{array}{l} y + A(x)z - \phi(x), \\ w - k(\phi(x) - A(x)z) - \chi(x) + z \end{array} \right) \\ \nabla_{(y,w,z)} F(x,y,w,z) & = & \left(\begin{array}{ccc} I & 0 & A(x), \\ 0 & I & I + kA(x) \end{array} \right). \end{array}$$

We prove that the partition of $\nabla_{(y,w,z)} F(x,y,w,z)$ into blocks corresponding to the variables y, w, z is a mixed P partition, and thus **A3** is satisfied. Take a vector $(\bar{y}, \bar{w}, \bar{z})^T$ that satisfies $\nabla_{(y,w,z)} F(x,y,w,z)\,(\bar{y}, \bar{w}, \bar{z})^T = 0$, that is,

$$\bar{y} + A(x)\bar{z} = 0, \quad \bar{w} + (I + kA(x))\bar{z} = 0,$$

as well as $\bar{y}_k \bar{w}_k \leqslant 0,\; k = 1, 2, \ldots, n_c$. this implies that $\bar{y}^T \bar{w} \leqslant 0$. Solving for $\bar{y}, \bar{w}$ from the displayed equations, we see that this implies $\bar{z}^T A(x)^T (I + kA(x))\bar{z} \leqslant 0$ which, in turn, implies $\bar{z}^T A(x)^T \bar{z} + k\bar{z}^T A(x)^T A(x)\bar{z} \leqslant 0$. Since the matrix $A(x)$ is positive definite and $k \geqslant 0$, we obtain $\bar{z} = 0$, and subsequently $\bar{y} = 0$ and $\bar{w} = 0$. This proves that **A3** holds for (OBST) and completes the proof. □

Concerning the level sets of $\psi(x)$, we have the following lemma.

LEMMA 5.2. *For any $\beta > 0$, we have that the set*

$$\mathcal{L}_\beta = \{(x, y, w, z) \in \mathbb{R}^{n+2n_c+l} | y \le 0,\ w \le 0,\ g(x) \le 0,\ \psi(x, y, w, z) < \beta\}$$

is bounded. Therefore, problem (OBST) satisfies ***A*****4**.

Proof. We start by assuming that the conclusion is false. This means there is a $\beta > 0$ for which the level set $\mathcal{L}_\beta$ is unbounded.

We first note that we have box constraints on the variables x, which means that they can never be unbounded. Therefore, only the (y, w, z) part can be unbounded. Thus, there exists a sequence $\tilde{x}^n = (x^n, y^n, w^n, z^n)$ such that $\Gamma^n = ||(y^n, w^n, z^n)|| \to \infty$ and $\psi(x^n, y^n, w^n, z^n) < \beta$. In turn, the last statement implies

$$||F(x^n, y^n, w^n, z^n)||_\infty \le \beta, \quad {y^n}^T w^n \le \beta. \tag{5.1}$$

Since the sequence

$$\widehat{x}^n = \left(x^n, \frac{y^n}{\Gamma^n}, \frac{w^n}{\Gamma^n}, \frac{z^n}{\Gamma^n}\right)$$

is bounded, it admits a convergent subsequence. To simplify notation, we assume that the whole sequence $\widehat{x}^n$ is convergent to a point $(\overline{x}, \overline{y}, \overline{w}, \overline{z})$, satisfying $||(\overline{y}, \overline{w}, \overline{z})|| = 1$. We divide the first equation in (5.1) by Γ^n, the second equation by $(\Gamma^n)^2$, and we take the limit in both as $n \to \infty$. Since $F(x, y, w, z)$ is linear in y, w, z, and since the mappings $\chi(\cdot)$ and $\phi(\cdot)$ are continuous, we obtain that the limit point $(\overline{x}, \overline{y}, \overline{w}, \overline{z})$ satisfies the relations

$$\overline{y} + A(\overline{x})\overline{z} = 0, \quad \overline{w} + kA(x)\overline{z} + \overline{z} = 0, \quad \overline{w}^T\overline{y} = 0. \tag{5.2}$$

Solving for $\overline{y}$ and $\overline{w}$ from the first two equations, and replacing the results in the third equation, we obtained $\overline{z}^T A(\overline{x})^T (kA(x) + I)\overline{z} = 0$. In turn, the fact that $A(\overline{x})$ is a positive definite matrix implies $\overline{z} = 0$. Subsequently, from (5.2) we obtain $\overline{w} = \overline{y} = 0$. This contradicts $||(\overline{y}, \overline{w}, \overline{z})|| = 1$ and proves the claim. □

We are now ready to state our main result for the obstacle problem.

THEOREM 5.3. *Assume that $f(x, z)$, the objective function of (OBST) is bounded below on the set $g(x) \le 0$, $y \le 0$, $w \le 0$. If the algorithm from Table* 2.1 *is applied to (OBST) and the (OMPV(c)) subproblems are solved with an algorithm that satisfies* ***Alg*****1** *and* ***Alg*****2**, *then the sequence of iterates is bounded and any accumulation point is a C-stationary point.*

Proof. The result follows from Lemmas 5.1 and 5.2 and Theorem 4.1. □

6. Numerical results.

In this section we apply an algorithm like the one in Table 2.1 to three instances of the obstacle problem. All problems have an objective function that is nonnegative over the set defined by the parameter constraints and the bound constraints on the variables y, w. Therefore Theorem 5.3 applies and, if the algorithm we use satisfies **Alg1** and **Alg2**, then any accumulation point of the algorithm is a C-stationary point (and, from Theorem 2.7, even a strongly stationary point if the penalty parameter remains bounded).

In their original form, the problems are similar to (OMPV) except that the constraints $y, w \le 0$ are replaced by $y, w \ge 0$ [22]. One can immediately see that all of the results in the preceding sections apply if we change the signs of the variables y, w and correspondingly switch the signs of the multipliers η_w, η_y, $\widehat{\eta}_y$, and $\widehat{\eta}_w$. In all problems, we deal with an elastic membrane hanging over an obstacle. The membrane is attached to a support whose shape can change as a part of the optimization process.

- **The incidence set identification problem [22, sect. 9.4]**. In this problem, the shape of the support must be changed so that the shape of the contact region is as close as possible to a prescribed shape. The objective function here is the discrepancy between the current contact region and the sought after contact region. Therefore, in this problem the final objective function should be as close as possible to zero. In Table 6.1, instances of this problem are the `is` problems, which in reference [9] are called the `incid-set` problems.
- **The packaging problem with compliant obstacle [22, sect. 9.3]**. In this problem we try to find the shape of the support that will minimize the area of the membrane, while keeping the membrane in contact with the obstacle over at least a prescribed region. The obstacle here is compliant (it can deform under pressure from the membrane). The objective function is the area of the membrane. In Table 6.1, instances of this problem are the `pc` problems, which in reference [9] are the `pack-comp` problems.
- **The packaging problem with rigid obstacle [22, sect. 9.2]**. This is the same as the preceding problem except that now the obstacle is constrained to be rigid. In Table 6.1, instances of this problem are the `pr` problems, which in reference [9] are the `pack-rig` problems. The shape of the optimal membrane for the problem `pr-2-32` is displayed in Figure 6.1 both in a transparent fashion on top of the parabolic obstacle and by itself with the final mesh projected on the bottom plane.

The additional constraints of the problems that would not fit within **A3** are treated by means of a penalty function. We emphasize that this was not a choice we made in order to have the problems fit our framework. The use of a penalty function was the modeling choice from [22], and it was necessary there for the computation of the generalized gradient. Therefore, the problems solved here have the same formulation as [22].

For each problem we have six variants. We consider three different grid sizes, all related to a finite element discretization: 8 × 8, 16 × 16, and 32 × 32. The names of the associated problems contain `8`, `16`, or `32`. We also consider two types of obstacles. The first obstacle is linear [22, Ex. 9.1]. The corresponding problems have in their names the digit `1`. The second obstacle is parabolic [22, Ex. 9.2]. The corresponding problems have in their names the digit `2`.

The problems have been modeled using the AMPL modeling language [12], starting from the AMPL model files from the MacMPEC library of Sven Leyffer [9, 17]. We have implemented the algorithm in Table 2.1 as an AMPL script. We chose the following parameters: $q = 2$, $K = 10$, $c_0 = 10$, and $\epsilon^n = 10^{-3}12^{-n}$. Note that $c^n \leqslant 10^{n+1}$, which means that $c^n\epsilon^n \to 0$, as required by our results. In addition, we stop the algorithm in Table 2.1 when $\zeta_1^n + \zeta_2^n \leq 1e - 7$.

To solve the nonlinear programming problem for fixed penalty parameter c, which corresponds to the section following the label **OMPV** in Table 2.1, we have used the interior point solver `knitro` [21]. To produce an ϵ^n stationary point, as required in Table 2.1, we have set parameters `opttol=feastol=`ϵ^n. We have set the maximum number of iterations of `knitro` to 4000.

Does `knitro` satisfy our assumptions? Since `knitro` is an interior point algorithm, it satisfies the bound constraints in the definition of an ϵ stationary point exactly. We note that `knitro` satisfies a weaker version of the property **Alg1**, where MFCQ is replaced by LICQ [21]. Note, however, that **Alg1** is merely a way to ensure that an approximate KKT point of (OMPV(c)) can be found. In all our experiments, knitro always returned an ϵ stationary point with a prescribed $\epsilon > 0$. We cannot a priori guarantee that **Alg2** holds for knitro for any problem, because the algorithm uses a completely different technique to approach the optimal point from that used by SQP algorithms with an exact penalty function, for which **Alg2** can be shown to hold [3]. Nevertheless, **Alg2** did hold for the examples we have tried (at least with respect to the first and last iterates for a fixed penalty parameter c). As a result, Theorem 5.3 applies to give global convergence to C-stationary points.

The problem of verifying the assumptions of Theorem 3.1, by either an a priori guarantee or an a posteriori test, is more difficult. We need to guarantee that the outcome of a given algorithm satisfies the approximate second-order conditions. Our definition of second-order stationary points is oriented toward active-set methods and cannot be guaranteed for knitro. Unfortunately, it also involves the use of derivatives that are not interactively provided in AMPL, and we could not test for it, or for a variant of it that may have been appropriate for interior point methods. In addition, in order to guarantee that the assumptions of Theorem 3.1 hold, we need to verify that MPCC-LICQ holds, which is also difficult to do while using AMPL. Therefore, the only test that we performed on the outcome, with respect to convergence to M-stationary points, was to see whether the solution point and multipliers at hand satisfy the M-stationarity condition.

For the following reasons, we did not choose an active-set software that was available to us to solve the subproblem (OMPV(c))—though it would seem appropriate from the discussion following the definition of (ϵ, χ) second-order stationary points:

(1) Our attempts to solve the subproblems (OMPV(c)) by either lancelot or minos were unsuccessful on problems for grids of size 16 and above (at least in a reasonable amount of time).

(2) For the package SNOPT it has already been demonstrated that its elastic-mode approach works on problems such as the one described here [9]. Similarly, the package FilterSQP was proved to be efficient on problems like the one in this work, due to the incorporation of a feasibility restoration phase.

(3) As far as we understand from the information available on the NEOS server [20], neither the elastic mode of SNOPT nor the feasibility restoration of FilterSQP can be turned off, at least not through the AMPL interface that we have used to run our models. As a result, it did not seem likely that we would be able to evaluate the benefit of our approach when used in conjunction with an active-set algorithm.

We study whether we are approaching either a C-stationary point or an M-stationary point. Following the proof of Theorem 2.5, once we have obtained the Lagrange multipliers η_w, η_y of the constraints $y, w \leq 0$ in (OMPV(c)), we can construct the following approximation to the MPCC multipliers $\widehat{\eta}_w$, $\widehat{\eta}_y$:

$$\widehat{\eta}_{w,k} = \eta_{w,k} + cy_k, \quad \widehat{\eta}_{y,k} = \eta_{y,k} + cw_k, \quad k = 1, 2, \ldots, n_C.$$

This approximation is based on (2.5) and appears in the proof of the C-stationary part of Theorem 2.5.

We now define the parameters

$$\text{Cstat} = \min_{k=1,2,\ldots,n_C} \{\widehat{\eta}_{w,k}\widehat{\eta}_{y,k}\}, \quad \text{Mstat} = \max_{k=1,2,\ldots,n_C} \min\{\widehat{\eta}_{w,k}, \widehat{\eta}_{y,k}\}.$$

To preserve the clarity of the presentation, we ignore as this point the superscript ^{n}. Following the proof of Theorem 2.5, we get that, if $c \to \infty$ and $\liminf \text{Cstat} \geq 0$,

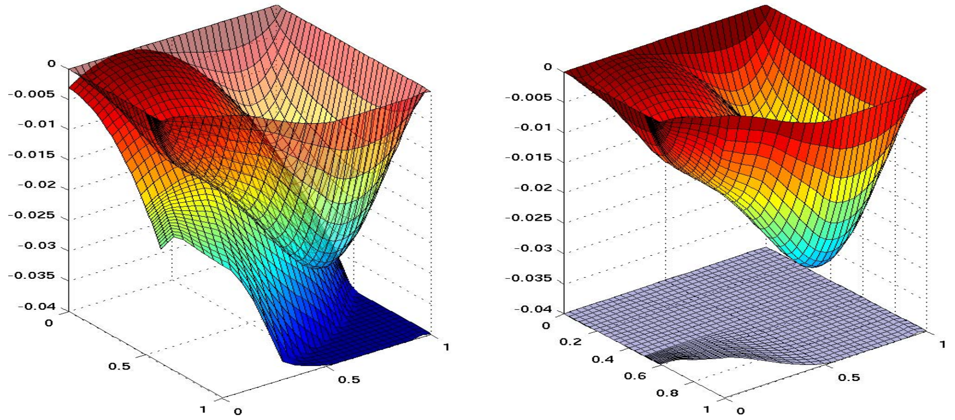


Fig. 6.1. *Solution of the obstacle problem with a rigid parabolic obstacle on a* 32×32 *mesh.*

Table 6.1
Numerical results.

Problem	Obj	Uc	Ut	Cstat	Mstat	Feval	KFeval
is-1-8	2.352e-08	0	5	4.10e-11	2.89e-09	204	390
is-1-16	8.639e-06	1	6	9.38e-08	7.85e-06	451	4001
is-1-32	5.904e-06	2	7	3.36e-08	5.52e-05	2906	1097
is-2-8	4.517e-03	1	6	5.12e-08	2.84e-07	302	1712
is-2-16	3.006e-03	1	6	1.27e-06	1.02e-03	434	4001
is-2-32	1.774e-03	2	5	1.01e-05	3.54e-03	2083	4001
pc-1-8	6.000e-01	1	5	6.32e-14	1.40e-03	75	4001
pc-1-16	6.169e-01	1	7	3.82e-21	5.65e-07	297	4001
pc-1-32	6.529e-01	2	6	9.60e-18	8.93e-05	4999	3081
pc-2-8	6.731e-01	1	5	1.01e-19	3.03e-06	78	1421
pc-2-16	7.271e-01	2	5	3.60e-16	1.77e-03	289	1358
pc-2-32	7.826e-01	2	6	1.84e-16	1.22e-04	645	1350
pr-1-8	7.879e-01	1	6	9.28e-18	1.03e-06	193	81
pr-1-16	8.260e-01	2	5	1.68e-16	1.14e-05	218	54
pr-1-32	8.508e-01	2	5	1.95e-17	1.17e-03	644	3040
pr-2-8	7.804e-01	1	6	3.20e-18	1.46e-06	183	33
pr-2-16	1.085e+00	3	6	2.32e-15	1.73e-05	342	208
pr-2-32	1.135e+00	3	6	1.36e-14	1.59e-04	661	2792

then any accumulation point of the algorithm in Table 2.1 is a C-stationary point. In addition, when (OMPV) is formulated with the constraints $y, w \geq 0$ (as we have done in our AMPL files), the M-stationarity condition becomes $\widehat{\eta}_{w,k}\widehat{\eta}_{y,k} \geq 0$ (C-stationarity) and $\min\{\widehat{\eta}_{w,k}, \widehat{\eta}_{y,k}\} \geq 0$ for $k = 1, 2, \ldots, n_c$. As a consequence, when $\limsup \mathrm{Mstat} \leq 0$, the limiting point is an M-stationary point.

The numerical results are organized in Table 6.1. We have displayed the name of the problem (Problem), the final values of the objective function (Obj), the number of penalty updates (Uc), the number of tolerance updates (Ut), the C-stationarity indicator (Cstat), the M-stationarity indicator (Mstat), the number of function evaluations (Feval) needed by our implementation of the algorithm in Table 2.1, and the number of function evaluations (KFeval) needed by knitro to solve (OMPV) directly. The final tolerance parameter is $\epsilon = 10^{-3}12^{-Ut}$ and the final penalty parameter is $c = 10^{-Uc-1}$. When knitro was used to solve the problems directly, it was used with the default tolerance options opttol=feastol=1e-6.

From Table 6.1 it cannot be claimed that one solver is constantly better than the other (in terms of the number of function evaluations), though the elastic-mode approach solves 2/3 of the problems faster than `knitro`. We see that `knitro`, when applied directly to the chosen instances of the obstacle problem, stops with a maximum number of iterations reached in five instances, which is a sign of lack of convergence. Note that we went substantially beyond the 1,000 maximum iterations that represent the default settings of `knitro`. However, our elastic-mode approach leads to resolution of all the instances of these problems, as predicted by Theorem 4.1, although it also uses `knitro` in the inner loop! This conclusion is in line with previous work that has shown that the elastic-mode results in substantially more robust behavior of nonlinear programming solvers when applied to MPCC [1, 9].

In verifying the conclusion of Theorems 2.7 and 3.1 for the numerical outcome at hand, we have to decide whether $c \to \infty$. In absence of any additional information, such as whether the point toward which we are converging satisfies MPCC-LICQ and whether strict complementarity holds, which cannot be extracted from those solvers in a simple and robust manner, a robust test for this condition is difficult to design. On the one hand, we can consider the values of the penalty parameter derived from the value of `Uc` in Table 6.1, $c = 10^{-Uc-1}$, to be sufficiently small as to indicate that we have converged in all cases to strongly stationary points. This conclusion would be in line with the fact that, in a statistical sense, this is the expected outcome [23], as well as with preceding numerical investigations [9]. This outcome would confirm the first case of Theorem 2.7. We note that the values of the objective function are consistent with previous numerical experiments [9].

On the other hand, we see that Cstat $\geqslant 0$ in all cases, which validates the result of Theorem 2.5. The conclusion about M-stationarity is less solidly founded. We notice that Mstat is always less than 10^{-2}, most of the times less than 10^{-4}, and sometimes as small as 10^{-8}, whereas our stopping criterion required a feasibility of 10^{-7} in the relaxed constraints. Attempts at requiring more stringent stopping criteria resulted in failure or exceedingly long times of solving the subproblems. We cannot conclusively state that $\limsup$ Mstat is 0 (in the absence of a valid asymptotic test) though the evidence seems to be leaning that way, at least for some of the numerical experiments. We plan to study in the future the connection between such M-stationarity indicators and the relevant tolerances.

As a side note, it is perhaps a little bit surprising that all the final values of Mstat are positive, whereas negative values are what we are looking for in order to ascertain M-stationarity. In any case, such an outcome does not contradict either of our theoretical results.

On the issue of distinguishing between M-stationary points and strongly stationary points in finite arithmetic, we have observed what was predicted by Theorem 3.3. For example, for problem `pr-1-16` formulated with $y \geq 0$ and $w \geq 0$, the numerical results for index $k = 19$ were $y_{19} = 1.039e - 05$, $w_{19} = 1.42e - 04$, $\widehat{\eta}_{y,19} = 0.14$, $\widehat{\eta}_{w,19} = 1.03e-02$. In absence of any additional information (such as whether MPCC-LICQ holds, which cannot be tested for in AMPL), it is difficult to decide whether the algorithm converges to an M-stationary point, at which descent is still possible, or whether it converges to a strongly stationary point. The same situation, in general, was noticed for all problems.

7. Conclusions. We have shown that any accumulation point of the elastic-mode approach that solves subproblems inexactly, but with increasing accuracy, is a C-stationary point of an optimization problem of parameterized mixed P variational inequalities (Theorem 2.5). If, in addition, the accumulation point satisfies MPCC-

LICQ and the solver used provides a point that approximately satisfies the second-order conditions, then the resulting point is M-stationary (Theorem 3.1). We have also shown that any M-stationary point of such problems is in the neighborhood of a strongly stationary point of a perturbed problem for arbitrarily small perturbations (Theorem 3.3). In practical terms, this means that strongly stationary points and M-stationary points are difficult to distinguish in finite arithmetic.

In the process of guaranteeing that the iterates do not drift to infinity, we construct a merit function with bounded level sets that is decreased at every step, even when the penalty parameter is updated (Theorem 4.1). In turn, this ensures the boundedness of the iterates of the algorithm. We have shown that optimization problems built around the obstacle problem [22, sect. 9] satisfy the problem assumptions **A1**–**A5** that we have used in proving our convergence results (Theorem 5.3). We have implemented our algorithm and applied it to 18 instances of the obstacle problem from the MacMPEC [17] library. The numerical results demonstrate our theoretical findings as well as the significant increase in robustness that occurs when the elastic mode is used with a nonlinear programming solver.

Acknowledgments. Thanks to Sven Leyffer for providing the source files to his AMPL examples and to Danny Ralph for providing very useful comments to an earlier version of this article. I am particularly grateful to Todd Munson for assistance in running the examples on the NEOS server [20] using the `kestrel` interface [7]. I am also grateful to the anonymous referees for the suggestions that have improved the quality of the paper.

REFERENCES

[1] M. Anitescu, *On using the elastic mode in nonlinear programming approaches to mathematical programs with complementarity constraints*, SIAM J. Optim., 15 (2005), pp. 1203–1236.

[2] H. Y. Benson, A. Sen, D. F. Shanno, and R. J. Vanderbei, *Interior-Point Algorithms, Penalty Methods, and Equilibrium Problems*, Preprint ORFE 03-02, Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ, 2003. Available online at http://www.optimization-online.org/DB_FILE/2003/10/744.pdf

[3] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995.

[4] F. Bonnans and A. Shapiro, *Perturbation Analysis of Optimization Problems*, Springer-Verlag, New York, 2000.

[5] J. R. Bunch and L. Kaufman, *Some stable methods for calculating inertia and solving symmetric linear systems*, Math. Comput., 31 (1977), pp. 163–179.

[6] R. W. Cottle, J.-S. Pan, and R. E. Stone, *The Linear Complementarity Problem*, Academic Press, Boston, 1992.

[7] E. D. Dolan, R. Fourer, J.-P. Goux, and T. S. Munson, *Kestrel: An Interface from Modeling Systems to the NEOS Server*, Preprint ANL/MCS-P986-0802, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 2002. Available online at http://www.optimization-online.org/DB_FILE/2003/10/546.pdf

[8] R. Fletcher, *Practical Methods of Optimization*, John Wiley and Sons, Chichester, 1987.

[9] R. Fletcher and S. Leyffer, *Solving mathematical programs with complementarity constraints as nonlinear programs*, Optim. Methods Software, 19 (2004), pp. 15–40.

[10] R. Fletcher, S. Leyffer, D. Ralph, and S. Sholtes, *Local Convergence of SQP Methods for Mathematical Programs with Equilibrium Constraints*, Tech. report NA/210, University of Dundee, Dundee, UK, 2002. Available online at http://www.optimization-online.org/DB_FILE/2002/05/476.pdf

[11] R. Fletcher, S. Leyffer, and P. L. Toint, *On the global convergence of a filter–SQP algorithm*, SIAM J. Optim., 13 (2002), pp. 44–59.

[12] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, Brooks/Cole Publishing Company, Pacific Grove, CA, 2002.

[13] M. Fukushima and J.-S. Pang, *Convergence of a smoothing continuation method for mathematical programs with complementarity constraints*, in Ill-Posed Variational Problems and

Regularization Techniques, Lecture Notes in Econom. Math. Systems, 477, M. Thera and R. Tichatschke, eds., Springer-Verlag, Berlin, 1999, pp. 99–110.

[14] M. Fukushima and P. Tseng, *An implementable active-set algorithm for computing a B-stationary point of a mathematical program with linear complementarity constraints*, SIAM J. Optim., 12 (2002), pp. 724–739.

[15] P. E. Gill, W. Murray, and M. A. Saunders, *User's Guide for SNOPT* 5.3: *A Fortran Package for Large-scale Nonlinear Programming*, Tech. report NA 97-5, Department of Mathematics, University of California at San Diego, San Diego, CA, 1997. Available online at www.cam.ucsd.edu/~peg/papers/sndoc5.pdf

[16] H. Jiang and D. Ralph, *Smooth SQP methods for mathematical programs with nonlinear complementarity constraints*, SIAM J. Optim., 10 (2000), pp. 779–808.

[17] S. Leyffer, *Ampl Collection of Mathematical Programs with Equilibrium Constraints*, http://www-unix.mcs.anl.gov/~leyffer/MacMPEC.

[18] Z.-Q. Luo, J.-S. Pang, and D. Ralph, *Mathematical Programs with Complementarity Constraints*, Cambridge University Press, Cambridge, UK, 1996.

[19] O. L. Mangasarian and S. Fromovitz, *The Fritz John necessary optimality conditions in the presence of equality constraints*, J. Math. Anal. Appl., 17 (1967), pp. 34–47.

[20] *The NEOS Guide*, http://www.mcs.anl.gov/otc/Guide.

[21] R. H. Byrd, M. E. Hribar, and J. Nocedal, *An interior point algorithm for large-scale nonlinear programming*, SIAM J. Optim., 9 (1999), pp. 877–900.

[22] J. Outrata, M. Kocvara, and J. Zowe, *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints: Theory, Applications and Numerical Results*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[23] H. Scheel and S. Scholtes, *Mathematical programs with equilibrium constraints: Stationarity, optimality and sensitivity*, Math. Oper. Res., 25 (2000), pp. 1–25.

[24] S. Scholtes, *Convergence properties of a regularization scheme for mathematical programs with complementarity constraints*, SIAM J. Optim., 11 (2001), pp. 918–936.

SIAM J. OPTIM.
Vol. 16, No. 1, pp. 146–169

AN INFEASIBLE BUNDLE METHOD FOR NONSMOOTH CONVEX CONSTRAINED OPTIMIZATION WITHOUT A PENALTY FUNCTION OR A FILTER*

CLAUDIA SAGASTIZÁBAL† AND MIKHAIL SOLODOV†

Abstract. Global convergence in constrained optimization algorithms has traditionally been enforced by the use of parametrized penalty functions. Recently, the filter strategy has been introduced as an alternative. At least part of the motivation for using filter methods consists of avoiding the need for estimating a suitable penalty parameter, which is often a delicate task. In this paper, we demonstrate that the use of a parametrized penalty function in nonsmooth convex optimization can be avoided without using the relatively complex filter methods. We propose an approach which appears to be more direct and easier to implement, in the sense that it is closer in spirit and structure to the well-developed unconstrained bundle methods. Preliminary computational results are also reported.

Key words. constrained optimization, nonsmooth convex optimization, bundle methods

AMS subject classifications. 90C30, 65K05, 49D27

DOI. 10.1137/040603875

1. Introduction and motivation.

We consider the problem

$$\begin{cases} \min\limits_{x\in\mathbb{R}^n} f(x), \\ c(x) \le 0, \end{cases} \tag{1.1}$$

where $f, c : \mathbb{R}^n \to \mathbb{R}$ are convex functions which are, in general, nondifferentiable. We note that there is no loss of generality in formulating (1.1) with only one constraint: if necessary, c can be defined as the pointwise maximum of finitely many convex functions, thus covering the case of multiple inequality constraints. In our development, we assume that the Slater constraint qualification [28] holds; i.e., there exists $x \in \mathbb{R}^n$ such that $c(x) < 0$. We also assume that an *oracle* is available, which for any given $x \in \mathbb{R}^n$ computes the values $f(x)$ and $c(x)$, and *one* subgradient for each of the functions, i.e., some $g_f \in \partial f(x)$ and some $g_c \in \partial c(x)$. We do not assume that there is any control over which particular subgradients are computed by the oracle (for example, for problems with more than one constraint, i.e., when c is defined by the maximum operation, we may have subgradient information about only one constraint among those with the largest value).

Nonsmooth optimization (NSO) problems are, in general, difficult to solve, even when they are unconstrained. Among algorithms for NSO, we mention the subgradient [37], cutting-planes [6, 16], analytic center cutting-planes (ACCP) [12], and bundle methods [14, 36]. Bundle and ACCP methods are stabilized versions of the cutting-planes method, and they are currently recognized as the most robust and reliable NSO

*Received by the editors February 5, 2004; accepted for publication (in revised form) December 6, 2004; published electronically September 8, 2005.

http://www.siam.org/journals/siopt/16-1/60387.html

†Instituto de Matemática Pura e Aplicada, Estrada Dona Castorina 110, Jardim Botânico, Rio de Janeiro, RJ 22460-320, Brazil (sagastiz@impa.br, solodov@impa.br). The first author is on leave from INRIA-Rocquencourt, Le Chesnay, France. The research of the first author is supported by CNPq grant 303540/03-6, by PRONEX-Optimization, and by FAPERJ. The research of the second author is supported in part by CNPq grants 300734/95-6(RN) and 471780/2003-0, by PRONEX-Optimization, and by FAPERJ.

algorithms. ACCP methods are based on information given by a certain separation procedure, which puts it somewhat outside of the oracle framework considered here. In this paper we focus on bundle methods, specifically on their proximal form.

For unconstrained problems, iterates of a proximal bundle algorithm are generated by solving a quadratic programming problem (QP). Each QP is defined by means of a cutting-planes model of the objective function, stabilized by a quadratic term centered at the best point obtained so far (which is referred to as the last *descent* or *serious step*). An important feature of bundle methods is that the size of each QP can be controlled via the so-called *aggregation* techniques; see, for instance, [3, Ch. 9] and also section 2 below. We emphasize that the latter is crucial for any practical implementation.

Constrained nonsmooth problems are more complex, and only a few practical methods can be found in the literature. Convex problems with "easy" constraints (such as bound or linear constraints) can be solved either by inserting the constraints directly into each QP or by projecting iterates onto the feasible set; see, for instance, [11] and [20, 21]. For general convex constrained problems, such as problem (1.1) considered here, one possibility is to solve an equivalent unconstrained problem with an exact penalty objective function; see [18, 23]. This approach, however, presents some drawbacks, which are typical whenever a penalty function is employed. Specifically, estimating a suitable value of the penalty parameter is sometimes a delicate task. Furthermore, if a large value of the parameter is required to guarantee the exactness of a given penalty function, then numerical difficulties arise.

More recently, Fletcher and Leyffer [8] proposed the filter strategy [9] as an alternative to the use of a penalty function in the framework of bundle methods for solving (1.1). However, the development of [8] is quite involved and, in particular, the resulting method appears considerably more complicated when compared, for example, to standard bundle methods for the unconstrained case. Furthermore, techniques for bundle compression and aggregation, although mentioned in [8], are not explicitly addressed. As stated, the method of [8] does not guarantee that the number of constraints in the subproblems can be kept smaller than a given desired bound, even if "inactive cuts" are removed from the bundle. Without this feature, a method cannot be guaranteed to be practical.

For other bundle-type methods for (1.1) that do not use penalization, see [29, 30] and [17, Ch. 5]. But it should be emphasized that in the cited methods all the serious iterates, including the starting point, are required to be feasible. Therefore, there are no concerns associated with the use of penalty functions and no need for alternative strategies, such as filter methods. It should be noted that feasible methods suffer from a serious drawback: computing a feasible point is required to start the algorithm. This "phase I" general (nonsmooth) convex feasibility problem may be as difficult to solve as (1.1) itself. As a result, the overall computational burden of solving the problem may increase considerably. On the other hand, feasible methods can be useful in applications in which problem function(s) may not be defined everywhere outside of the feasible region. We point out that our method, if started from a feasible point, stays feasible (see Proposition 4.1) and thus can operate "in feasible mode" if an appropriate starting point is provided.

Before proceeding with our discussion, we introduce the *improvement function* associated with problem (1.1). For a given $x \in \mathbb{R}^n$, let

$$h_x(y) := \max\{f(y) - f(x), c(y)\}, \quad y \in \mathbb{R}^n. \tag{1.2}$$

Among other things, it holds that $\bar{x}$ is a solution to (1.1) if and only if $\bar{x}$ solves the unconstrained problem of minimizing $h_{\bar{x}}$ (see Theorem 2.1 below). The use of $h_{\bar{x}}$ as a theoretical tool in the convergence analysis of bundle-type methods can be traced back to [29]; see also [17]. However, in none of these works is the improvement function used in the algorithms themselves. In addition, since in [29, 17] the infeasible iterates are automatically declared "null steps," the test to accept an iterate as the next serious step involves the objective function f only. Thus the resulting sequence of serious steps is both feasible and monotone in f. The piecewise linearization of the improvement function has also been used in the methods of feasible directions for solving *smooth* problems (see, e.g., [39, 33]). However, those methods are again feasible and the improvement function itself is not involved in the algorithms. To our knowledge, the only algorithms where the improvement function has been used directly are the (inexact) proximal point methods of [1]. In some sense, [1] can be considered as a predecessor of the present paper, but keeping in mind the well-known important differences between proximal point and bundle methods (conceptually solving proximal subproblems, even if approximately, to obtain a new iterate versus accepting a new iterate once a computationally realistic sufficient descent condition is satisfied).

Infeasible bundle methods are very rare. Prior to [8], we could find in the literature only the "phase I–phase II" modification of the feasible method in [17, Ch. 5.7] and the constrained level bundle methods of [25]. In [25], successive approximations of the exact improvement function $h_{\bar{x}}$ are used in the algorithm. Specifically, in the expression

$$h_{\bar{x}}(y) = \max\{f(y) - f(\bar{x}), c(y)\} = \lambda f(y) + (1-\lambda)c(y) - \lambda f(\bar{x}) \quad \text{for some } \lambda \in [0,1],$$

the values of λ and $f(\bar{x})$ are estimated at each iteration. Those estimates are used to define a certain gap function and an associated level parameter for the QP. It is well known that level methods are especially suitable for those problems in which the optimal value $f(\bar{x})$ is either known or easy to estimate. This certainly is not true in general. In fact, estimating the optimal value is a delicate issue, and inappropriately chosen values may lead to infeasible QPs.

In this paper, we propose an infeasible proximal bundle method for solving (1.1), which uses neither a penalty function nor a filter. With respect to [30, 17], the advantage is that it is not necessary to compute a feasible point to start the algorithm. Also, since serious steps can be infeasible, monotonicity in f is not enforced (outside of the feasible set). Rather, there is a balance between the search for feasibility and the reduction of the objective function. But this balance is followed in a manner different from the filter strategy. We also emphasize that, compared to [8], our method is much closer to the well-developed unconstrained bundle methods, and thus is easier to implement. For example, we can manage the size of QPs by a suitable modification of the standard aggregation techniques. Finally, compared to [25], QPs in our method are always feasible independently of the choice of parameters.

Our approach can be viewed as an unconstrained proximal bundle method [14, 17, 3] applied to the function $h_x(\cdot)$ directly, with the important distinction that x is the last serious step, and thus, the function being minimized varies along the iterations; see section 3 for details. We emphasize that serious steps need not be monotone in f or feasible. Of course, the fact that the improvement function changes along the iterations makes standard convergence results not applicable directly, and specific analysis is needed. Actually, some subtle modifications are needed also in the bundle method itself. Nevertheless, our approach is quite close to standard unconstrained bundle methods. Apart from leading to relative ease in the computer implementation, this also opens the potential for extending various results obtained for the unconstrained bundle methods to the constrained case, e.g., the variable metric [2, 26, 27] and quasi-Newton [5, 32] techniques, methods with inexact data [13, 38], etc.

This paper is organized as follows. In section 2, we state some basic properties of the improvement function and also give an overview of proximal bundle methods for the unconstrained case, including the aggregation and compression techniques. This is done in order to set the notation for the algorithm, and also to build a link from the well-known unconstrained method to the constrained one. The algorithm itself is stated in section 3, where some preliminary properties also are established. Convergence analysis is provided in section 4, and computational experience is reported in section 5.

Our notation is fairly standard. The Euclidean inner product in $\mathbb{R}^n$ is denoted by $\langle x, y\rangle = \sum_{j=1}^n x_j y_j$, and the associated norm by $\|\cdot\|$. The positive-part function is denoted by $x^+ := \max\{x, 0\}$. For a set X in $\mathbb{R}^n$, $\operatorname{conv} X$ denotes its convex hull. By $\partial_\varepsilon h(x)$ we denote the ε-subdifferential of a convex function h at the point $x \in \mathbb{R}^n$, with $\partial_0 h(x) = \partial h(x)$ being the usual subdifferential.

2. Preliminaries. We start with the properties of the improvement function to be used in what follows. Next, we discuss some basics of the standard bundle methods, mainly to fix notation and remind the reader of the principal relations. Also, we use this discussion to point out where appropriate modifications would be needed when passing from the unconstrained to the constrained case. No proofs are given in this section. Proofs and calculations are worked out in detail for the constrained algorithm in section 4.

2.1. The improvement function. Directly by the definition (1.2), the subdifferential of the improvement function is given by

$$\partial h_x(y) = \begin{cases} \partial f(y) & \text{if } f(y) - f(x) > c(y), \\ \operatorname{conv}\{\partial f(y) \bigcup \partial c(y)\} & \text{if } f(y) - f(x) = c(y), \\ \partial c(y) & \text{if } f(y) - f(x) < c(y). \end{cases} \tag{2.1}$$

In addition, we have that

$$h_x(x) = c^+(x) = \max\{c(x), 0\} \quad \text{for all } x \in \mathbb{R}^n.$$

Finally (see, e.g., [17, Lem. 2.16, p. 17]), the following holds.

THEOREM 2.1. *Suppose that the Slater constraint qualification is satisfied for* (1.1)*. Then the following statements are equivalent:*

(i) $\bar{x}$ *is a solution to* (1.1);

(ii) $\min\{h_{\bar{x}}(y) \mid y \in \mathbb{R}^n\} = h_{\bar{x}}(\bar{x}) = 0$;

(iii) $0 \in \partial h_{\bar{x}}(\bar{x})$*, i.e.,* $0 \in \partial\varphi(\bar{x})$*, where* $\varphi(\cdot) := h_{\bar{x}}(\cdot)$.

2.2. An overview of unconstrained bundle methods. Consider, for the moment, the unconstrained problem

$$\min_{x \in \mathbb{R}^n} h(x),$$

where $h(\cdot)$ is some fixed convex function. For the sake of simplicity, we also suppose for now that there is no bundle compression/aggregation in the algorithm. We refer the reader to [3, Ch. 9.3] for proofs of the relations stated in this subsection.

Let ℓ be the current iteration index. Bundle methods keep memory of the past in a *bundle* of information

$$\mathcal{B}_\ell := \bigcup_{i<\ell}\{(y^i, h_i = h(y^i), g_h^i \in \partial h(y^i))\} \quad \text{and} \quad (x^k, h(x^k)),\ \ k = k(\ell),$$

where $k(\ell)$ denotes the index of the last serious step preceding the iteration ℓ. Serious iterates, also called stability centers, form a subsequence $\{x^k\} \subset \{y^i\}$ such that $\{h(x^k)\}$ is strictly decreasing. This will be made more precise later.

We mention two peculiarities of our notation. When it is clear from the context, we shall not explicitly specify the dependence of k on the current iteration index ℓ. Also, in text that follows we shall write $i \in \mathcal{B}_\ell$ to mean that there exists an element in the set $\mathcal{B}_\ell$ indexed by i. Although this notation is formally improper, it does not lead to any confusion, while simplifying some relations below.

The bundle of past information is used to define at each iteration a cutting-planes model of the objective function,

$$\psi_\ell(y) := \max_{i\in\mathcal{B}_\ell}\{h_i + \langle g_h^i, y - y^i\rangle\}.$$

An equivalent expression, better suited for implementations, centers the cutting-planes model at the stability center x^k:

$$\psi_\ell(y) = h(x^k) + \max_{i\in\mathcal{B}_\ell}\{-e_i^k + \langle g_h^i, y - x^k\rangle\}, \tag{2.2}$$

where the terms e_i^k are the (nonnegative) linearization errors

$$e_i^k := h(x^k) - h_i - \langle g_h^i, x^k - y^i\rangle.$$

In particular,

$$g_h^i \in \partial_{e_i^k} h(x^k),$$

i.e., $h(y) \geq h(x^k) + \langle g_h^i, y - x^k\rangle - e_i^k$ for all $y \in \mathbb{R}^n$.

Since the linearization errors depend on x^k, they need to be updated every time x^k changes (for this reason, they are indexed by both k and i). For further reference, note that the linearization errors obviously depend also on h (h is fixed in this section, but not in the rest of the paper). Thus in the update of the linearization errors in our algorithm, we shall also have to account for an eventual change in h.

The advantage of expressing the model in the form of (2.2) is that it requires less memory for storing the relevant information: the bundle becomes

$$\mathcal{B}_\ell = \bigcup_{i<\ell}\{(e_i^k \in \mathbb{R}_+, g_h^i \in \partial_{e_i^k} h(x^k))\} \quad \text{and} \quad (x^k, h(x^k)).$$

Given μ_ℓ, a positive proximal parameter, the next iterate y^ℓ is generated by solving a QP reformulation of the problem

$$\min_{y\in\mathbb{R}^n} \psi_\ell(y) + \frac{1}{2}\mu_\ell\|y - x^k\|^2.$$

Clearly, y^ℓ is unique. Furthermore, it is characterized by the following conditions (see [3, Lem. 9.8]):

$$y^\ell = x^k - \frac{1}{\mu_\ell}\hat{g}^\ell, \quad \text{where } \hat{g}^\ell \in \partial\psi_\ell(y^\ell),$$

$$\hat{g}^\ell \in \partial_{\hat{\varepsilon}_\ell^k} h(x^k), \quad \text{where } \hat{\varepsilon}_\ell^k = h(x^k) - \psi_\ell(y^\ell) - \frac{1}{\mu_\ell}\|\hat{g}^\ell\|^2 \geq 0.$$

An iterate y^ℓ is considered good enough to become the next serious step when $h(y^\ell)$ provides a significant decrease (with respect to $h(x^k)$), measured in terms of a nominal decrease. Specifically, let $m \in (0,1)$ be a given parameter. The nominal decrease is defined by

$$\delta_\ell := h(x^k) - \psi_\ell(y^\ell) - \frac{1}{2}\mu_\ell \|y^\ell - x^k\|^2 = \hat{\varepsilon}_\ell^k + \frac{1}{2\mu_\ell}\|\hat{g}^\ell\|^2 \geq 0.$$

When y^ℓ satisfies the descent test

$$h(y^\ell) \leq h(x^k) - m\delta_\ell, \tag{2.3}$$

a serious step is declared: $x^{k+1} = y^\ell$. Otherwise, y^ℓ is declared a null step and x^k remains unchanged.

The algorithm stops when δ_ℓ is small enough (when compared to a given tolerance). In this case, both $\hat{\varepsilon}_\ell^k$ and $\|\hat{g}^\ell\|$ are small and, since $\hat{g}^\ell \in \partial_{\hat{\varepsilon}_\ell^k} h(x^k)$, for any $M > 0$ and all $y \in \mathbb{R}^n$ such that $\|y - x^k\| \leq M$, the approximate optimality condition $h(y) \geq h(x^k) - \hat{\varepsilon}_\ell^k - M\|\hat{g}^\ell\|$ holds.

We next consider the effect of compressing the bundle.

2.3. Aggregation technique. The number of constraints in the QP used to generate y^ℓ is precisely the number of elements in the bundle $\mathcal{B}_\ell$. Obviously, one has to keep this number computationally manageable. Thus, the bundle has to be *compressed* when the number of elements reaches some chosen bound. This has to be done without impairing convergence of the algorithm. For this purpose, the so-called *aggregate* function is fundamental:

$$l_{k,\ell}(y) := h(x^k) - \hat{\varepsilon}_\ell^k + \langle \hat{g}^\ell, y - x^k \rangle, \quad k = k(\ell).$$

Note that this function can be defined directly from the aggregate couple $(\hat{\varepsilon}_\ell^k, \hat{g}^\ell \in \partial_{\hat{\varepsilon}_\ell^k} h(x^k))$. Alternatively, the same information can be retrieved from all the "active" bundle elements, i.e., those defining $\psi_\ell(y^\ell)$.

Before looping from ℓ to $\ell+1$, the next bundle $\mathcal{B}_{\ell+1}$ is defined. If the bundle has reached its maximum allowed size, it must be compressed. Reducing the bundle amounts to replacing (at iteration $\ell+1$) the cutting-planes model (2.2) with another function, defined by a smaller number of cutting-planes, which we shall still denote by $\psi_{\ell+1}$. As pointed out in [7, sect. 4, eqs. (4.7)–(4.9)], one can use any collection of functions satisfying (for all $y \in \mathbb{R}^n$) the following three conditions:

$$\psi_\ell(y) \leq h(y) \qquad \text{for all } \ell \geq 1, \tag{2.4a}$$

$$l_{k(\ell),\ell}(y) \leq \psi_{\ell+1}(y) \quad \text{for those } \ell \text{ for which } y^\ell \text{ is a null step,} \tag{2.4b}$$

$$h_\ell + \langle g_h^\ell, y - y^\ell \rangle \leq \psi_{\ell+1}(y) \quad \text{for those } \ell \text{ for which } y^\ell \text{ is a null step.} \tag{2.4c}$$

We note that (2.4a) will not be automatic in our setting. Indeed, as already mentioned, the function h will change after every serious step. As a consequence, (2.4a) can be violated unless appropriate care is taken.

Suppose, however, that (2.4a) holds. In terms of bundle information, the remaining conditions mean that it is enough for the new bundle to contain both the aggregate information (to ensure (2.4b)) and the last generated information (to ensure (2.4c)). These values are, respectively, $(\hat{\varepsilon}_\ell^k, \hat{g}^\ell)$ and $(y^\ell, h_\ell, g_h^\ell \in \partial h(y^\ell))$. In particular, at any iteration, the bundle can contain as few elements as we wish (as long as the two specified above are included). Note also that if the bundle is not compressed at the current iteration, then the aggregate information is redundant (because it is already contained in the bundle elements, which are active in the QP subproblem).

Accordingly, we shall write the next bundle in the form

$$\mathcal{B}_{\ell+1} := \mathcal{B}^{\rm oracle}_{\ell+1} \bigcup \mathcal{B}^{\rm agg}_{\ell+1} \quad \text{and} \quad (x^k, h(x^k)),\ k = k(\ell+1),\ \text{the last serious iterate,}$$

where the oracle bundle is any set such that

$$\{(e^k_\ell, g^\ell_h)\} \subseteq \mathcal{B}^{\rm oracle}_{\ell+1} \subseteq \bigcup_{i\le\ell} \{(e^k_i \in \mathbb{R}_+, g^i_h \in \partial_{e^k_i} h(x^k))\},$$

while the aggregate bundle satisfies

$$\{(\hat\varepsilon^k_\ell, \hat g^\ell)\} \subseteq \mathcal{B}^{\rm agg}_{\ell+1} \subseteq \bigcup_{i\le\ell} \{(\hat\varepsilon^k_i \in \mathbb{R}_+, \hat g^i \in \partial_{\hat\varepsilon^k_i} h(x^k))\}.$$

The leftmost inclusions in the last two relations above need to be specified explicitly only when there is bundle compression at the ℓth iteration (if there is no compression, they hold automatically because of the rightmost inclusions). We note that, similarly to updating the linearization errors e^k_i, the quantities $\hat\varepsilon^k_i$ also need to be updated every time k changes; see (2.5) and (3.6) below.

The next cutting-planes model is then defined by

$$\begin{aligned}\psi_{\ell+1}(y) = h(x^k) + \max\Bigg\{ &\max_{i\in\mathcal{B}^{\rm oracle}_{\ell+1}} \{-e^k_i + \langle g^i_h, y - x^k\rangle\},\\ &\max_{i\in\mathcal{B}^{\rm agg}_{\ell+1}} \{-\hat\varepsilon^k_i + \langle \hat g^i, y - x^k\rangle\}\Bigg\}, \quad k = k(\ell+1).\end{aligned}$$

As already mentioned, every time a new serious step has been declared, both linearization and aggregate errors need to be modified. The update aims at satisfying the key relations

$$g^i_h \in \partial_{e^k_i} h(x^{k+1}) \quad \text{and} \quad \hat g^i \in \partial_{\hat\varepsilon^k_i} h(x^{k+1}),$$

which should hold for all elements in the new bundle. The following simple updating formulas guarantee the required properties (when h is fixed):

$$\text{(2.5)} \qquad \begin{cases} e^{k+1}_i := e^k_i + h(x^{k+1}) - h(x^k) + \langle g^i_h, x^k - x^{k+1}\rangle & \text{if } i \in \mathcal{B}^{\rm oracle}_{\ell+1},\\ \hat\varepsilon^{k+1}_i := \hat\varepsilon^k_i + h(x^{k+1}) - h(x^k) + \langle \hat g^i, x^k - x^{k+1}\rangle & \text{if } i \in \mathcal{B}^{\rm agg}_{\ell+1}.\end{cases}$$

We next show how to adapt the unconstrained bundle methodology described above to solving the constrained problem (1.1).

3. Defining the constrained algorithm. Given the last serious iterate x^k (we note that the starting point x^0 is considered a serious iterate), we apply an unconstrained proximal bundle method to the function $h(\cdot) := h_k(\cdot) = h_{x^k}(\cdot)$ until the next serious iterate x^{k+1} is generated. At this time, we change $h(\cdot)$ to $h_{k+1}(\cdot) = h_{x^{k+1}}(\cdot)$, make the necessary changes to the bundle, and repeat the process. We point out that the development is not straightforward. For one thing, it is possible that $f(x^{k+1}) > f(x^k)$. As is easy to observe, in that case we have $h_{k+1}(\cdot) \le h_k(\cdot)$. As a consequence, the accumulated cutting-planes model for $h_k(\cdot)$ may not be a valid (lower) approximation for $h_{k+1}(\cdot)$. Thus, the model has to be revised and adjusted to ensure that conditions (2.4a)–(2.4c) (in particular, (2.4a)) are satisfied for the new $h(\cdot) := h_{k+1}(\cdot)$. Note that this adjustment is independent of compressing the bundle, which will require additional care.

In the following, we explain how to build the model ψ_ℓ satisfying (2.4a)–(2.4c), even when $h(\cdot)$ changes at a serious step.

3.1. Bundle information. Since $h(\cdot)$ varies with k, past information relevant for constructing the model is no longer just (e_i, g_h^i). In particular, separate information about the objective and constraint functions needs to be kept. This information is $(f_i = f(y^i),\ c_i = c(y^i))$ and $(g_f^i \in \partial f(y^i),\ g_c^i \in \partial c(y^i))$, or, equivalently, $(e_{f_i}^k, e_{c_i}^k, g_f^i \in \partial_{e_{f_i}^k} f(x^k),\ g_c^i \in \partial_{e_{c_i}^k} c(x^k))$, where the linearization errors for f and c, respectively, are

$$\begin{aligned} e_{f_i}^k &:= f(x^k) - f_i - \langle g_f^i, x^k - y^i\rangle, \\ e_{c_i}^k &:= c(x^k) - c_i - \langle g_c^i, x^k - y^i\rangle. \end{aligned} \tag{3.1}$$

The purpose of keeping the bundle information separated is twofold:

- First, knowing (f_i, c_i) makes it possible to compute the function and subgradient values for different functions h; see Lemma 3.1 below.
- Second, as shown in Lemma 3.2 below, separate linearization errors can be updated by a simple formula, even when h changes.

Therefore, we define

$$\begin{aligned} &\mathcal{B}_\ell := \mathcal{B}_\ell^{\text{oracle}} \bigcup \mathcal{B}_\ell^{\text{agg}} \quad \text{and} \quad (x^k, f(x^k), c(x^k)),\ k = k(\ell), \text{ the last serious iterate,} \\ &\text{with } \mathcal{B}_\ell^{\text{oracle}} \subseteq \bigcup_{i<\ell} \left\{\left(f_i, c_i, e_{f_i}^k, e_{c_i}^k, g_f^i \in \partial_{e_{f_i}^k} f(x^k), g_c^i \in \partial_{e_{c_i}^k} c(x^k)\right)\right\} \\ &\text{and } \mathcal{B}_\ell^{\text{agg}} \subseteq \bigcup_{i<\ell} \{(\hat\varepsilon_i^k, \hat g^i \in \partial_{\hat\varepsilon_i^k} h_k(x^k))\}. \end{aligned} \tag{3.2}$$

LEMMA 3.1. *In the notation of* (3.1) *and* (3.2), *for each* $i \in \mathcal{B}_\ell^{\text{oracle}}$, *define*

$$\begin{cases} e_i^k := e_{f_i}^k + c^+(x^k) \quad \text{and} \quad g_{h_k}^i := g_f^i & \text{if } f_i - f(x^k) \ge c_i, \\ e_i^k := e_{c_i}^k + c^+(x^k) - c(x^k) \quad \text{and} \quad g_{h_k}^i := g_c^i & \text{if } f_i - f(x^k) < c_i. \end{cases} \tag{3.3}$$

Then $e_i^k \ge 0$ *and* $g_{h_k}^i \in \partial_{e_i^k} h_k(x^k)$.

Proof. By (3.1) and the convexity of f and $c, e_{f_i}^k \ge 0$ and $e_{c_i}^k \ge 0$. Since also $c^+(x^k) \ge 0$ and $c^+(x^k) - c(x^k) \ge 0$, (3.3) implies that $e_i^k \ge 0$.

Recalling that $h_k(x^k) = c^+(x^k)$, we have to show that for all $y \in \mathbb{R}^n$, it holds that $h_k(y) \ge c^+(x^k) + \langle g_{h_k}^i, y - x^k\rangle - e_i^k$. By using the definitions of h_k, of the subdifferential, and of the errors $e_{f_i}^k$, $e_{c_i}^k$, we obtain that

$$\begin{aligned} h_k(y) &= \max \begin{cases} f(y) - f(x^k) \\ c(y) \end{cases} \\ &\ge \max \begin{cases} f_i - f(x^k) + \langle g_f^i, y - y^i\rangle \\ c_i + \langle g_c^i, y - y^i\rangle \end{cases} \\ &= \max \begin{cases} \langle g_f^i, y - x^k\rangle - e_{f_i}^k \\ c(x^k) + \langle g_c^i, y - x^k\rangle - e_{c_i}^k. \end{cases} \end{aligned}$$

By adding and subtracting $c^+(x^k)$ in the right-hand side of the relation above, and using the definition of $g^i_{h_k}$, we obtain that

$$h_k(y) \geq c^+(x^k) + \langle g^i_{h_k}, y - x^k \rangle - \begin{cases} (e^k_{f_i} + c^+(x^k)) & \text{if } f_i - f(x^k) \geq c_i, \\ (e^k_{c_i} + c^+(x^k) - c(x^k)) & \text{if } f_i - f(x^k) < c_i. \end{cases}$$

The result now follows from the definition of e^k_i in (3.3). □

The cutting-planes model associated with (3.2), (3.3) is given by

$$\begin{aligned} \psi_\ell(y) = c^+(x^k) + \max \Bigg\{ & \max_{i \in \mathcal{B}^{\text{oracle}}_\ell} \{-e^k_i + \langle g^i_{h_k}, y - x^k \rangle\}, \\ & \max_{i \in \mathcal{B}^{\text{agg}}_\ell} \{-\hat{\varepsilon}^k_i + \langle \hat{g}^i, y - x^k \rangle\} \Bigg\}, \quad k = k(\ell). \end{aligned} \tag{3.4}$$

For this model to satisfy (2.4a)–(2.4c) when passing to the iteration $\ell + 1$, we consider separately the two cases of the ℓth iteration being a null step and the ℓth iteration being a serious step.

Suppose first that the QP subproblem defined with ψ_ℓ given by (3.4) generates y^ℓ as a null step. By construction, the new bundle satisfies (3.2) and (3.3) with ℓ replaced by $\ell + 1$ (k remains the same). Thus, Lemma 3.1 holds, and $g^i_{h_k} \in \partial_{e^k_i} h_k(x^k)$ for all $i \in \mathcal{B}^{\text{oracle}}_{\ell+1}$. Likewise, aggregate subgradients satisfy the inclusion $\hat{g}^i \in \partial_{\hat{\varepsilon}^k_i} h_k(x^k)$ for all $i \in \mathcal{B}^{\text{agg}}_{\ell+1}$. Therefore, (2.4a) (with ℓ replaced by $\ell + 1$) is automatically satisfied. Finally, for conditions (2.4b) and (2.4c) to hold, it is enough to make sure that

$$\begin{aligned} \{(e^k_\ell, g^\ell_{h_k} \in \partial_{e^k_\ell} h_k(x^k))\} &\subseteq \mathcal{B}^{\text{oracle}}_{\ell+1} \quad \text{and} \\ \{(\hat{\varepsilon}^k_\ell, \hat{g}^\ell \in \partial_{\hat{\varepsilon}^k_\ell} h_k(x^k))\} &\subseteq \mathcal{B}^{\text{agg}}_{\ell+1} \quad \text{if there is compression.} \end{aligned}$$

Those inclusions are also automatically satisfied if the bundle is managed as in any standard method; see Step 4 in Algorithm 3.1 below.

Therefore, when there is a null step, the update of the bundle (and of the model) does not present any problem. This is as expected, since the function $h(\cdot) = h_k(\cdot)$ is fixed between consecutive serious steps. The situation changes when y^ℓ is declared a serious step. Specifically, the aggregate bundle elements need a special update. We discuss this case next.

3.2. Adjusting the model after a serious step.

Suppose that for some iteration ℓ the descent test is satisfied (i.e., condition (2.3) with h replaced by h_k) so that a new stability center $x^{k+1} = y^\ell$ is generated. This means, in particular, that at the next iterate we start working with the new function $h_{k+1}(\cdot) = h_{x^{k+1}}(\cdot)$.

As mentioned in [7], conditions (2.4a)–(2.4c) guarantee that the bundle technique applied to the new function $h(\cdot) = h_{k+1}(\cdot)$ produces a descent step after a finite number of null steps, or else the point x^{k+1} is a minimum of $h_{k+1}(\cdot)$. However, condition (2.4a) (with $\ell = \ell + 1$) is not automatic in our setting, and the model may need to be properly adjusted. Indeed, even though

$$\begin{gathered} \psi_\ell(y) \leq h_k(y), \\ \text{and } c^+(x^k) + \langle \hat{g}^i, y - x^k \rangle - \hat{\varepsilon}^k_i \leq h_k(y), \quad i \in \mathcal{B}^{\text{agg}}_\ell, \end{gathered}$$

the same inequalities may not hold with h_k replaced by h_{k+1}. Specifically, if $f(x^k) < f(x^{k+1})$, which is possible, then we have that $h_k(y) \geq h_{k+1}(y)$. Thus, the key relations (2.4a)–(2.4c) are not guaranteed and, in general, do not hold.

There are various ways to ensure (2.4a)–(2.4c) after a serious step is taken. In fact, as discussed in [7], any approximation satisfying (2.4a) is acceptable—even a bad one—because the future null steps satisfying (2.4b) and (2.4c) would eventually build up a good approximation (of course, starting with a bad approximation is computationally inefficient). We next present one approach to ensure that all bundle elements correspond to appropriate approximate subgradients of the new function h_{k+1} at x^{k+1} so that both convergence and computational efficiency are guaranteed. For oracle bundle elements, we need only center (separate) linearization errors of f and c at the new point x^{k+1}. For the aggregate bundle elements, some special care is needed.

LEMMA 3.2. *Let ψ_ℓ be defined by* (3.4)*, using* (3.2) *and* (3.3)*. Suppose that the associated y^ℓ is declared a serious step, i.e., $x^{k+1} = y^\ell$. Then the following holds:*

(i) *For each $i \in \mathcal{B}_\ell^{\text{oracle}}$, the linearization errors*

$$\begin{aligned} {e_f}_i^{k+1} &= {e_f}_i^k + f(x^{k+1}) - f(x^k) + \langle g_f^i, x^k - x^{k+1}\rangle, \\ {e_c}_i^{k+1} &= {e_c}_i^k + c(x^{k+1}) - c(x^k) + \langle g_c^i, x^k - x^{k+1}\rangle \end{aligned} \tag{3.5}$$

satisfy (3.1) *with $k = k+1$. As a result, $g^i_{h_{k+1}} \in \partial_{e_i^{k+1}} h_{k+1}(x^{k+1})$, where $e_i^{k+1} \geq 0$ and $g^i_{h_{k+1}}$ are defined in* (3.3) *with k replaced by $k+1$.*

(ii) *For each $i \in \mathcal{B}_\ell^{\text{agg}}$, define*

$$\hat{\varepsilon}_i^{k+1} := \hat{\varepsilon}_i^k + c^+(x^{k+1}) - c^+(x^k) + (f(x^{k+1}) - f(x^k))^+ + \langle \hat{g}^i, x^k - x^{k+1}\rangle. \tag{3.6}$$

Then $\hat{\varepsilon}_i^{k+1} \geq 0$ and $\hat{g}^i \in \partial_{\hat{\varepsilon}_i^{k+1}} h_{k+1}(x^{k+1})$.

Proof. Let $i \in \mathcal{B}_\ell^{\text{oracle}}$. Because $g_f^i \in \partial_{{e_f}_i^k} f(x^k)$, for all $y \in \mathbb{R}^n$ we have that

$$\begin{aligned} f(y) &\geq f(x^k) + \langle g_f^i, y - x^k\rangle - {e_f}_i^k \\ &= f(x^{k+1}) + \langle g_f^i, y - x^{k+1}\rangle \\ &\quad - ({e_f}_i^k + f(x^{k+1}) - f(x^k) + \langle g_f^i, x^k - x^{k+1}\rangle). \end{aligned}$$

Hence, $g_f^i \in \partial_{{e_f}_i^{k+1}} f(x^{k+1})$. By the same argument, $g_c^i \in \partial_{{e_c}_i^{k+1}} c(x^{k+1})$. Since f and c are convex, and ${e_f}_i^k$ and ${e_c}_i^k$ are nonnegative, (3.5) implies that ${e_f}_i^{k+1}, {e_c}_i^{k+1} \geq 0$. The remaining assertion of item (i) then follows by applying Lemma 3.1, where the quantities $(\ell, k, {e_f}_i^k, {e_c}_i^k)$ are replaced by $(\ell+1, k+1, {e_f}_i^{k+1}, {e_c}_i^{k+1})$, respectively.

Now, let $i \in \mathcal{B}_\ell^{\text{agg}}$. By (3.2), $\hat{g}^i \in \partial_{\hat{\varepsilon}_i^k} h_k(x^k)$. Hence, for all $y \in \mathbb{R}^n$, it holds that

$$h_k(y) \geq h_k(x^k) + \langle \hat{g}^i, y - x^k\rangle - \hat{\varepsilon}_i^k = c^+(x^k) + \langle \hat{g}^i, y - x^k\rangle - \hat{\varepsilon}_i^k. \tag{3.7}$$

In particular, for $y = x^{k+1}$, using the definitions of h_k and $\hat{\varepsilon}_i^{k+1}$, (3.7) yields

$$\begin{aligned} \max\{f(x^{k+1}) - f(x^k), c(x^{k+1})\} &\geq c^+(x^k) + \langle \hat{g}^i, x^{k+1} - x^k\rangle - \hat{\varepsilon}_i^k \\ &= -\hat{\varepsilon}_i^{k+1} + c^+(x^{k+1}) + (f(x^{k+1}) - f(x^k))^+. \end{aligned}$$

Thus, $\hat{\varepsilon}_i^{k+1} \geq c^+(x^{k+1}) + (f(x^{k+1}) - f(x^k))^+ - \max\{f(x^{k+1}) - f(x^k), c(x^{k+1})\} \geq 0$. Now rewrite (3.7) as follows:

$$h_k(y) \geq c^+(x^{k+1}) + \langle \hat{g}^i, y - x^{k+1}\rangle - \left(\hat{\varepsilon}_i^k + c^+(x^{k+1}) - c^+(x^k) + \langle \hat{g}^i, x^k - x^{k+1}\rangle\right).$$

Using (3.6), we obtain that

$$h_k(y) \geq h_{k+1}(x^{k+1}) + \langle \hat{g}^i, y - x^{k+1}\rangle - (\hat{\varepsilon}_i^{k+1} - (f(x^{k+1}) - f(x^k))^+). \tag{3.8}$$

The assertion of item (ii) follows from (3.8) if we establish that

$$h_{k+1}(y) \geq h_k(y) - (f(x^{k+1}) - f(x^k))^+ \quad \text{for all } y \in \mathbb{R}^n. \tag{3.9}$$

We now prove (3.9).

If $f(x^{k+1}) \leq f(x^k)$, it easily follows that $h_{k+1}(y) \geq h_k(y)$ for all $y \in \mathbb{R}^n$. This obviously implies (3.9). Suppose now that $f(x^{k+1}) > f(x^k)$. For $y \in \mathbb{R}^n$ such that $f(y) - f(x^{k+1}) \geq c(y)$, we have that $h_{k+1}(y) - h_k(y) = -f(x^{k+1}) + f(x^k) = -(f(x^{k+1}) - f(x^k))^+$, and so (3.9) holds. If $f(y) - f(x^{k+1}) < c(y)$ and $f(y) - f(x^k) \leq c(y)$, then $h_{k+1}(y) = h_k(y) = c(y)$, implying (3.9). Finally, if $f(y) - f(x^{k+1}) < c(y)$ and $f(y) - f(x^k) > c(y)$, we have that $h_{k+1}(y) > f(y) - f(x^{k+1}) = h_k(y) + f(x^k) - f(x^{k+1})$, which again gives (3.9). The proof is complete. □

As a consequence of Lemma 3.2, regardless of whether the ℓth iteration produced a null step or a serious step, if

$$\mathcal{B}^{\text{oracle}}_{\ell+1} \subseteq \bigcup_{i \leq \ell} \left\{ \left(f_i, c_i, e_{f_i}^{k+1}, e_{c_i}^{k+1}, g_f^i, g_c^i\right) \right\} \quad \text{and} \quad \mathcal{B}^{\text{agg}}_{\ell+1} \subseteq \bigcup_{i \leq \ell} \{(\hat{\varepsilon}_i^{k+1}, \hat{g}^i)\},$$

then the model

$$\psi_{\ell+1}(y) = c^+(x^k) + \max \left\{ \begin{array}{l} \max\limits_{i \in \mathcal{B}^{\text{oracle}}_{\ell+1}} \{-e_i^k + \langle g^i_{h_k}, y - x^k\rangle\}, \\ \max\limits_{i \in \mathcal{B}^{\text{agg}}_{\ell+1}} \{-\hat{\varepsilon}_i^k + \langle \hat{g}^i, y - x^k\rangle\} \end{array} \right\}, \quad k = k(\ell+1),$$

satisfies (2.4a) with ℓ replaced by $\ell+1$, with $h(\cdot) = h_k(\cdot)$, and with $k = k(\ell+1)$. Furthermore, if $(\hat{\varepsilon}_\ell^k, \hat{g}^\ell) \subseteq \mathcal{B}^{\text{agg}}_{\ell+1}$, then $\psi_{\ell+1}$ satisfies (2.4b), and if $(f_\ell, c_\ell, e_{f_\ell}, e_{c\ell}, g_f^\ell, g_c^\ell) \subseteq \mathcal{B}^{\text{oracle}}_{\ell+1}$, then $\psi_{\ell+1}$ satisfies (2.4c).

3.3. An infeasible constrained proximal bundle method. We are now in a position to give the algorithm in full detail.

ALGORITHM 3.1 (infeasible constrained proximal bundle method (ICPBM)).

Step 0. `Initialization.`

Choose parameters $m \in (0,1)$, *tol* ≥ 0, *and an integer* $|\mathcal{B}|_{\max} \geq 2$.

Choose $x^0 \in \mathbb{R}^n$. *Set* $y^0 := x^0$, *and compute* (f_0, c_0, g_f^0, g_c^0). *Set* $k = 0$, $\ell = 1$, $e_{f_0} := 0$, $e_{c0} := 0$ *and define the starting bundles* $\mathcal{B}_1^{\text{oracle}} := \{(e_{f_0}^0, e_{c0}^0, f_0, c_0, g_f^0, g_c^0)\}$ *and* $\mathcal{B}_1^{\text{agg}} := \emptyset$.

Step 1. `Quadratic programming subproblem.`

Choose $\mu_\ell > 0$ *and compute* y^ℓ *as the solution to*

$$\min_{y \in \mathbb{R}^n} \psi_\ell(y) + \frac{1}{2}\mu_\ell \|y - x^k\|^2, \tag{3.10}$$

where ψ_ℓ *is defined by* (3.4) *and* (3.3). *Compute*

$$\hat{g}^\ell = \mu_\ell(x^k - y^\ell), \quad \hat{\varepsilon}_\ell^k = c^+(x^k) - \psi_\ell(y^\ell) - \frac{1}{\mu_\ell}\|\hat{g}^\ell\|^2, \quad \delta_\ell = \hat{\varepsilon}_\ell^k + \frac{1}{2\mu_\ell}\|\hat{g}^\ell\|^2.$$

Compute $(f_\ell, c_\ell, g_f^\ell, g_c^\ell)$ *and* $(e_{f_\ell}^k, e_{c\ell}^k)$ *using* (3.1) *written with* $i = \ell$.

```
Step 2. Stopping test.
        If $\delta_\ell \leq tol$, stop.
Step 3. Descent test.
        Compute $h_\ell := h_k(y^\ell) = \max\{f_\ell - f(x^k), c_\ell\}$.
        If $h_\ell \leq c^+(x^k) - m\delta_\ell$, then declare a serious step. Otherwise, declare a null step.
Step 4. Bundle management.
        Set $\mathcal{B}^{\rm oracle}_{\ell+1} := \mathcal{B}^{\rm oracle}_{\ell}$ and $\mathcal{B}^{\rm agg}_{\ell+1} := \mathcal{B}^{\rm agg}_{\ell}$.
        If the bundle has reached the maximum bundle size, i.e.,
        if $|\mathcal{B}^{\rm oracle}_{\ell+1} \cup \mathcal{B}^{\rm agg}_{\ell+1}| = |\mathcal{B}|_{\max}$, then:
        Delete at least two elements from $\mathcal{B}^{\rm oracle}_{\ell+1} \cup \mathcal{B}^{\rm agg}_{\ell+1}$.
        Insert the aggregate couple $(\hat{\varepsilon}^k_\ell, \hat{g}^\ell)$ in $\mathcal{B}^{\rm agg}_{\ell+1}$.
        Append $(e^k_{f\ell}, e^k_{c\ell}, f_\ell, c_\ell, g^\ell_f, g^\ell_c)$ to $\mathcal{B}^{\rm oracle}_{\ell+1}$.
Step 5. Model adjustment (serious step).
        If $y^\ell$ is a serious step, then:
        Define the next stability center: $(x^{k+1}, f(x^{k+1}), c(x^{k+1})) := (y^\ell, f_\ell, c_\ell)$.
        Update the linearization errors for $i \in \mathcal{B}^{\rm oracle}_{\ell+1}$ using (3.5) in Lemma 3.2.
        Update the aggregate errors for $i \in \mathcal{B}^{\rm agg}_{\ell+1}$ using (3.6) in Lemma 3.2.
        Set $k = k + 1$.
  Loop. Set $\ell = \ell + 1$ and go to Step 1.
```

Some remarks are in order. Recalling the definition of $h_k(\cdot)$, we conclude that if the descent test is satisfied and a serious step is declared, then it must hold that

$$f(x^{k+1}) - f(x^k) \leq c^+(x^k) - m\delta_\ell \tag{3.11}$$

and

$$c(x^{k+1}) \leq c^+(x^k) - m\delta_\ell. \tag{3.12}$$

In particular, if x^k is infeasible, then $f(x^{k+1}) > f(x^k)$ is possible (since $c^+(x^k) > 0$). Therefore, the method is not monotone with respect to f when outside of the feasible region. However, outside of the feasible region it is monotone with respect to c because $c(x^{k+1}) < c^+(x^k) = c(x^k)$ for x^k infeasible. This seems intuitively reasonable: while it is natural to accept the increase in the objective function value in order to decrease infeasibility, it is not so clear why one would want to decrease the objective function at the expense of moving away from the feasible region. The situation reverses when x^k is feasible. In that case, $c^+(x^k) = 0$ so that $f(x^{k+1}) < f(x^k)$. But although (3.12) implies that x^{k+1} is feasible too, it is possible that $c(x^{k+1}) > c(x^k)$ (except when $c(x^k)$ is exactly zero). This also appears completely reasonable: while preserving feasibility, we allow c to increase (so that the boundary of the feasible set can be approached), while at the same time obtaining a decrease in the objective function.

In Algorithm 3.1, we do not specify any rule for choosing the proximal parameter μ_ℓ. Conditions that μ_ℓ should satisfy for convergence are very mild, and they are stated in the convergence results of section 4. That said, a sound strategy for choosing this parameter is important for computational efficiency. Actually, this is yet another advantage of having our development closely follow the well-established and well-tested unconstrained bundle methods: we can use the update rules for the former, which are already known to perform well in practice; see, e.g., [22, 26, 35].

Subproblem (3.10) is handled by solving its equivalent quadratic programming formulation

$$c^+(x^k) + \begin{cases} \displaystyle\min_{(y,t)\in\mathbb{R}^{n+1}} & t + \dfrac{1}{2}\mu_\ell\|y - x^k\|^2, \\ \quad\text{s.t.} & -e_i^k + \langle g^i_{h_k}, y - x^k\rangle \le t,\ i \in \mathcal{B}_\ell^{\text{oracle}}, \\ & -\hat{\varepsilon}_i^k + \langle \hat{g}^i, y - x^k\rangle \le t,\ i \in \mathcal{B}_\ell^{\text{agg}} \end{cases} \tag{3.13}$$

or the dual of this problem. The dual of (3.13) can be written as a QP on the unit simplex, for which specialized and highly effective methods are available; see, e.g., [10, 19]. The number of variables in the latter is precisely the number of elements in the bundle, which shows the importance of Step 4 of Algorithm 3.1.

The following well-known characterization of the solution of (3.10) follows from [3, Lem. 9.8]. Those relations have already been discussed in section 2, but here we state them in the notation of Algorithm 3.1.

Lemma 3.3. *In the setting of Algorithm* 3.1*, it holds that*

(i) $y^\ell = x^k - \frac{1}{\mu_\ell}\hat{g}^\ell$*, where* $\hat{g}^\ell \in \partial\psi_\ell(y^\ell)$.

(ii) $\hat{g}^\ell \in \partial_{\hat{\varepsilon}_\ell^k} h_k(x^k)$*, where* $\hat{\varepsilon}_\ell^k \ge 0$.

In particular, it follows that $\delta_\ell \ge 0$ in Algorithm 3.1. Moreover, if $\delta_\ell = 0$ for some k, then $\hat{\varepsilon}_\ell^k = 0$ and $\hat{g}^\ell = 0$. Hence, $0 \in \partial h_k(x^k)$, and x^k is a solution to (1.1) by Theorem 2.1.

4. Convergence results.

We assume from now on that the stopping tolerance *tol* is set to zero, $\delta_\ell > 0$ for all ℓ, and thus Algorithm 3.1 does not terminate and generates an infinite sequence of iterates. As usual in the convergence analysis of bundle methods, we consider the following two possible cases: the number of serious steps is either infinite or finite (in the second case, the last generated serious step is followed by an infinite number of null steps).

In what follows, D denotes the feasible set of (1.1), i.e.,

$$D := \{x \in \mathbb{R}^n \mid c(x) \le 0\}.$$

Given an index k of a serious step, let $\ell(k)$ be the index of y^ℓ, which produced this serious step, i.e., $y^{\ell(k)} = x^k$. Finally, the set $\mathcal{L}_s := \{\ell \mid y^\ell \text{ is a serious step}\}$ collects the indices of serious steps in the sequence $\{y^\ell\}$.

Proposition 4.1. *For any serious iteration index* $k_0 \ge 0$*, it holds that*

$$x^k \in \{x \in \mathbb{R}^n \mid c(x) \le c^+(x^{k_0})\} \quad \textit{for all } k \ge k_0.$$

In particular, if $x^{k_1} \in D$ *for some* $k_1 \ge 0$*, then* $x^k \in D$ *for all* $k \ge k_1$.

Proof. Fix an arbitrary $k_0 \ge 0$. If k_0 is the last serious step (i.e., all the subsequent steps are declared null), then the first assertion is trivial.

Suppose now that there exists the (k_0+1)st serious step. If $x^{k_0} \notin D$, then (3.12) implies that $c(x^{k_0+1}) < c^+(x^{k_0}) = c(x^{k_0})$. Furthermore, if $x^k \notin D$ for all $k \ge k_0$, then repeating the above argument we conclude that the sequence $\{c(x^k)\}$ is nonincreasing. In particular, $c(x^k) \le c(x^{k_0}) = c^+(x^{k_0})$ for all $k \ge k_0$.

Suppose now that $x^{k_1} \in D$ for some k_1. If k_1 is the last serious step, the second assertion is trivial. If there exists the (k_1+1)st serious step, then (3.12) implies that $c(x^{k_1+1}) \le -m\delta_{\ell(k_1+1)} < 0$. Using (3.12) recursively, we conclude that $c(x^k) < 0 = c^+(x^{k_1})$ for all $k > k_1$, i.e., $x^k \in D$. Thus the second assertion holds.

Noting that $c(x^k) < c(x^{k_0}) = c^+(x^{k_0})$ for $k_0 \le k \le k_1$, and that $c(x^k) < 0 \le c^+(x^{k_0})$ for $k > k_1$, concludes the proof. □

PROPOSITION 4.2. *Let f be bounded below on D, and suppose that Algorithm* 3.1 *generates an infinite number of serious steps. Then $\{\hat{\varepsilon}_\ell^k\}_{\ell\in\mathcal{L}_s} \to 0$. Furthermore,*

(i) *if*

$$\sum_{\ell\in\mathcal{L}_s} \frac{1}{\mu_\ell} = +\infty, \tag{4.1}$$

then zero is an accumulation point of the sequence $\{\hat{g}^\ell\}_{\ell\in\mathcal{L}_s}$.

(ii) *if for some $\bar{\mu} > 0$ it holds that*

$$\mu_\ell \le \bar{\mu}, \quad \ell \in \mathcal{L}_s, \tag{4.2}$$

then $\{\hat{g}^\ell\}_{\ell\in\mathcal{L}_s} \to 0$.

Proof. We first show that

$$\sum_{\ell\in\mathcal{L}_s} \delta_\ell < +\infty. \tag{4.3}$$

By Proposition 4.1, either $x^k \notin D$ for all k, or there exists some index k_1 such that $x^k \in D$ for all $k \ge k_1$. We examine the two possibilities separately.

In the first case, (3.12) gives that

$$m\delta_{\ell(k+1)} \le c(x^k) - c(x^{k+1}), \quad k \ge 0. \tag{4.4}$$

Thus, the sequence $\{c(x^k)\}$ is decreasing, and since $x^k \notin D$ for all k, this sequence is bounded below (by zero). It follows that it converges to some $\bar{c} \ge 0$ and, furthermore, that $c(x^k) \ge \bar{c}$ for all k. Therefore, summing up the relation (4.4) over all $\ell \in \mathcal{L}_s$, we obtain that

$$\sum_{\ell\in\mathcal{L}_s} \delta_\ell \le \frac{c(x^0) - \bar{c}}{m}.$$

Consider now the second case, i.e., $x^k \in D$ for $k \ge k_1$ (and let k_1 be the first index such that $x^{k_1} \in D$). Then (3.11) yields

$$m\delta_{\ell(k+1)} \le f(x^k) - f(x^{k+1}), \quad k \ge k_1. \tag{4.5}$$

Hence, the sequence $\{f(x^k)\}_{k\ge k_1}$ is decreasing and bounded below by $\bar{f} = \inf\{f(x) \mid x \in D\}$. Recall that for $k < k_1$, $x^k \notin D$, and thus (4.4) holds. Summing up the relations in (4.4) and (4.5), we obtain that

$$\sum_{\ell\in\mathcal{L}_s} \delta_\ell = \sum_{\ell\in\mathcal{L}_s,\ \ell<\ell(k_1)} \delta_\ell + \sum_{\ell\in\mathcal{L}_s,\ \ell\ge\ell(k_1)} \delta_\ell \le \frac{1}{m}(c(x^0) - c(x^{k_1-1}) + f(x^{k_1}) - \bar{f}).$$

This completes the proof of (4.3).

By the definition of δ_ℓ in Algorithm 3.1, for all ℓ it holds that

$$\frac{1}{2\mu_\ell}\|\hat{g}^\ell\|^2 \le \delta_\ell \quad \text{and} \quad \hat{\varepsilon}_\ell^k \le \delta_\ell. \tag{4.6}$$

By (4.3), $\{\delta_\ell\}_{\ell\in\mathcal{L}_s} \to 0$. It immediately follows that $\{\hat{\varepsilon}_\ell^k\}_{\ell\in\mathcal{L}_s} \to 0$. If (4.2) holds, it clearly follows also that $\{\hat{g}^\ell\}_{\ell\in\mathcal{L}_s} \to 0$.

To prove item (i), suppose that the sequence $\{\|\hat{g}^\ell\|\}_{\ell\in\mathcal{L}_s}$ is bounded away from zero. Then, by (4.6) and (4.1), we obtain that $\sum_{\ell\in\mathcal{L}_s}\delta_\ell = +\infty$, in contradiction with (4.3). ∎

We next exhibit the conditions under which the serious iterates are bounded.

Proposition 4.3. *Suppose that problem* (1.1) *has a solution $\bar{x}$ and that Algorithm* 3.1 *generates an infinite sequence of serious steps. Then the sequence $\{x^k\}$ is bounded if either of the following conditions is satisfied:*

(i) *The feasible set D is bounded,*

or

(ii) *there exists some iteration index k_1 such that $f(\bar{x}) \le f(x^k) + c^+(x^k)$ for all $k \ge k_1$ (in particular, this is true if $x^{k_1} \in D$ for some k_1) and $\mu_\ell \ge \hat{\mu}$, $\ell \in \mathcal{L}_s$ for some $\hat{\mu} > 0$.*

Proof. Since D is a level set of c, it follows that if (i) holds, then the convexity of c implies that all the level sets of c are bounded. Boundedness of $\{x^k\}$ now follows from the first assertion of Proposition 4.1.

Suppose now that (ii) holds. (Observe that if $x^{k_1} \in D$ for some k_1, then $x^k \in D$ for all $k \ge k_1$ by Proposition 4.1. In that case, $f(\bar{x}) \le f(x^k) = f(x^k) + c^+(x^k)$ holds automatically.) For $\ell = \ell(k+1)$, we have that

$$\begin{aligned}\|x^{k+1}-\bar{x}\|^2 &= \|x^k-\bar{x}\|^2 - \frac{2}{\mu_\ell}\langle \hat{g}^\ell, x^k-\bar{x}\rangle + \frac{1}{\mu_\ell^2}\|\hat{g}^\ell\|^2\\ &\le \|x^k-\bar{x}\|^2 + \frac{2}{\mu_\ell}\left(h_k(\bar{x}) - h_k(x^k) + \hat{\varepsilon}_\ell^k + \frac{1}{2\mu_\ell}\|\hat{g}^\ell\|^2\right)\\ &= \|x^k-\bar{x}\|^2 + \frac{2}{\mu_\ell}(h_k(\bar{x}) - h_k(x^k) + \delta_\ell),\end{aligned} \tag{4.7}$$

where we have used the fact that $x^{k+1} - x^k = y^\ell - x^k = \hat{g}^\ell/\mu_\ell$ and $\hat{g}^\ell \in \partial_{\hat{\varepsilon}_\ell^k} h_k(x^k)$ (see Lemma 3.3) and the definition of δ_ℓ in Algorithm 3.1.

Observe further that

$$h_k(\bar{x}) - h_k(x^k) = \max\{f(\bar{x}) - f(x^k), c(\bar{x})\} - c^+(x^k).$$

The quantity above is clearly nonpositive if $f(\bar{x}) - f(x^k) - c^+(x^k) \le 0$. This inequality is ensured by the second condition in (ii) for all $k \ge k_1$. In that case, (4.7) (using also that $\mu_\ell \ge \hat{\mu} > 0$) yields

$$\|x^{k+1}-\bar{x}\|^2 \le \|x^k-\bar{x}\|^2 + \frac{2}{\hat{\mu}}\delta_\ell, \quad k \ge k_1,\ \ell = \ell(k+1) \in \mathcal{L}_s. \tag{4.8}$$

By (4.3) and [34, Lem. 2, p. 44], we conclude that the sequence $\{\|x^{k+1}-\bar{x}\|\}$ converges. Hence, the sequence $\{x^k\}$ is bounded. ∎

The assumption that the feasible set of (1.1) is bounded was also imposed in [8, 20, 23, 25, 29]. According to Proposition 4.3, we do not need this assumption if the iterates enter the feasible region. Methods in [17] are all feasible, except for the "phase I–phase II" modifications briefly sketched in [17, Ch. 5.7]. The main convergence result therein is [17, Thm. 5.7.4], which does not assume boundedness of the feasible set, but also does not establish the existence of accumulation points for infeasible sequences of serious steps. Rather, the analysis concerns properties of accumulation points, without claiming their existence.

We next present the final convergence result for the case of the infinite number of serious steps.

THEOREM 4.4. *Assume that* (1.1) *satisfies the Slater constraint qualification and that its solution set is nonempty. Suppose that Algorithm* 3.1 *generates an infinite sequence of serious steps, which is bounded (this holds, for example, under any of the two assumptions of Proposition* 4.3*).*

If condition (4.1) *holds, then the sequence* $\{x^k\}$ *has an accumulation point which is a solution to* (1.1)*.*

If condition (4.2) *holds, then all the accumulation points of* $\{x^k\}$ *are solutions to* (1.1)*.*

If either (4.1) *or* (4.2) *holds, then in the setting of Proposition* 4.3(ii)*, the whole sequence* $\{x^k\}$ *converges to a solution to* (1.1)*.*

Proof. Fix an arbitrary $y \in \mathbb{R}^n$. By Lemma 3.3, for any serious step index k, it holds that

$$h_{x^k}(y) \geq c^+(x^k) + \langle \hat{g}^\ell, y - x^k \rangle - \hat{\varepsilon}_\ell^k, \quad \ell = \ell(k) \in \mathcal{L}_s. \tag{4.9}$$

If (4.1) holds, there exists a subsequence of $\{\hat{g}^\ell\}_{\ell \in \mathcal{L}_s}$ converging to zero (by Proposition 4.2). Also, $\{\hat{\varepsilon}_\ell^k\}_{\ell \in \mathcal{L}_s} \to 0$. Since $\{x^k\}$ is bounded, taking a further subsequence (if necessary), and passing to the limit in (4.9), we obtain that

$$h_{\bar{x}}(y) \geq c^+(\bar{x}) + \langle 0, y - \bar{x} \rangle - 0 = c^+(\bar{x}) = h_{\bar{x}}(\bar{x}),$$

where $\bar{x}$ is an accumulation point of $\{x^k\}$. Since $y \in \mathbb{R}^n$ is arbitrary, the above means that

$$\min\{h_{\bar{x}}(y) \mid y \in \mathbb{R}^n\} = h_{\bar{x}}(\bar{x}) = c^+(\bar{x}). \tag{4.10}$$

According to Theorem 2.1, it remains to prove that

$$h_{\bar{x}}(\bar{x}) = c^+(\bar{x}) = 0.$$

If $c^+(\bar{x}) > 0$, by continuity it holds that $c^+(y) = c(y) > f(y) - f(\bar{x})$ for all y in some neighborhood of $\bar{x}$. Hence, in such a neighborhood, $h_{\bar{x}}(y) = c^+(y)$. It follows from (4.10) that $c^+(\cdot)$ has a local minimum at $\bar{x}$, with $c^+(\bar{x}) > 0$. Since $c^+(\cdot)$ is convex, this minimum must be also global, which contradicts the fact that $D = \{x \in \mathbb{R}^n \mid c^+(x) = 0\} \neq \emptyset$.

If (4.2) holds, then $\{\hat{g}^\ell\}_{\ell \in \mathcal{L}_s} \to 0$, and we can repeat the above argument by passing to the limit along any convergent subsequence of $\{x^k\}$.

Finally, in the setting of Proposition 4.3(ii), we can choose $\bar{x}$ in (4.8) as an accumulation point of $\{x^k\}$, which is a solution to (1.1). Then $\{\|x^k - \bar{x}\|\}$ converges. Since it has a subsequence converging to zero, it must be the case that $\{x^k\} \to \bar{x}$. □

We conclude by considering the case when the number of serious steps is finite, i.e., there exists $\ell ast = \max\{\ell \mid \ell \in \mathcal{L}_s\}$. We denote the corresponding last serious iteration by $k_{\ell ast}$. Then the function $h(\cdot) = h_{x^{k_{\ell ast}}}(\cdot)$ is fixed from that point on, and the algorithm generates only null steps. The fact that $x^{k_{\ell ast}}$ is a solution to (1.1) can be proved similarly to standard results on bundle methods; see, e.g., [7]. Note, however, that unlike [7] we do not assume that the proximal parameter is fixed after the last serious step.

THEOREM 4.5. *Assume that* (1.1) *satisfies the Slater constraint qualification. Suppose that Algorithm* 3.1 *takes a finite number of serious steps. If* $\bar{\mu} \geq \mu_{\ell+1} \geq \mu_\ell$ *for all* $\ell \geq \ell ast$*, then* $x^{k_{\ell ast}}$ *is a solution to* (1.1)*.*

Proof. In what follows, we consider $\ell \geq \ell ast$ and denote $h(\cdot) = h_{x^{k_{\ell ast}}}(\cdot)$. Observe first that for any $y \in \mathbb{R}^n$,

$$\begin{aligned} l_{k,\ell}(y) &= h(x^{k_{\ell ast}}) - \hat{\varepsilon}_\ell^k + \langle \hat{g}^\ell, y - x^{k_{\ell ast}} \rangle \\ &= \psi_\ell(y^\ell) + \mu_\ell \langle x^{k_{\ell ast}} - y^\ell, y - y^\ell \rangle. \end{aligned}$$

In particular, $l_{k,\ell}(y^\ell) = \psi_\ell(y^\ell)$ and, further,

$$\begin{aligned} &l_{k,\ell}(y) + \frac{1}{2}\mu_\ell \|y - x^{k_{\ell ast}}\|^2 \\ &\quad = \psi_\ell(y^\ell) + \frac{1}{2}\mu_\ell \|y^\ell - x^{k_{\ell ast}}\|^2 + \frac{1}{2}\mu_\ell \|y - y^\ell\|^2, \quad y \in \mathbb{R}^n. \end{aligned} \tag{4.11}$$

We have that

$$\begin{aligned} h(x^{k_{\ell ast}}) &\geq \psi_{\ell+1}(x^{k_{\ell ast}}) \\ &\geq \psi_{\ell+1}(y^{\ell+1}) + \frac{1}{2}\mu_{\ell+1} \|y^{\ell+1} - x^{k_{\ell ast}}\|^2 \\ &\geq l_{k,\ell}(y^{\ell+1}) + \frac{1}{2}\mu_\ell \|y^{\ell+1} - x^{k_{\ell ast}}\|^2 \\ &= \psi_\ell(y^\ell) + \frac{1}{2}\mu_\ell \|y^\ell - x^{k_{\ell ast}}\|^2 + \frac{1}{2}\mu_\ell \|y^{\ell+1} - y^\ell\|^2, \end{aligned} \tag{4.12}$$

where the first inequality holds by (2.4a), the second inequality by the definition of $y^{\ell+1}$, the third by (2.4b) and $\mu_{\ell+1} \geq \mu_\ell$, and the equality holds by (4.11).

It follows from (4.12) that the sequence $\{\psi_\ell(y^\ell) + \frac{1}{2}\mu_\ell \|y^\ell - x^{k_{\ell ast}}\|^2\}$ is nondecreasing and bounded above. Hence, it converges. Fixing $y = x^{k_{\ell ast}}$ in (4.11), we have that

$$h(x^{k_{\ell ast}}) \geq l_{k,\ell}(x^{k_{\ell ast}}) = \left(\psi_\ell(y^\ell) + \frac{1}{2}\mu_\ell \|y^\ell - x^{k_{\ell ast}}\|^2 \right) + \frac{1}{2}\mu_\ell \|y^\ell - x^{k_{\ell ast}}\|^2,$$

where the inequality holds by (2.4a) and (2.4b). Since $\{\psi_\ell(y^\ell) + \frac{1}{2}\mu_\ell \|y^\ell - x^{k_{\ell ast}}\|^2\}$ converges, it follows that $\{y^\ell\}$ is bounded. Also, since $\{\mu_\ell\}$ is bounded below by $\mu_{\ell ast} > 0$, (4.12) implies that

$$\{y^{\ell+1} - y^\ell\} \longrightarrow 0, \quad \ell \longrightarrow \infty. \tag{4.13}$$

Since $\{y^\ell\}$ is bounded, the convex function h can be considered Lipschitz-continuous (on the bounded set of interest), and we further have that $\{\hat{g}^\ell\}$ is bounded on that set. Hence,

$$\begin{aligned} L\|y^{\ell+1} - y^\ell\| &\geq h(y^{\ell+1}) - h(y^\ell) \\ &\geq \psi_{\ell+1}(y^{\ell+1}) - h(y^\ell) \\ &\geq \langle \hat{g}^\ell, y^{\ell+1} - y^\ell \rangle, \end{aligned}$$

where the second inequality holds by (2.4a) and the third by (2.4c). Thus (4.13) implies that

$$\{h(y^\ell) - \psi_{\ell+1}(y^{\ell+1})\} \longrightarrow 0, \quad \ell \longrightarrow \infty. \tag{4.14}$$

Let $\bar{y}$ be any accumulation point of $\{y^\ell\}$, i.e., $\{y^{\ell_i}\} \to \bar{y}$ as $i \to \infty$. Note that, by (4.13), we have that $\{y^{\ell_i - 1}\} \to \bar{y}$. Then (4.14) and the continuity of h imply that

$$\{\psi_{\ell_i}(y^{\ell_i})\} \longrightarrow h(\bar{y}), \quad i \longrightarrow \infty. \tag{4.15}$$

Moreover, for any $y \in \mathbb{R}^n$, we have that

$$h(y) \geq \psi_\ell(y) \geq \psi_\ell(y^\ell) + \langle \hat{g}^\ell, y - y^\ell \rangle = \psi_\ell(y^\ell) + \mu_\ell \langle x^{k_{\ell ast}} - y^\ell, y - y^\ell \rangle,$$

where the first inequality holds by (2.4a) and the other relations hold by Lemma 3.3. Passing to the limit along the specified subsequence as $i \to \infty$, and using (4.15), we conclude that

$$h(y) \geq h(\bar{y}) + \tilde{\mu} \langle x^{k_{\ell ast}} - \bar{y}, y - \bar{y} \rangle, \quad y \in \mathbb{R}^n,$$

where $\tilde{\mu}$ is the limit of the (nondecreasing and bounded above) sequence $\{\mu_\ell\}$. It follows that

$$\tilde{\mu}(x^{k_{\ell ast}} - \bar{y}) \in \partial h(\bar{y}), \tag{4.16}$$

or equivalently,

$$\bar{y} \text{ is the solution to } \min_{y \in \mathbb{R}^n} h(y) + \frac{1}{2} \tilde{\mu} \|y - x^{k_{\ell ast}}\|^2.$$

In particular, the latter means that

$$h(\bar{y}) + \frac{1}{2} \tilde{\mu} \|\bar{y} - x^{k_{\ell ast}}\|^2 \leq h(x^{k_{\ell ast}}). \tag{4.17}$$

On the other hand, since the descent test never holds for $\ell \geq \ell ast$, we obtain that

$$\begin{aligned} h(y^\ell) - h(x^{k_{\ell ast}}) &> -m\delta_\ell \\ &= -m \left(h(x^{k_{\ell ast}}) - \psi_\ell(y^\ell) - \frac{1}{2} \mu_\ell \|y^\ell - x^{k_{\ell ast}}\|^2 \right) \\ &\geq -m(h(x^{k_{\ell ast}}) - \psi_\ell(y^\ell)). \end{aligned}$$

Passing to the limit along the specified subsequence as $i \to \infty$, and using (4.15), we obtain that

$$0 \geq (1 - m)(h(x^{k_{\ell ast}}) - h(\bar{y})).$$

As $m \in (0, 1)$, we have that $h(x^{k_{\ell ast}}) \leq h(\bar{y})$. But then (4.17) implies that $\bar{y} = x^{k_{\ell ast}}$. Recalling (4.16), we have that $0 \in \partial h(x^{k_{\ell ast}})$, where $h(\cdot) = h_{x^{k_{\ell ast}}}(\cdot)$. By Theorem 2.1, $x^{k_{\ell ast}}$ is a solution to (1.1). □

5. Preliminary computational experience. For our numerical assessment, we use the following set of academic problems:

- `CHAIN`, a problem that minimizes a linear function over a piecewise quadratic constraint set. The physical interpretation of this problem is to find the equilibrium of a bidimensional chain formed by 20 links. The chain has end points fixed at the

Table 5.1
Some problem data.

Name	n	$f(\bar{x})$	Feasible x^0	Infeasible x^0
`CHAIN`	38	-9.103962328	see (5.1a)	see (5.1b)
`MAXQUAD`	10	-0.368166	$x_i = 0$	$x_i = 10$
`LOCAT`	4	23.88676767	$(15, 22, 26, 11)$	$x_i = 10$
`MINSUM`	6	68.82956	$(0, 0, 0, 0, 3, 0)$	$x_i = 10$
`ROSEN`	4	-44	$x_i = 0$	$(-1, 2, -3, -4)$
`HILBERT`	50	0	–	$x_i = 10$

coordinates $(0, 0)$ and $(1, 0)$. The length of each link should be less than 0.10; see [25, p. 146]. To choose the starting point, we consider two chains lying on the horizontal axis with end points as above, but different lengths:

(5.1a) Feasible: 20 identical links, each one of length 0.10.

(5.1b) Infeasible: 20 identical links, each one of length 0.12.

• `MAXQUAD`, a problem in which the piecewise quadratic objective function is taken from [24, p. 151], and the constraint is given by $c(x) = \max\{\max_{i=1,\dots,10} |x_i| \le 0.05, \sum_{i=1}^{10} x_i \le 0.05\}$.

• `LOCAT`, a minimax location problem of dimension 4 with the objective function given by the maximum of weighted normed functions, and with a piecewise quadratic constraint [4].

• `MINSUM`, a minsum location problem of dimension 6, with the objective function given by a weighted sum of norms, and a linear constraint [4].

• `ROSEN`, the Rosen–Susuki problem from [15, p. 66]. It has dimension 4, solution $\bar{x} = (0, 1, 2, -1)$, $f(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4$, and

$$c(x) = \max \left\{ \begin{matrix} x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 \\ x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10 \\ x_1^2 + x_2^2 + x_3^2 + 2x_1 - x_2 - x_4 - 5 \end{matrix} \right\}.$$

• `HILBERT`, a Hilbert-like feasibility problem of dimension n. For $f(x) \equiv 0$, the constraint is given by $c(x) = \max_{i \le n}\{\max_{k \le n} |\sum_{=1}^{n} \frac{x_j - 1}{i+k+j-2}|\}$, so $\bar{x} = (1, \dots, 1)$ is the solution.

Table 5.1 shows some additional relevant data for the problems, including the dimensionality, optimal value, and starting points. For each of the problems we used two starting points: feasible and infeasible. The exception is `HILBERT`, which is a feasibility problem, and so only an infeasible starting point is of interest.

Since all these problems have known optimal values, the exact improvement function $h_{\bar{x}}$ is available. For comparison purposes, we first solve the unconstrained problem of minimizing $h_{\bar{x}}$ using n1cv2, the proximal bundle method for unconstrained problems described in [26] (with QP subproblems solved by the method described in [19]) and available upon request from www-rocq.inria.fr/estime/modulopt/optimization-routines/n1cv2.html. These runs can be thought of as providing an ideal situation, in which the constrained optimization problem (1.1) is replaced by a single (equivalent) unconstrained problem. The obtained results can therefore be used as a benchmark for icpbm, whose implementation was built on top of n1cv2 and, in

Table 5.2
Summary of results.

	CHAIN		MAXQUAD		LOCAT		MINSUM		ROSEN		HILBERT	
	Iter	Acc	Iter	Acc	Iter	Acc	Iter	Acc	Iter	Acc	Iter	Acc
N1CV2 x^0 feas.	112	4	65	4	22	6	57	6	20	6	-	-
N1CV2 x^0 infeas.	167	4	97	5	33	6	63	6	22	6	31	7
ICPBM x^0 feas.	210	5	160	6	19	6	86	5	38	5	-	-
ICPBM x^0 infeas.	361	4	241	5	30	6	115	5	37	5	29	5

particular, employs the same warm start-ups and update rules for all parameters, including the crucial update of μ_ℓ.

All runs were performed on a 400 MHz Pentium II with 128 Mb RAM. The size of the bundle was limited to 100 elements. Optimality is declared when

$$\hat{\varepsilon}_\ell^k \leq 10^{-4} \quad \text{and} \quad \|\hat{g}^\ell\|^2 \leq 10^{-8}.$$

We note that the above split stopping criterion is generally preferable to the one based on δ_ℓ, because the split criterion does not depend on μ_ℓ.

Our numerical results are reported in Table 5.2. For each run, and for both algorithms, we give the total number of iterations (i.e., calls to the oracle) and the final accuracy with respect to the (known) optimal value of the problem (i.e., the number of exact digits in the final objective function value). In all cases, the final value of the constraint obtained with ICPBM was less than 10^{-4}.

In our opinion, Table 5.2 shows a reasonable performance of ICPBM. In all the cases, the method succeeds in obtaining a reasonably high accuracy, at the price of less than three times the number of oracle calls required by N1CV2 to solve the "ideal" unconstrained problem of minimizing $h_{\bar{x}}$. Furthermore, the results for HILBERT (whose objective function is constant) confirm that the two codes are almost equally efficient when solving a problem of the same complexity (in that case, in some sense unconstrained). With respect to the influence of starting points, ICPBM's behavior does not seem much affected by the choice of an infeasible x^0. The slowest convergence is observed in CHAIN and MAXQUAD. We conjecture that some undesirable bound interferes with these problems (a similar behavior is observed for N1CV2).

In our opinion, comparing our numerical results with those obtained by other authors is problematic, even if some of our test problems seem similar to theirs. First of all, when discussing numerical results in NSO, an important issue arises, which is sometimes referred to as the "curse of nondifferentiability." Namely, because of discontinuity of the subdifferential, even the same code can produce very different outputs when running on different computational platforms; see [3, p. 102]. This phenomenon, together with the lack of a standard, universally accepted NSO problems library, makes broad numerical comparisons difficult. Thus some caution should be exercised when making the conclusions. Nevertheless, the limited experience reported above suggests that the approach presented in this paper is computationally viable. But to be fair, we should mention that our results appear worse than those reported in [23] for some of the same problems. However, we note that even our results for minimizing the fixed "ideal" unconstrained function $h_{\bar{x}}$ by N1CV2 are worse than what

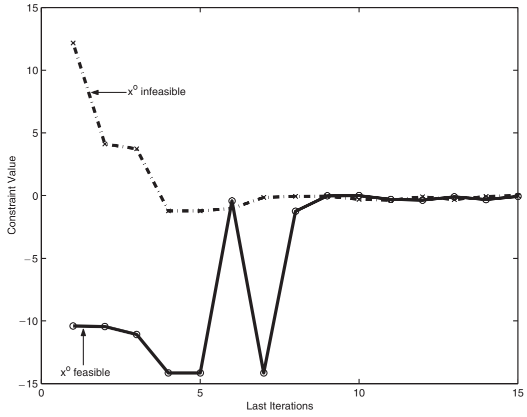


FIG. 5.1. *Constraint values for* `LOCAT`*.*

is reported in [23]. For example, while on average the five algorithms in [23] solve `MAXQUAD` in 33/42 iterations (for the feasible/infeasible starting point, respectively), N1CV2 needs 65/97 iterations to minimize $h_{\bar{x}}$. We do not have a specific explanation, as this could be caused by various implementational differences, all secondary to the ideas of the respective methods themselves: rules for choosing the proximal parameter, linesearch rules, bundle selection rules, oracle rules for choosing subgradients, treating linear constraints and bounds inside or outside of QP subproblems, etc.

In particular, one implementational difference, which can be significant, is that in [23] some linear constraints are inserted into the QP subproblems, while we do not make a distinction between linear and nonlinear constraints. In fact, the results of ours that compare better to the results in [23] are precisely the two problems which do not have any linear constraints, i.e., `LOCAT` and `ROSEN` (19/30 and 38/37 iterations for ICPBM versus 12/15 and 22/30 iterations on average for the five algorithms in [23]). For this reason, we made a more thorough study of the performance of ICPBM on these two problems. Specifically, we analyze whether the algorithm is closely following the (curved) boundary of the feasible set, a behavior that is known to prevent fast convergence. Figures 5.1 and 5.2 show, for `LOCAT` and `ROSEN`, respectively, the (last iterates of the) constraint values generated by ICBPM, starting from feasible and infeasible points.

In Table 5.2 we see that the faster convergence is achieved for `LOCAT`, which, as shown in Figure 5.1, is not generating iterates close to the boundary of the feasible set. By contrast, this phenomenon does occur in `ROSEN`: note that the scale in the vertical axis of Figure 5.1 is 10 times larger than in Figure 5.2.

We next analyze the effect of constraint scaling, which is a general concern in constrained optimization. We ran ICPBM on nine instances of the test problem `LOCAT`, with the constraints multiplied by a factor in the range $\{0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100\}$. We kept the same stopping tolerances as for the unscaled problem and obtained, for all instances, between 6 and 9 digits of accuracy, with (unscaled) constraint values of the order of 10^{-3} or better.

Figure 5.3 shows, for both feasible and infeasible starting points, ICPBM's total number of iterations in relation to the scaling factor, displayed in the semilogarithmic scale. As expected, the number of iterations increases as the factor gets bigger.

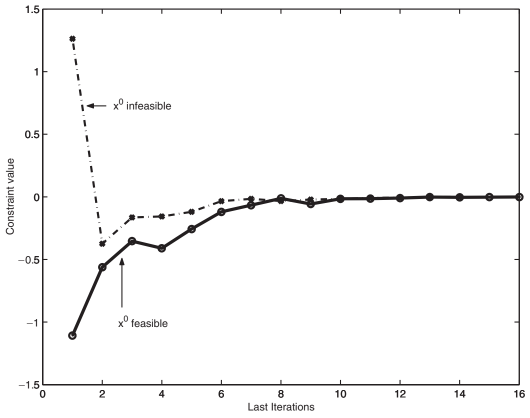


FIG. 5.2. *Constraint values for* ROSEN.

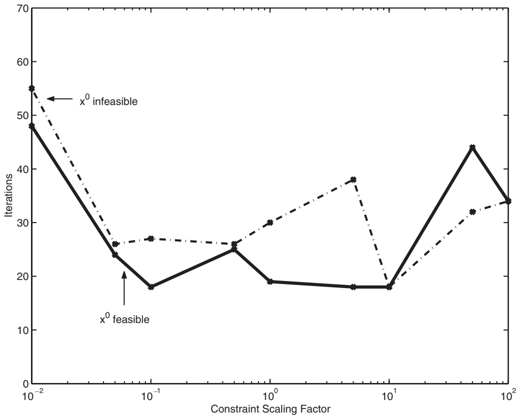


FIG. 5.3. *The effect of scaling.*

However, the overall behavior of the algorithm is not dramatically changed, especially for the infeasible starting point. More precisely, the average number of iterations for the 9 instances with a feasible starting point is 28, about a 50% increase over the 19 iterations for the unscaled case in Table 5.2. In contrast, the infeasible starting point, which could be thought of as more difficult, gave an average number of iterations equal to 32, versus 30 iterations needed for the unscaled case in Table 5.2.

To conclude, we note that in [31, sec. 5] a bundle algorithm for one-dimensional problems is presented, where an appropriate modification in the definition of linearization errors makes directions independent of constraint scaling. Those ideas are also valid in $\mathbb{R}^n$, and therefore could be incorporated in ICPBM, if scaling is of concern.

6. Concluding remarks. We have presented a new idea for handling constraints in nonsmooth convex minimization. Among the features of this approach, which can be useful, we mention the following:

- the method can start from infeasible points;
- the method does not use penalty functions, and thus does not require estimating a suitable value of penalty parameter;

- the method does not use complex filter technologies;
- the method is close in spirit and structure to standard unconstrained bundle methods, and thus can build on the available software and theory (e.g., aggregation and compression techniques);
- convergence is established under mild assumptions.

Our preliminary numerical results show the viability of the method, although implementational improvements are both possible and necessary.

An interesting subject of future research could be an extension of the method to the nonconvex case. This, however, seems to be a nontrivial task, since underlying the method are properties of the improvement function defined by (1.2), which strongly rely on convexity. But if a suitable extension of Theorem 2.1 to the nonconvex case can be found, then one can try to extend the algorithm by using the subgradient locality measures, instead of the linearization errors, along the lines of [17, 29].

Acknowledgments. We thank the two anonymous referees for careful reading of our first version and for constructive suggestions.

REFERENCES

[1] A. Auslender, *Numerical methods for nondifferentiable convex optimization,* Math. Programming Stud., 30 (1987), pp. 102–126.

[2] J.F. Bonnans, J. Ch. Gilbert, C. Lemaréchal, and C. Sagastizábal, *A family of variable metric proximal point methods*, Math. Programming Ser. A, 68 (1995), pp. 15–47.

[3] J.F. Bonnans, J. Ch. Gilbert, C. Lemaréchal, and C. Sagastizábal, *Numerical Optimization. Theoretical and Practical Aspects*, Universitext, Springer-Verlag, Berlin, 2003.

[4] P.N. Brown, *Decay to uniform states in ecological interactions*, SIAM J. Appl. Math., 38 (1980), pp. 22–37.

[5] X. Chen and M. Fukushima, *Proximal quasi-Newton methods for nondifferentiable convex optimization*, Math. Program., 85 (1999), pp. 313–334.

[6] E. Cheney and A. Goldstein, *Newton's method for convex programming and Tchebycheff approximations*, Numer. Math., 1 (1959), pp. 253–268.

[7] R. Correa and C. Lemaréchal, *Convergence of some algorithms for convex minimization*, Math. Program., 62 (1993), pp. 261–275.

[8] R. Fletcher and S. Leyffer, *A Bundle Filter Method for Nonsmooth Nonlinear Optimization*, Numerical Analysis Report NA/195, Department of Mathematics, The University of Dundee, Scotland, 1999.

[9] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Math. Program., 91 (2002), pp. 239–269.

[10] A. Frangioni, *Solving semidefinite quadratic optimization problems within nonsmooth optimization problems*, Comput. Oper. Res., 23 (1996), pp. 1099–1118.

[11] A. Frangioni, *Generalized bundle methods*, SIAM J. Optim., 13 (2002), pp. 117–156.

[12] J.L. Goffin, A. Haurie, and J. Ph. Vial, *Decomposition and nondifferentiable optimization with the projective algorithm*, Management Sci., 38 (1992), pp. 284–302.

[13] M. Hintermüller, *A proximal bundle method based on approximate subgradients*, Comput. Optim. Appl., 20 (2001), pp. 245–266.

[14] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, 2 Vols., Grund. der Math. Wiss. 305–306, Springer-Verlag, Berlin, 1993.

[15] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Econ. Math. Syst. 187, Springer-Verlag, Berlin, 1981.

[16] J.E. Kelley, Jr., *The cutting-plane method for solving convex programs*, J. Soc. Indust. Appl. Math., 8 (1960), pp. 703–712.

[17] K.C. Kiwiel, *Methods of Descent for Nondifferentiable Optimization*, Lecture Notes in Math. 1133, Springer-Verlag, Berlin, 1985.

[18] K.C. Kiwiel, *An exact penalty function algorithm for nonsmooth convex constrained minimization problems*, IMA J. Numer. Anal., 5 (1985), pp. 111–119.

[19] K.C. Kiwiel, *A method for solving certain quadratic programming problems arising in non smooth optimization*, IMA J. Numer. Anal., 6 (1986), pp. 137–152.

[20] K.C. Kiwiel, *A constraint linearization method for nondifferentiable convex minimization*, Numer. Math., 51 (1987), pp. 395–414.

[21] K.C. Kiwiel, *A subgradient selection method for minimizing convex functions subject to linear constraints*, Computing, 39 (1987), pp. 293–305.

[22] K.C. Kiwiel, *Proximity control in bundle methods for convex nondifferentiable minimization*, Math. Program., 46 (1990), pp. 105–122.

[23] K.C. Kiwiel, *Exact penalty functions in proximal bundle methods for constrained convex nondifferentiable minimization*, Math. Program., 52 (1991), pp. 285–302.

[24] C. Lemaréchal and R. Mifflin, *A set of nonsmooth optimization test problems* in Nonsmooth Optimization, C. Lemaréchal and R. Mifflin, eds., Pergamon Press, Oxford, 1978, pp. 151–165.

[25] C. Lemaréchal, A. Nemirovskii, and Yu. Nesterov, *New variants of bundle methods*, Math. Program., 69 (1995), pp. 111–148.

[26] C. Lemaréchal and C. Sagastizábal, *Variable metric bundle methods: From conceptual to implementable forms*, Math. Program., 76 (1997), pp. 393–410.

[27] L. Lukšan and J. Vlček, *Globally convergent variable metric method for convex nonsmooth unconstrained optimization*, J. Optim. Theory Appl., 102 (1999), pp. 593–613.

[28] O.L. Mangasarian, *Nonlinear Programming*, McGraw–Hill, New York, 1969.

[29] R. Mifflin, *An algorithm for constrained optimization with semismooth functions*, Math. Oper. Res., 2 (1977), pp. 191–207.

[30] R. Mifflin, *A modification and extension of Lemarechal's algorithm for nonsmooth minimization*, Math. Programming Stud., 17 (1982), pp. 77–90.

[31] R. Mifflin, *A superlinearly convergent algorithm for one-dimensional constrained minimization problems with convex functions*, Math. Oper. Res., 8 (1983), pp. 185–195.

[32] R. Mifflin, *A quasi-second-order proximal bundle algorithm*, Math. Program., 73 (1996), pp. 51–72.

[33] O. Pironneau and E. Polak, *Rate of convergence of a class of methods of feasible directions*, SIAM J. Numer. Anal., 10 (1973), pp. 161–174.

[34] B.T. Polyak, *Introduction to Optimization*, Optimization Software, Inc., New York, 1987.

[35] P. Rey and C. Sagastizábal, *Dynamical adjustment of the prox-parameter in bundle methods*, Optimization, 51 (2002), pp. 423–447.

[36] H. Schramm and J. Zowe, *A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results*, SIAM J. Optim., 2 (1992), pp. 121–152.

[37] N. Shor, *Minimization Methods for Non-differentiable Functions*, Springer-Verlag, Berlin, 1985.

[38] M.V. Solodov, *On approximations with finite precision in bundle methods for nonsmooth optimization*, J. Optim. Theory Appl., 119 (2003), pp. 151–165.

[39] G. Zoutendijk, *Methods of Feasible Directions: A Study in Linear and Nonlinear Programming*, Elsevier, Amsterdam, 1960.

SIAM J. Optim.
Vol. 16, No. 1, pp. 170–192

A NEW CONJUGATE GRADIENT METHOD WITH GUARANTEED DESCENT AND AN EFFICIENT LINE SEARCH[*]

WILLIAM W. HAGER[†] AND HONGCHAO ZHANG[†]

Abstract. A new nonlinear conjugate gradient method and an associated implementation, based on an inexact line search, are proposed and analyzed. With exact line search, our method reduces to a nonlinear version of the Hestenes–Stiefel conjugate gradient scheme. For any (inexact) line search, our scheme satisfies the descent condition $\mathbf{g}_k^\mathsf{T}\mathbf{d}_k \le -\frac{7}{8}\|\mathbf{g}_k\|^2$. Moreover, a global convergence result is established when the line search fulfills the Wolfe conditions. A new line search scheme is developed that is efficient and highly accurate. Efficiency is achieved by exploiting properties of linear interpolants in a neighborhood of a local minimizer. High accuracy is achieved by using a convergence criterion, which we call the "approximate Wolfe" conditions, obtained by replacing the sufficient decrease criterion in the Wolfe conditions with an approximation that can be evaluated with greater precision in a neighborhood of a local minimum than the usual sufficient decrease criterion. Numerical comparisons are given with both L-BFGS and conjugate gradient methods using the unconstrained optimization problems in the CUTE library.

Key words. conjugate gradient method, unconstrained optimization, convergence, line search, Wolfe conditions

AMS subject classifications. 90C06, 90C26, 65Y20

DOI. 10.1137/030601880

1. Introduction.

We develop a new nonlinear conjugate gradient algorithm for the unconstrained optimization problem

$$\min\{f(\mathbf{x}) : \mathbf{x} \in \Re^n\}, \tag{1.1}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is continuously differentiable. The iterates $\mathbf{x}_0$, $\mathbf{x}_1$, $\mathbf{x}_2$, ... satisfy the recurrence

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k,$$

where the stepsize α_k is positive and the directions $\mathbf{d}_k$ are generated by the rule

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k^N \mathbf{d}_k, \quad \mathbf{d}_0 = -\mathbf{g}_0, \tag{1.2}$$

$$\beta_k^N = \frac{1}{\mathbf{d}_k^\mathsf{T}\mathbf{y}_k}\left(\mathbf{y}_k - 2\mathbf{d}_k \frac{\|\mathbf{y}_k\|^2}{\mathbf{d}_k^\mathsf{T}\mathbf{y}_k}\right)^\mathsf{T} \mathbf{g}_{k+1}. \tag{1.3}$$

Here $\|\cdot\|$ is the Euclidean norm, $\mathbf{g}_k = \nabla f(\mathbf{x}_k)^\mathsf{T}$, and $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$; the gradient $\nabla f(\mathbf{x}_k)$ of f at $\mathbf{x}_k$ is a row vector and $\mathbf{g}_k$ is a column vector. If f is a quadratic and α_k is chosen to achieve the exact minimum of f in the direction $\mathbf{d}_k$, then $\mathbf{d}_k^\mathsf{T}\mathbf{g}_{k+1} = 0$, and the formula (1.3) for β_k^N reduces to the Hestenes–Stiefel scheme [22]. In this paper, however, we consider general nonlinear functions and an inexact line search.

As explained in our survey paper [19], the nonlinear conjugate gradient scheme developed and analyzed in this paper is one member of a one-parameter family of conjugate gradient methods with guaranteed descent. Different choices for the parameter

[*] Received by the editors November 18, 2003; accepted for publication December 10, 2004; published electronically September 8, 2005. This research was supported by National Science Foundation grant 0203270.

http://www.siam.org/journals/siopt/16-1/60188.html

[†] Department of Mathematics, University of Florida, Gainesville, FL 32611-8105 (hager@math.ufl.edu, http://www.math.ufl.edu/∼hager; hzhang@math.ufl.edu, http://www.math.ufl.edu/∼hzhang).

correspond to differences in the relative importance of conjugacy versus descent. The specific scheme analyzed in this paper is closely connected with the memoryless quasi-Newton scheme of Perry [30] and Shanno [36]. In particular, the scheme (1.2)–(1.3) can be obtained by deleting a term in the Perry–Shanno scheme. If $\mathbf{d}_{k+1}$ is the direction generated by the new scheme (1.2)–(1.3), then the direction $\mathbf{d}_{k+1}^{PS}$ of the Perry–Shanno scheme can be expressed as

$$\mathbf{d}_{k+1}^{PS} = \frac{\mathbf{y}_k^\mathsf{T}\mathbf{s}_k}{\|\mathbf{y}_k\|^2}\left(\mathbf{d}_{k+1} + \frac{\mathbf{d}_k^\mathsf{T}\mathbf{g}_{k+1}}{\mathbf{d}_k^\mathsf{T}\mathbf{y}_k}\mathbf{y}_k\right), \tag{1.4}$$

where $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$. We observe in section 2 that the $\mathbf{d}_{k+1}$ term in (1.4) dominates the $\mathbf{y}_k$ term to the right when the cosine of the angle between $\mathbf{d}_k$ and $\mathbf{g}_{k+1}$ is sufficiently small and f is strongly convex. In this case, the directions generated by the new scheme are approximate multiples of $\mathbf{d}_{k+1}^{PS}$. The Perry–Shanno scheme, analyzed further in [34, 37, 39], has global convergence for convex functions and for an inexact line search [36], but in general, it does not necessarily converge, even when the line search is exact [33]. Of course, the Perry–Shanno scheme is convergent if restarts are employed; however, the speed of convergence can decrease. Han, Liu, and Yin [21] proved that if a standard Wolfe line search is employed, then convergence to a stationary point is achieved when $\lim_{k\to\infty} \|\mathbf{y}_k\|_2 = 0$ and the gradient of f is Lipschitz continuous.

Although we are able to prove a global convergence result for (1.2)–(1.3) when f is strongly convex, our analysis breaks down for a general nonlinear function since β_k^N can be negative. Similar to the approach [13, 20, 38] taken for the Polak–Ribière–Polyak [31, 32] version of the conjugate gradient method, we establish convergence for general nonlinear functions by restricting the lower value of β_k^N. Although restricting β_k^N to be nonnegative ensures convergence, the resulting iterates may differ significantly from those of (1.2)–(1.3), and convergence speed may be reduced, especially when f is quadratic. In our restricted scheme, we dynamically adjust the lower bound on β_k^N in order to make the lower bound smaller as the iterates converge:

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \bar{\beta}_k^N \mathbf{d}_k, \quad \mathbf{d}_0 = -\mathbf{g}_0, \tag{1.5}$$

$$\bar{\beta}_k^N = \max\left\{\beta_k^N, \eta_k\right\}, \quad \eta_k = \frac{-1}{\|\mathbf{d}_k\| \min\{\eta, \|\mathbf{g}_k\|\}}, \tag{1.6}$$

where $\eta > 0$ is a constant; we took $\eta = .01$ in the experiments of section 5.

For this modified scheme, we prove a global convergence result with inexact line search. When $\|\mathbf{g}_k\|$ tends to zero as k grows, it follows that η_k in (1.6) tends to $-\infty$ as k grows when $\mathbf{d}_k$ is bounded. Moreover, for strongly convex functions, we show that $\mathbf{d}_k$ is bounded. In this case, where $\mathbf{d}_k$ is bounded, the scheme (1.5)–(1.6) is essentially the scheme (1.2)–(1.3) when k is large since η_k tends to $-\infty$.

Another method related to (1.2)–(1.3) is the Dai–Liao version [7] of the conjugate gradient method, in which β_k^N in (1.2) is replaced with

$$\beta_k^{DL} = \frac{1}{\mathbf{d}_k^\mathsf{T}\mathbf{y}_k}(\mathbf{y}_k - t\mathbf{s}_k)^\mathsf{T}\mathbf{g}_{k+1}, \tag{1.7}$$

where $t > 0$ is a constant parameter. Numerical results are reported in [7] for $t = 0.1$ and $t = 1$; for different choices of t, the numerical results are quite different. The method (1.2)–(1.3) can be viewed as an adaptive version of (1.7) corresponding to $t = 2\|\mathbf{y}_k\|^2/\mathbf{s}_k^\mathsf{T}\mathbf{y}_k$.

With conjugate gradient methods, the line search typically requires sufficient accuracy to ensure that the search directions yield descent [6, 16]. Moreover, it has been shown [9] that for the Fletcher–Reeves [12] and Polak–Ribière–Polyak [31, 32] conjugate gradient methods, a line search that satisfies the strong Wolfe conditions may not yield a direction of descent for a suitable choice of the Wolfe line search parameters, even for the function $f(\mathbf{x}) = \lambda\|\mathbf{x}\|^2$, where $\lambda > 0$ is a constant. An attractive feature of the new conjugate gradient scheme, which we now establish, is that the search directions always yield descent when $\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k \neq 0$, a condition which is satisfied when f is strongly convex, or the line search satisfies the Wolfe conditions.

THEOREM 1.1. *If* $\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k \neq 0$ *and*

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \tau\mathbf{d}_k, \quad \mathbf{d}_0 = -\mathbf{g}_0, \tag{1.8}$$

for any $\tau \in [\beta_k^N, \max\{\beta_k^N, 0\}]$, *then*

$$\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_{k+1} \leq -\frac{7}{8}\|\mathbf{g}_{k+1}\|^2. \tag{1.9}$$

Proof. Since $\mathbf{d}_0 = -\mathbf{g}_0$, we have $\mathbf{g}_0^{\mathsf{T}}\mathbf{d}_0 = -\|\mathbf{g}_0\|^2$, which satisfies (1.9). Suppose $\tau = \beta_k^N$. Multiplying (1.8) by $\mathbf{g}_{k+1}^{\mathsf{T}}$, we have

$$\begin{aligned}
\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_{k+1} &= -\|\mathbf{g}_{k+1}\|^2 + \beta_k^N\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k \\
&= -\|\mathbf{g}_{k+1}\|^2 + \mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k\left(\frac{\mathbf{y}_k^{\mathsf{T}}\mathbf{g}_{k+1}}{\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k} - 2\frac{\|\mathbf{y}_k\|^2\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k}{(\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k)^2}\right) \\
&= \frac{\mathbf{y}_k^{\mathsf{T}}\mathbf{g}_{k+1}(\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k)(\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k) - \|\mathbf{g}_{k+1}\|^2(\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k)^2 - 2\|\mathbf{y}_k\|^2(\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k)^2}{(\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k)^2}. \qquad (1.10)
\end{aligned}$$

We apply the inequality

$$\mathbf{u}^{\mathsf{T}}\mathbf{v} \leq \frac{1}{2}(\|\mathbf{u}\|^2 + \|\mathbf{v}\|^2)$$

to the first term in (1.10) with

$$\mathbf{u} = \frac{1}{2}(\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k)\mathbf{g}_{k+1} \quad \text{and} \quad \mathbf{v} = 2(\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k)\mathbf{y}_k$$

to obtain (1.9). On the other hand, if $\tau \neq \beta_k^N$, then $\beta_k^N \leq \tau \leq 0$. After multiplying (1.8) by $\mathbf{g}_{k+1}^{\mathsf{T}}$, we have

$$\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_{k+1} = -\|\mathbf{g}_{k+1}\|^2 + \tau\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k.$$

If $\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k \geq 0$, then (1.9) follows immediately since $\tau \leq 0$. If $\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k < 0$, then

$$\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_{k+1} = -\|\mathbf{g}_{k+1}\|^2 + \tau\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k \leq -\|\mathbf{g}_{k+1}\|^2 + \beta_k^N\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k$$

since $\beta_k^N \leq \tau \leq 0$. Hence, (1.9) follows by our previous analysis. □

By taking $\tau = \beta_k^N$, we see that the directions generated by (1.2)–(1.3) are descent directions when $\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k \neq 0$. Since η_k in (1.6) is negative, it follows that

$$\bar{\beta}_k^N = \max\left\{\beta_k^N, \eta_k\right\} \in [\beta_k^N, \max\{\beta_k^N, 0\}].$$

Hence, the direction given by (1.5) and (1.6) is a descent direction. Dai and Yuan [8, 10] present conjugate gradient schemes with the property that $\mathbf{d}_{k+1}^\mathsf{T}\mathbf{g}_{k+1} < 0$ when $\mathbf{d}_k^\mathsf{T}\mathbf{y}_k > 0$. If f is strongly convex or the line search satisfies the Wolfe conditions, then $\mathbf{d}_k^\mathsf{T}\mathbf{y}_k > 0$ and the Dai–Yuan schemes yield descent. Note that in (1.9) we bound $\mathbf{d}_{k+1}^\mathsf{T}\mathbf{g}_{k+1}$ by $-(7/8)||\mathbf{g}_{k+1}||^2$, while for the schemes [8, 10], the negativity of $\mathbf{d}_{k+1}^\mathsf{T}\mathbf{g}_{k+1}$ is established.

Our paper is organized as follows: In section 2 we prove convergence of (1.2)–(1.3) for strongly convex functions, while in section 3 we prove convergence of (1.5)–(1.6) for more general nonlinear functions. In section 4 we develop a new line search that is both efficient and highly accurate. This line search exploits properties of linear interpolants to achieve rapid convergence of the line search. High accuracy is achieved by replacing the sufficient decrease criterion in the Wolfe conditions with an approximation that can be evaluated with greater precision in a neighborhood of a local minimum. In section 5 we compare the Dolan–Moré [11] performance profile of the new conjugate gradient scheme to the profiles for the L-BFGS (limited memory Broyden–Fletcher–Goldfarb–Shanno) quasi-Newton method [25, 28], the Polak–Ribière–Polyak PRP+ method [13], and the Dai–Yuan schemes [8, 10] using the unconstrained problems in the test problem library CUTE (constrained and unconstrained testing environment) [4].

2. Convergence analysis for strongly convex functions. Although the search directions generated by either (1.2)–(1.3) or (1.5)–(1.6) are always descent directions, we need to constrain the choice of α_k to ensure convergence. We consider line searches that satisfy either the Goldstein conditions [14],

$$\delta_1\alpha_k\mathbf{g}_k^\mathsf{T}\mathbf{d}_k \le f(\mathbf{x}_k+\alpha_k\mathbf{d}_k) - f(\mathbf{x}_k) \le \delta_2\alpha_k\mathbf{g}_k^\mathsf{T}\mathbf{d}_k, \tag{2.1}$$

where $0 < \delta_2 < \frac{1}{2} < \delta_1 < 1$ and $\alpha_k > 0$, or the Wolfe conditions [40, 41],

$$f(\mathbf{x}_k+\alpha_k\mathbf{d}_k) - f(\mathbf{x}_k) \le \delta\alpha_k\mathbf{g}_k^\mathsf{T}\mathbf{d}_k, \tag{2.2}$$

$$\mathbf{g}_{k+1}^\mathsf{T}\mathbf{d}_k \ge \sigma\mathbf{g}_k^\mathsf{T}\mathbf{d}_k, \tag{2.3}$$

where $0 < \delta \le \sigma < 1$. As in [8], we do not require the "strong Wolfe" condition $|\mathbf{g}_{k+1}^\mathsf{T}\mathbf{d}_k| \le -\sigma\mathbf{g}_k^\mathsf{T}\mathbf{d}_k$, which is often used to prove convergence of nonlinear conjugate gradient methods.

LEMMA 2.1. *Suppose that* $\mathbf{d}_k$ *is a descent direction and* ∇f *satisfies the Lipschitz condition*

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}_k)\| \le L\|\mathbf{x}-\mathbf{x}_k\|$$

for all $\mathbf{x}$ *on the line segment connecting* $\mathbf{x}_k$ *and* $\mathbf{x}_{k+1}$*, where* L *is a constant. If the line search satisfies the Goldstein conditions, then*

$$\alpha_k \ge \frac{(1-\delta_1)}{L}\frac{|\mathbf{g}_k^\mathsf{T}\mathbf{d}_k|}{\|\mathbf{d}_k\|^2}. \tag{2.4}$$

If the line search satisfies the Wolfe conditions, then

$$\alpha_k \ge \frac{1-\sigma}{L}\frac{|\mathbf{g}_k^\mathsf{T}\mathbf{d}_k|}{\|\mathbf{d}_k\|^2}. \tag{2.5}$$

Proof. For the convenience of the reader, we include a proof of these well-known results. If the Goldstein conditions hold, then by (2.1) and the mean value theorem, we have

$$\begin{aligned}\delta_1 \alpha_k \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k &\le f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) - f(\mathbf{x}_k) \\ &= \alpha_k \nabla f(\mathbf{x}_k + \xi \mathbf{d}_k) \mathbf{d}_k \\ &\le \alpha_k \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k + L\alpha_k^2 \|\mathbf{d}_k\|^2,\end{aligned}$$

where $\xi \in [0, \alpha_k]$. Rearranging this inequality gives (2.4).

Subtracting $\mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k$ from both sides of (2.3) using the Lipschitz condition gives

$$(\sigma - 1)\mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k \le (\mathbf{g}_{k+1} - \mathbf{g}_k)^{\mathsf{T}} \mathbf{d}_k \le \alpha_k L \|\mathbf{d}_k\|^2.$$

Since $\mathbf{d}_k$ is a descent direction and $\sigma < 1$, (2.5) follows immediately. □

We now prove convergence of the unrestricted scheme (1.2)–(1.3) for the case when f is strongly convex.

Theorem 2.2. *Suppose that f is strongly convex and Lipschitz continuous on the level set*

$$\mathcal{L} = \{\mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) \le f(\mathbf{x}_0)\}. \tag{2.6}$$

That is, there exist constants L and $\mu > 0$ such that

$$\begin{aligned}\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| &\le L\|\mathbf{x} - \mathbf{y}\| \text{ and} \\ \mu\|\mathbf{x} - \mathbf{y}\|^2 &\le (\nabla f(\mathbf{x}) - \nabla f(\mathbf{y}))(\mathbf{x} - \mathbf{y})\end{aligned} \tag{2.7}$$

for all $\mathbf{x}$ and $\mathbf{y} \in \mathcal{L}$. If the conjugate gradient method (1.2)–(1.3) *is implemented using a line search that satisfies either the Wolfe or the Goldstein conditions in each step, then either $\mathbf{g}_k = \mathbf{0}$ for some k, or*

$$\lim_{k\to\infty} \mathbf{g}_k = \mathbf{0}. \tag{2.8}$$

Proof. Suppose that $\mathbf{g}_k \ne \mathbf{0}$ for all k. By the strong convexity assumption,

$$\mathbf{y}_k^{\mathsf{T}} \mathbf{d}_k = (\mathbf{g}_{k+1} - \mathbf{g}_k)^{\mathsf{T}} \mathbf{d}_k \ge \mu \alpha_k \|\mathbf{d}_k\|^2. \tag{2.9}$$

Theorem 1.1 and the assumption $\mathbf{g}_k \ne \mathbf{0}$ imply that $\mathbf{d}_k \ne \mathbf{0}$. Since $\alpha_k > 0$, it follows from (2.9) that $\mathbf{y}_k^{\mathsf{T}} \mathbf{d}_k > 0$. Since f is strongly convex over $\mathcal{L}$, f is bounded from below. After summing over k the upper bound in either (2.1) or (2.2), we conclude that

$$\sum_{k=0}^{\infty} \alpha_k \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k > -\infty.$$

Combining this with the lower bound for α_k given in Lemma 2.1 and the descent property (1.9) gives

$$\sum_{k=0}^{\infty} \frac{\|\mathbf{g}_k\|^4}{\|\mathbf{d}_k\|^2} < \infty. \tag{2.10}$$

By Lipschitz continuity (2.7),

$$\|\mathbf{y}_k\| = \|\mathbf{g}_{k+1} - \mathbf{g}_k\| = \|\nabla f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) - \nabla f(\mathbf{x}_k)\| \le L\alpha_k \|\mathbf{d}_k\|. \tag{2.11}$$

Utilizing (2.9) and (1.3), we have

$$
\begin{aligned}
|\beta_k^N| &= \left| \frac{\mathbf{y}_k^\mathsf{T}\mathbf{g}_{k+1}}{\mathbf{d}_k^\mathsf{T}\mathbf{y}_k} - 2\frac{\|\mathbf{y}_k\|^2\mathbf{d}_k^\mathsf{T}\mathbf{g}_{k+1}}{(\mathbf{d}_k^\mathsf{T}\mathbf{y}_k)^2} \right| \\
&\le \frac{\|\mathbf{y}_k\|\|\mathbf{g}_{k+1}\|}{\mu\alpha_k\|\mathbf{d}_k\|^2} + 2\frac{\|\mathbf{y}_k\|^2\|\mathbf{d}_k\|\|\mathbf{g}_{k+1}\|}{\mu^2\alpha_k^2\|\mathbf{d}_k\|^4} \\
&\le \frac{L\alpha_k\|\mathbf{d}_k\|\|\mathbf{g}_{k+1}\|}{\mu\alpha_k\|\mathbf{d}_k\|^2} + 2\frac{L^2\alpha_k^2\|\mathbf{d}_k\|^3\|\mathbf{g}_{k+1}\|}{\mu^2\alpha_k^2\|\mathbf{d}_k\|^4} \\
&\le \left(\frac{L}{\mu} + \frac{2L^2}{\mu^2}\right)\frac{\|\mathbf{g}_{k+1}\|}{\|\mathbf{d}_k\|}. \qquad (2.12)
\end{aligned}
$$

Hence, we have

$$\|\mathbf{d}_{k+1}\| \le \|\mathbf{g}_{k+1}\| + |\beta_k^N|\|\mathbf{d}_k\| \le \left(1 + \frac{L}{\mu} + \frac{2L^2}{\mu^2}\right)\|\mathbf{g}_{k+1}\|.$$

Inserting this upper bound for $\mathbf{d}_k$ in (2.10) yields

$$\sum_{k=1}^{\infty} \|\mathbf{g}_k\|^2 < \infty,$$

which completes the proof. □

We now observe that the directions generated by the new conjugate gradient update (1.2) point approximately in the Perry–Shanno direction (1.4) when f is strongly convex and the cosine of the angle between $\mathbf{d}_k$ and $\mathbf{g}_{k+1}$ is sufficiently small. By (2.9) and (2.11), we have

$$\frac{|\mathbf{d}_k^\mathsf{T}\mathbf{g}_{k+1}|}{|\mathbf{d}_k^\mathsf{T}\mathbf{y}_k|}\|\mathbf{y}_k\| \le \frac{L}{\mu}|\mathbf{u}_k^\mathsf{T}\mathbf{g}_{k+1}| = c_1\epsilon\|\mathbf{g}_{k+1}\|, \qquad (2.13)$$

where $\mathbf{u}_k = \mathbf{d}_k/\|\mathbf{d}_k\|$ is the unit vector in the direction $\mathbf{d}_k$, ϵ is the cosine of the angle between $\mathbf{d}_k$ and $\mathbf{g}_{k+1}$, and $c_1 = L/\mu$. By the definition of $\mathbf{d}_{k+1}$ in (1.2), we have

$$\|\mathbf{d}_{k+1}\|^2 \ge \|\mathbf{g}_{k+1}\|^2 - 2\beta_k^N\mathbf{d}_k^\mathsf{T}\mathbf{g}_{k+1}. \qquad (2.14)$$

By the bound for β_k^N in (2.12),

$$|\beta_k^N\mathbf{d}_k^\mathsf{T}\mathbf{g}_{k+1}| \le c_2|\mathbf{u}_k^\mathsf{T}\mathbf{g}_{k+1}|\|\mathbf{g}_{k+1}\| = c_2\epsilon\|\mathbf{g}_{k+1}\|^2, \qquad (2.15)$$

where c_2 is the constant appearing in (2.12). Combining (2.14) and (2.15), we have

$$\|\mathbf{d}_{k+1}\| \ge \sqrt{1 - 2c_2\epsilon}\|\mathbf{g}_{k+1}\|.$$

This lower bound for $\|\mathbf{d}_{k+1}\|$ and the upper bound (2.13) for the $\mathbf{y}_k$ term in (1.4) imply that the ratio between them is bounded by $c_1\epsilon/\sqrt{1 - 2c_2\epsilon}$. As a result, when ϵ is small, the direction generated by (1.2) is approximately a multiple of the Perry–Shanno direction (1.4).

3. Convergence analysis for general nonlinear functions. Our analysis of (1.5)–(1.6) for general nonlinear functions exploits insights developed by Gilbert and Nocedal in their analysis [13] of the PRP+ scheme. Similar to the approach taken in [13], we establish a bound for the change $\mathbf{u}_{k+1} - \mathbf{u}_k$ in the normalized direction $\mathbf{u}_k = \mathbf{d}_k/\|\mathbf{d}_k\|$, which we use to conclude, by contradiction, that the gradients cannot be bounded away from zero. The following theorem is the analogue of [13, Lem. 4.1]; it differs in the treatment of the direction update formula (1.5).

Lemma 3.1. *If the level set* (2.6) *is bounded and the Lipschitz condition* (2.7) *holds, then for the scheme* (1.5)–(1.6) *and a line search that satisfies the Wolfe conditions* (2.2)–(2.3)*, we have*

$$\mathbf{d}_k \neq \mathbf{0} \quad \textit{for each } k \textit{ and} \quad \sum_{k=0}^{\infty} \|\mathbf{u}_{k+1} - \mathbf{u}_k\|^2 < \infty$$

whenever $\inf \{\|\mathbf{g}_k\| : k \geq 0\} > 0$.

Proof. Define $\gamma = \inf \{\|\mathbf{g}_k\| : k \geq 0\}$. Since $\gamma > 0$ by assumption, it follows from the descent property, Theorem 1.1, that $\mathbf{d}_k \neq \mathbf{0}$ for each k. Since $\mathcal{L}$ is bounded, f is bounded from below, and by (2.2) and (2.5), the following Zoutendijk condition [42] holds:

$$\sum_{k=0}^{\infty} \frac{(\mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k)^2}{\|\mathbf{d}_k\|^2} < \infty.$$

Again, the descent property yields

$$\gamma^4 \sum_{k=0}^{\infty} \frac{1}{\|\mathbf{d}_k\|^2} \leq \sum_{k=0}^{\infty} \frac{\|\mathbf{g}_k\|^4}{\|\mathbf{d}_k\|^2} \leq \frac{64}{49} \sum_{k=0}^{\infty} \frac{(\mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k)^2}{\|\mathbf{d}_k\|^2} < \infty. \tag{3.1}$$

Define the quantities

$$\beta_k^+ = \max\{\bar{\beta}_k^N, 0\}, \quad \beta_k^- = \min\{\bar{\beta}_k^N, 0\}, \quad \mathbf{r}_k = \frac{-\mathbf{g}_k + \beta_{k-1}^- \mathbf{d}_{k-1}}{\|\mathbf{d}_k\|}, \quad \delta_k = \beta_{k-1}^+ \frac{\|\mathbf{d}_{k-1}\|}{\|\mathbf{d}_k\|}.$$

By (1.5)–(1.6), we have

$$\mathbf{u}_k = \frac{\mathbf{d}_k}{\|\mathbf{d}_k\|} = \frac{-\mathbf{g}_k + (\beta_{k-1}^+ + \beta_{k-1}^-)\mathbf{d}_{k-1}}{\|\mathbf{d}_k\|} = \mathbf{r}_k + \delta_k \mathbf{u}_{k-1}.$$

Since the $\mathbf{u}_k$ are unit vectors,

$$\|\mathbf{r}_k\| = \|\mathbf{u}_k - \delta_k \mathbf{u}_{k-1}\| = \|\delta_k \mathbf{u}_k - \mathbf{u}_{k-1}\|.$$

Since $\delta_k > 0$, it follows that

$$\begin{aligned} \|\mathbf{u}_k - \mathbf{u}_{k-1}\| &\leq \|(1 + \delta_k)(\mathbf{u}_k - \mathbf{u}_{k-1})\| \\ &\leq \|\mathbf{u}_k - \delta_k \mathbf{u}_{k-1}\| + \|\delta_k \mathbf{u}_k - \mathbf{u}_{k-1}\| \\ &= 2\|\mathbf{r}_k\|. \end{aligned} \tag{3.2}$$

By the definition of β_k^- and the fact that $\eta_k < 0$ and $\bar{\beta}_k^N \geq \eta_k$ in (1.6), we have the following bound for the numerator of $\mathbf{r}_k$:

$$\begin{aligned}\|-\mathbf{g}_k+\beta_{k-1}^{-}\mathbf{d}_{k-1}\| &\le \|\mathbf{g}_k\| - \min\{\bar\beta_{k-1}^N, 0\}\|\mathbf{d}_{k-1}\| \\ &\le \|\mathbf{g}_k\| - \eta_{k-1}\|\mathbf{d}_{k-1}\| \\ &\le \|\mathbf{g}_k\| + \frac{1}{\|\mathbf{d}_{k-1}\|\min\{\eta,\gamma\}}\|\mathbf{d}_{k-1}\| \\ &\le \Gamma + \frac{1}{\min\{\eta,\gamma\}}, \end{aligned} \tag{3.3}$$

where

$$\Gamma = \max_{\mathbf{x}\in\mathcal{L}} \|\nabla f(\mathbf{x})\|. \tag{3.4}$$

Let c denote the expression $\Gamma + 1/\min\{\eta,\gamma\}$ in (3.3). This bound for the numerator of $\mathbf{r}_k$ coupled with (3.2) gives

$$\|\mathbf{u}_k - \mathbf{u}_{k-1}\| \le 2\|\mathbf{r}_k\| \le \frac{2c}{\|\mathbf{d}_k\|}. \tag{3.5}$$

Finally, by squaring (3.5), summing over k, and utilizing (3.1), we complete the proof. □

THEOREM 3.2. *If the level set* (2.6) *is bounded and the Lipschitz condition* (2.7) *holds, then for the scheme* (1.5)–(1.6) *and a line search that satisfies the Wolfe conditions* (2.2)–(2.3)*, either* $\mathbf{g}_k = \mathbf{0}$ *for some* k*, or*

$$\liminf_{k\to\infty} \|\mathbf{g}_k\| = 0. \tag{3.6}$$

Proof. We suppose that both $\mathbf{g}_k \ne \mathbf{0}$ for all k and $\liminf_{k\to\infty}\|\mathbf{g}_k\| > 0$. In the following, we obtain a contradiction. Defining $\gamma = \inf\,\{\|\mathbf{g}_k\| : k \ge 0\}$, we have $\gamma > 0$ due to (3.6) and the fact that $\mathbf{g}_k \ne \mathbf{0}$ for all k. The proof is divided into the following three steps:

I. *A bound for* $\bar\beta_k^N$. By the Wolfe condition $\mathbf{g}_{k+1}^{\mathsf T}\mathbf{d}_k \ge \sigma\mathbf{g}_k^{\mathsf T}\mathbf{d}_k$, we have

$$\mathbf{y}_k^{\mathsf T}\mathbf{d}_k = (\mathbf{g}_{k+1}-\mathbf{g}_k)^{\mathsf T}\mathbf{d}_k \ge (\sigma-1)\mathbf{g}_k^{\mathsf T}\mathbf{d}_k = -(1-\sigma)\mathbf{g}_k^{\mathsf T}\mathbf{d}_k. \tag{3.7}$$

By Theorem 1.1,

$$-\mathbf{g}_k^{\mathsf T}\mathbf{d}_k \ge \frac{7}{8}\|\mathbf{g}_k\|^2 \ge \frac{7}{8}\gamma^2.$$

Combining this with (3.7) gives

$$\mathbf{y}_k^{\mathsf T}\mathbf{d}_k \ge (1-\sigma)\frac{7}{8}\gamma^2. \tag{3.8}$$

Also, observe that

$$\mathbf{g}_{k+1}^{\mathsf T}\mathbf{d}_k = \mathbf{y}_k^{\mathsf T}\mathbf{d}_k + \mathbf{g}_k^{\mathsf T}\mathbf{d}_k < \mathbf{y}_k^{\mathsf T}\mathbf{d}_k. \tag{3.9}$$

Again, the Wolfe condition gives

$$\mathbf{g}_{k+1}^{\mathsf T}\mathbf{d}_k \ge \sigma\mathbf{g}_k^{\mathsf T}\mathbf{d}_k = -\sigma\mathbf{y}_k^{\mathsf T}\mathbf{d}_k + \sigma\mathbf{g}_{k+1}^{\mathsf T}\mathbf{d}_k. \tag{3.10}$$

Since $\sigma < 1$, we can rearrange (3.10) to obtain

$$\mathbf{g}_{k+1}^{\mathsf T}\mathbf{d}_k \ge \frac{-\sigma}{1-\sigma}\mathbf{y}_k^{\mathsf T}\mathbf{d}_k.$$

Combining this lower bound for $\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k$ with the upper bound (3.9) yields

$$\left|\frac{\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k}{\mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k}\right| \le \max\left\{\frac{\sigma}{1-\sigma}, 1\right\}. \tag{3.11}$$

By the definition of $\bar{\beta}_k^N$ in (1.6), we have

$$\bar{\beta}_k^N = \beta_k^N \text{ if } \beta_k^N \ge 0 \quad \text{and} \quad 0 \ge \bar{\beta}_k^N \ge \beta_k^N \text{ if } \beta_k^N < 0.$$

Hence, $|\bar{\beta}_k^N| \le |\beta_k^N|$ for each k. We now insert the upper bound (3.11) for $|\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k|/|\mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k|$, the lower bound (3.8) for $\mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k$, and the Lipschitz estimate (2.11) for $\mathbf{y}_k$ into the expression (1.3) to obtain

$$\begin{aligned}
|\bar{\beta}_k^N| &\le |\beta_k^N| \\
&\le \frac{1}{|\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k|}\left(|\mathbf{y}_k^{\mathsf{T}}\mathbf{g}_{k+1}| + 2\|\mathbf{y}_k\|^2\frac{|\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k|}{|\mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k|}\right) \\
&\le \frac{8}{7}\frac{1}{(1-\sigma)\gamma^2}\left(L\Gamma\|\mathbf{s}_k\| + 2L^2\|\mathbf{s}_k\|^2\max\left\{\frac{\sigma}{1-\sigma}, 1\right\}\right) \\
&\le C\|\mathbf{s}_k\|, \qquad (3.12)
\end{aligned}$$

where Γ is defined in (3.4), and where C is defined as follows:

$$C = \frac{8}{7}\frac{1}{(1-\sigma)\gamma^2}\left(L\Gamma + 2L^2 D\max\left\{\frac{\sigma}{1-\sigma}, 1\right\}\right), \tag{3.13}$$

$$D = \max\{\|\mathbf{y}-\mathbf{z}\| : \mathbf{y}, \mathbf{z} \in \mathcal{L}\}. \tag{3.14}$$

Here D is the diameter of $\mathcal{L}$.

II. *A bound on the steps* $\mathbf{s}_k$. This is a modified version of [13, Thm. 4.3]. Observe that for any $l \ge k$,

$$\mathbf{x}_l - \mathbf{x}_k = \sum_{j=k}^{l-1} \mathbf{x}_{j+1} - \mathbf{x}_j = \sum_{j=k}^{l-1}\|\mathbf{s}_j\|\mathbf{u}_j = \sum_{j=k}^{l-1}\|\mathbf{s}_j\|\mathbf{u}_k + \sum_{j=k}^{l-1}\|\mathbf{s}_j\|(\mathbf{u}_j - \mathbf{u}_k).$$

By the triangle inequality,

$$\sum_{j=k}^{l-1}\|\mathbf{s}_j\| \le \|\mathbf{x}_l - \mathbf{x}_k\| + \sum_{j=k}^{l-1}\|\mathbf{s}_j\|\|\mathbf{u}_j - \mathbf{u}_k\| \le D + \sum_{j=k}^{l-1}\|\mathbf{s}_j\|\|\mathbf{u}_j - \mathbf{u}_k\|. \tag{3.15}$$

Let Δ be a positive integer, chosen large enough that

$$\Delta \ge 4CD, \tag{3.16}$$

where C and D appear in (3.13) and (3.14). Choose k_0 large enough that

$$\sum_{i \ge k_0}\|\mathbf{u}_{i+1} - \mathbf{u}_i\|^2 \le \frac{1}{4\Delta}. \tag{3.17}$$

By Lemma 3.1, k_0 can be chosen in this way. If $j > k \geq k_0$ and $j - k \leq \Delta$, then by (3.17) and the Cauchy–Schwarz inequality, we have

$$\begin{aligned} \|\mathbf{u}_j - \mathbf{u}_k\| &\leq \sum_{i=k}^{j-1} \|\mathbf{u}_{i+1} - \mathbf{u}_i\| \\ &\leq \sqrt{j-k} \left(\sum_{i=k}^{j-1} \|\mathbf{u}_{i+1} - \mathbf{u}_i\|^2 \right)^{1/2} \\ &\leq \sqrt{\Delta} \left(\frac{1}{4\Delta} \right)^{1/2} = \frac{1}{2}. \end{aligned}$$

Combining this with (3.15) yields

$$\sum_{j=k}^{l-1} \|\mathbf{s}_j\| \leq 2D, \tag{3.18}$$

when $l > k \geq k_0$ and $l - k \leq \Delta$.

III. *A bound on the directions* $\mathbf{d}_l$. By (1.5) and the bound on $\bar{\beta}_k^N$ given in step I, we have

$$\|\mathbf{d}_l\|^2 \leq (\|\mathbf{g}_l\| + |\bar{\beta}_{l-1}^N| \|\mathbf{d}_{l-1}\|)^2 \leq 2\Gamma^2 + 2C^2 \|\mathbf{s}_{l-1}\|^2 \|\mathbf{d}_{l-1}\|^2,$$

where Γ is the bound on the gradient given in (3.4). Defining $S_i = 2C^2 \|\mathbf{s}_i\|^2$, we conclude that for $l > k_0$,

$$\|\mathbf{d}_l\|^2 \leq 2\Gamma^2 \left(\sum_{i=k_0+1}^{l} \prod_{j=i}^{l-1} S_j \right) + \|\mathbf{d}_{k_0}\|^2 \prod_{j=k_0}^{l-1} S_j. \tag{3.19}$$

Above, the product is defined to be 1 whenever the index range is vacuous. Let us consider as follows a product of Δ consecutive S_j, where $k \geq k_0$:

$$\begin{aligned} \prod_{j=k}^{k+\Delta-1} S_j = \prod_{j=k}^{k+\Delta-1} 2C^2 \|\mathbf{s}_j\|^2 &= \left(\prod_{j=k}^{k+\Delta-1} \sqrt{2} C \|\mathbf{s}_j\| \right)^2 \\ &\leq \left(\frac{\sum_{j=k}^{k+\Delta-1} \sqrt{2} C \|\mathbf{s}_j\|}{\Delta} \right)^{2\Delta} \leq \left(\frac{2\sqrt{2} C D}{\Delta} \right)^{2\Delta} \leq \frac{1}{2^\Delta}. \end{aligned}$$

The first inequality above is the arithmetic-geometric mean inequality, the second is due to (3.18), and the third comes from (3.16). Since the product of Δ consecutive S_j is bounded by $1/2^\Delta$, it follows that the sum in (3.19) is bounded, and the bound is independent of l. This bound for $\|\mathbf{d}_l\|$, independent of $l > k_0$, contradicts (3.1). Hence, $\gamma = \liminf_{k \to \infty} \|\mathbf{g}_k\| = 0$. □

4. Line search.

The line search is an important factor in the overall efficiency of any optimization algorithm. Papers focusing on the development of efficient line search algorithms include [1, 2, 16, 24, 26, 27]. The algorithm [27] of Moré and Thuente is used widely; it is incorporated in the L-BFGS limited memory quasi-Newton code of Nocedal and in the PRP+ conjugate gradient code of Liu, Nocedal, and Waltz.

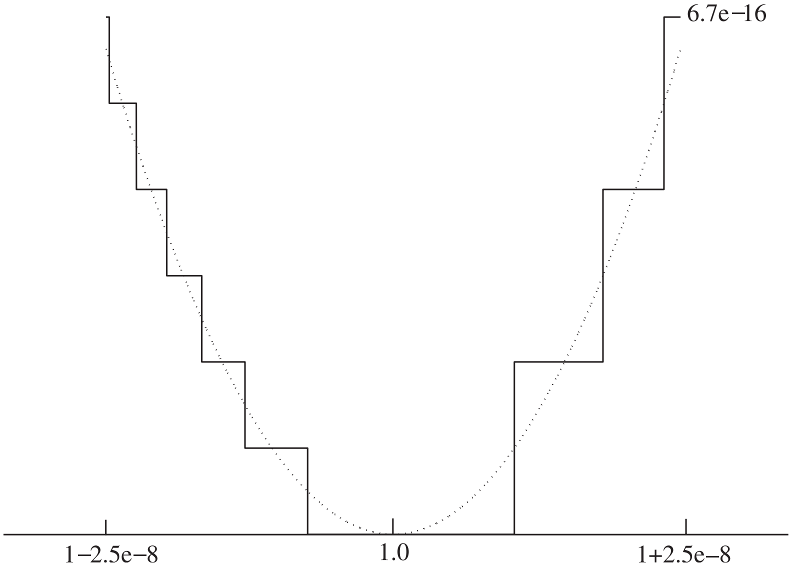


FIG. 4.1. *Numerical and exact graphs of $F(x) = 1 - 2x + x^2$ near $x = 1$.*

The approach we use to find a point satisfying the Wolfe conditions (2.2)–(2.3) is somewhat different from the earlier work cited. To begin, we note that there is a fundamental numerical issue connected with the first Wolfe condition, (2.2). In Figure 4.1 we plot $F(x) = 1 - 2x + x^2$ in a neighborhood of $x = 1$.

The graph, generated by a MATLAB program using a Sun workstation, is obtained by evaluating F at 10,000 values of x between $1 - 2.5 \times 10^{-8}$ and $1 + 2.5 \times 10^{-8}$ and by connecting the computed points on the graph by straight line segments. The true graph is the parabola in Figure 4.1, while the computed graph is piecewise constant.

When devising an algorithm to minimize a smooth function, we often visualize the graph as smooth. But, in actuality, the computer's representation of the function is piecewise constant. Observe that there is an interval of width 1.8×10^{-8} surrounding $x = 1$, where F vanishes. Each point in this interval is a minimizer of the computer's F. In contrast, the true F has a unique minimum at $x = 1$. The interval around $x = 1$, where F is flat, is much wider than the machine epsilon 2.2×10^{-16}. This relatively large flat region is a result of subtracting nearly equal numbers when F is evaluated. In particular, near $x = 1$, $1 - 2x$ is near -1, while x^2 is near $+1$. Hence, when the computer adds $1 - 2x$ to x^2, it is, in essence, subtracting nearly equal numbers. It is well known that there is a large relative error when nearly equal numbers are subtracted; the width of the flat interval near $x = 1$ is on the order of the square root of the machine epsilon (see [15]).

Now consider the function $\phi(\alpha) = f(\mathbf{x}_k + \alpha\mathbf{d}_k)$. If $\phi(0)$ corresponds to a point in the flat part of Figure 4.1 near $x = 1$, then the first Wolfe condition, (2.2), is never satisfied, assuming $\mathbf{d}_k$ is a descent direction, since the right side of (2.2) is always negative and the left side can be only nonnegative. On the other hand, when we compute with 16 significant digits, we would like to be able to compute a solution to the optimization problem with 16-digit accuracy. We can achieve this accuracy by looking for a zero of the derivative. In Figure 4.2 we plot the derivative $F'(x) =$

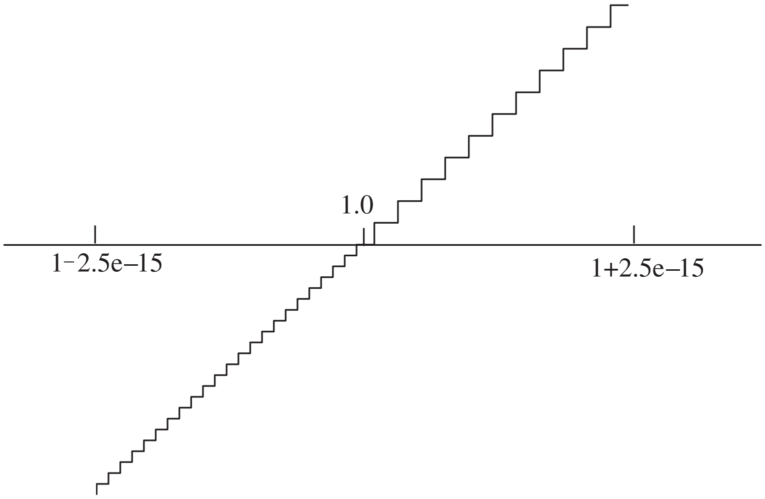


FIG. 4.2. *The numerical graph of the derivative $F'(x) = 2(x-1)$ near $x = 1$.*

$2(x-1)$ of the function in Figure 4.1 near $x = 1$. Since the interval where F' vanishes has width 1.6×10^{-16}, we can locate the zero of F' (Figure 4.2) with accuracy on the order of the machine epsilon 2.2×10^{-16}, while the minimum of F in Figure 4.1 is determined with accuracy on the order of the square root of the machine epsilon. Figures 4.1 and 4.2 are extracted from [17].

This leads us to introduce the *approximate Wolfe conditions*,

$$(2\delta - 1)\phi'(0) \geq \phi'(\alpha_k) \geq \sigma\phi'(0), \tag{4.1}$$

where $\delta < \min\{.5, \sigma\}$. The second inequality in (4.1) is identical to the second Wolfe condition, (2.3). The first inequality in (4.1) is identical to the first Wolfe condition, (2.2), when f is quadratic. For general f, we now show that the first inequality in (4.1) and the first Wolfe condition agree to the order of α_k^2. The interpolating (quadratic) polynomial q that matches $\phi(\alpha)$ at $\alpha = 0$, and $\phi'(\alpha)$ at $\alpha = 0$ and $\alpha = \alpha_k$, is

$$q(\alpha) = \frac{\phi'(\alpha_k) - \phi'(0)}{2\alpha_k}\alpha^2 + \phi'(0)\alpha + \phi(0).$$

For such an interpolating polynomial, $|q(\alpha) - \phi(\alpha)| = O(\alpha^3)$. After replacing ϕ with q in the first Wolfe condition, we obtain the first inequality in (4.1) (with an error term of order α_k^2). We emphasize that this first inequality is an approximation to the first Wolfe condition. On the other hand, this approximation can be evaluated with greater precision than the original condition when the iterates are near a local minimizer, since the approximate Wolfe conditions are expressed in terms of a derivative, not the difference of function values.

With these insights, we terminate the line search when either of the following conditions holds:

T1. The original Wolfe conditions (2.2)–(2.3) are satisfied.

T2. The approximate Wolfe conditions (4.1) are satisfied and

$$\phi(\alpha_k) \le \phi(0) + \epsilon_k, \tag{4.2}$$

where $\epsilon_k \ge 0$ is an estimate for the error in the value of f at iteration k. For the experiments in section 5, we took

$$\epsilon_k = \epsilon |f(\mathbf{x}_k)|, \tag{4.3}$$

where ϵ is a (small) fixed parameter. We would like to satisfy the original Wolfe conditions, so we terminate the line search whenever they are satisfied. On the other hand, when $\mathbf{x}_{k+1}$ and $\mathbf{x}_k$ are close together, numerical errors may make it impossible to satisfy (2.2). If the function value at $\alpha = \alpha_k$ is not much larger than the function value at $\alpha = 0$, then we view the iterates as close together, and we terminate when the approximate Wolfe conditions are satisfied.

We satisfy the termination criterion by constructing a nested sequence of (bracketing) intervals, which converge to a point satisfying either T1 or T2. A typical interval $[a, b]$ in the nested sequence satisfies the following *opposite slope condition*:

$$\phi(a) \le \phi(0) + \epsilon_k, \quad \phi'(a) < 0, \quad \phi'(b) \ge 0. \tag{4.4}$$

Given a parameter $\theta \in (0, 1)$, the *interval update rules* are specified in the following procedure "interval update." The input of this procedure is the current bracketing interval $[a, b]$ and a point c generated by either a secant step or a bisection step, as will be explained shortly. The output of the procedure is the updated bracketing interval $[\bar{a}, \bar{b}]$.

```
INTERVAL UPDATE. $[\bar{a}, \bar{b}]$ = update $(a, b, c)$.
U0. If $c \notin (a, b)$, then $\bar{a} = a$, $\bar{b} = b$, and return.
U1. If $\phi'(c) \ge 0$, then $\bar{a} = a$, $\bar{b} = c$, and return.
U2. If $\phi'(c) < 0$ and $\phi(c) \le \phi(0) + \epsilon_k$, then $\bar{a} = c$, $\bar{b} = b$, and return.
U3. If $\phi'(c) < 0$ and $\phi(c) > \phi(0) + \epsilon_k$, then set $\hat{a} = a$, $\hat{b} = c$, and do the following:
    a. Set $d = (1 - \theta)\hat{a} + \theta\hat{b}$; if $\phi'(d) \ge 0$, then set $\bar{b} = d$, $\bar{a} = \hat{a}$, and return.
    b. If $\phi'(d) < 0$ and $\phi(d) \le \phi(0) + \epsilon_k$, then set $\hat{a} = d$ and go to step a.
    c. If $\phi'(d) < 0$ and $\phi(d) > \phi(0) + \epsilon_k$, then set $\hat{b} = d$ and go to step a.
```

After completing U1–U3, we obtain a new interval $[\bar{a}, \bar{b}] \subset [a, b]$ whose endpoints satisfy (4.4). The loop embedded in U3a–c should terminate since the interval width $\hat{b} - \hat{a}$ tends to zero and, at $\hat{a}$ and $\hat{b}$, the following conditions hold:

$$\begin{aligned} \phi'(\hat{a}) < 0, \quad \phi(\hat{a}) \le \phi(0) + \epsilon_k, \\ \phi'(\hat{b}) < 0, \quad \phi(\hat{b}) > \phi(0) + \epsilon_k. \end{aligned}$$

The input c for the update routine is generated by polynomial interpolation. The interpolation is done in a special way to ensure that the line search interval shrinks quickly. In Figure 4.3, where ϕ' is concave, an initial secant step using function values at a and b yields a point $\bar{b}$ to the right of the zero. A second secant step using function values at $\bar{b}$ and b yields a point $\bar{a}$ to the left of the zero. On the other hand, if ϕ' is convex as shown in Figure 4.4, then an initial secant step using function values at a and b yields a point $\bar{a}$ to the left of the zero. A second secant step using function values at a and $\bar{a}$ yields a point $\bar{b}$ to the right of the zero. Hence, whether ϕ' is convex or concave, a pair of secant steps, implemented in this way, will update one side of the interval, bracketing the zero, and then the other side.

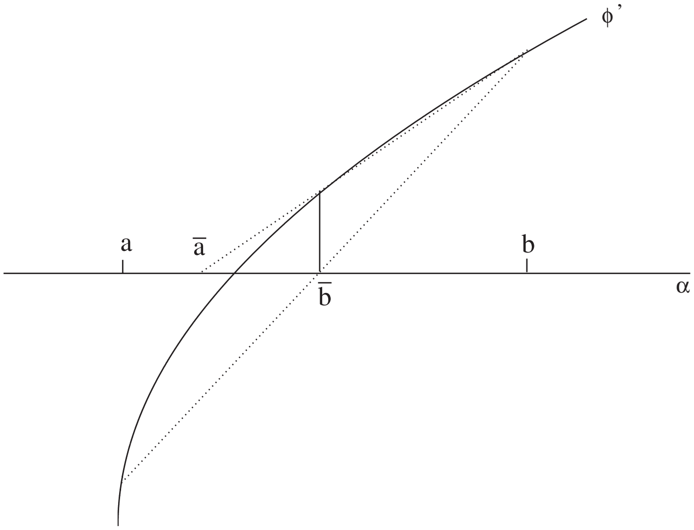


FIG. 4.3. *A pair* of secant *steps applied to a concave* ϕ'.

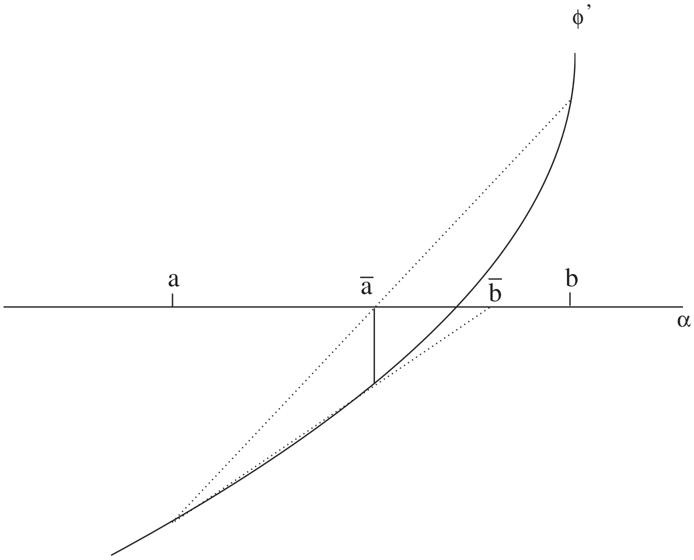


FIG. 4.4. *A pair* of secant *steps applied to a convex* ϕ'.

If c is obtained from a secant step based on function values at a and b, then we write

$$c = \text{ secant } (a, b) = \frac{a\phi'(b) - b\phi'(a)}{\phi'(b) - \phi'(a)}.$$

In general, we do not know whether ϕ' is convex or concave. Consequently, the pair of secant steps is generated by a routine denoted secant^2 defined in the following way.

Double secant step. $[\bar{a},\bar{b}] = \text{secant}^2\ (a,b)$.

S1. $c =$ secant (a,b) and $[A,B] =$ update (a,b,c).

S2. If $c = B$, then $\bar{c} =$ secant (b,B).

S3. If $c = A$, then $\bar{c} =$ secant (a,A).

S4. If $c = A$ or $c = B$, then $[\bar{a},\bar{b}] =$ update $(A,B,\bar{c})$. Otherwise, $[\bar{a},\bar{b}] = [A,B]$.

If we assume the initial interval $[a,b]$ in the secant step satisfies (4.4), then c lies between a and b. If $c = B$, then U1 is satisfied and ϕ' is nonnegative at both b and B. In this case, corresponding to Figure 4.3, we attempt a secant step based on the values of ϕ' at b and B. The attempted secant step fails if $\bar{c}$ lies out the interval $[a,b]$, in which case the update simply returns the initial interval $[A,B]$. If $c = A$ in S3, then U2 is satisfied and ϕ' is negative at both a and A. In this case, corresponding to Figure 4.4, we attempt a secant step based on the values of ϕ' at a and A.

Assuming ϕ is not monotone, an initial interval $[a,b] = [a_0,b_0]$ satisfying (4.4) can be generated by sampling $\phi(\alpha)$ for various choices of α. Starting from this interval, and initializing $k = 0$, we now give a complete statement of the line search used for the numerical experiments in section 5, beginning with a list of the parameters.

Line search/CG_DESCENT *parameters.*

δ - range $(0,.5)$, used in the Wolfe conditions (2.2) and (4.1)

σ - range $[\delta,1)$, used in the Wolfe conditions (2.3) and (4.1)

ϵ - range $[0,\infty)$, used in the approximate Wolfe termination (T2)

θ - range $(0,1)$, used in the update rules when the potential intervals $[a,c]$ or $[c,b]$ violate the opposite slope condition contained in (4.4)

γ - range $(0,1)$, determines when a bisection step is performed (L2 below)

η - range $(0,\infty)$, used in the lower bound for β_k^N in (1.6).

Algorithm. *Line search.*

L0. Terminate the line search if either (T1) or (T2) is satisfied.

L1. $[a,b] =$ secant$^2(a_k,b_k)$.

L2. If $b - a > \gamma(b_k - a_k)$, then $c = (a+b)/2$ and $[a,b] =$ update (a,b,c).

L3. Increment k, set $[a_k,b_k] = [a,b]$, and go to L0.

The line search is terminated whenever a point is generated for which either T1 or T2 holds.

Theorem 4.1. *Suppose that ϕ is continuously differentiable on an interval $[a_0,b_0]$, where* (4.4) *holds. If $\delta < 1/2$, then the line search algorithm terminates at a point satisfying either* T1 *or* T2.

Proof. Due to the bisection step L2, the interval width $b_k - a_k$ tends to zero. Since each interval $[a_k,b_k]$ satisfies the opposite slope condition (4.4), we conclude that $\phi'(a_k)$ approaches 0. Hence, T2 holds for k sufficiently large. □

We now analyze the convergence rate of the secant^2 iteration. Since the root convergence order [29] of the secant method is $(1+\sqrt{5})/2$, the order of convergence for a double secant step is $(1+\sqrt{5})^2/4$. However, the iteration secant^2 is not a conventional double secant step since the most recent iterates are not always used to compute the next iterate; our special secant iteration was devised to first update one side of the bracketing interval and then the other side. This behavior is more attractive than a high convergence order. We now show that the convergence order of secant^2 is $1+\sqrt{2} \approx 2.4$, slightly less than $(1+\sqrt{5})^2/4 \approx 2.6$.

Theorem 4.2. *Suppose that ϕ is three times continuously differentiable near a local minimizer α^*, with $\phi''(\alpha^*) > 0$ and $\phi'''(\alpha^*) \neq 0$. Then for a_0 and b_0 sufficiently close to α^* with $a_0 \le \alpha^* \le b_0$, the iteration*

$$[a_{k+1},b_{k+1}] = \text{ secant}^2(a_k,b_k)$$

converges to α^**. Moreover, the interval width* $|b_k - a_k|$ *tends to zero with root convergence order* $1+\sqrt{2}$*.*

Proof. Suppose that $\phi'''(\alpha^*) > 0$. The case $\phi'''(\alpha^*) < 0$ is treated in a similar way. Our double secant step, as seen in Figure 4.4, is

$$a_{k+1} = \text{ secant } (a_k, b_k) \quad \text{and} \quad b_{k+1} = \text{ secant } (a_k, a_{k+1}). \tag{4.5}$$

It is well known (e.g., see [3, p. 49]) that the error in the secant step $c =$ secant (a, b) can be expressed as

$$c - \alpha^* = (a - \alpha^*)(b - \alpha^*)\frac{\phi'''(\xi)}{2\phi''(\bar{\xi})},$$

where ξ, $\bar{\xi} \in [a, b]$. Hence, for our double secant step, we have

$$\begin{bmatrix} \alpha^* - a_{k+1} \\ b_{k+1} - \alpha^* \end{bmatrix} = \begin{bmatrix} C_k(\alpha^* - a_k)(b_k - \alpha^*) \\ D_k(\alpha^* - a_k)^2(b_k - \alpha^*) \end{bmatrix}, \tag{4.6}$$

where C_k and D_k are constants depending on the second and third derivatives of ϕ near α^*; C_k approaches $\phi'''(\alpha^*)/2\phi''(\alpha^*)$ as a_k and b_k approach α^*, while D_k approaches C_k^2.

Let $\mathbf{E}_k$ denote the error vector

$$\mathbf{E}_k = \begin{bmatrix} a_k - \alpha^* \\ b_k - \alpha^* \end{bmatrix}.$$

Given any $\lambda \in (0, 1)$, it follows from (4.6) that there exists a neighborhood $\mathcal{N}$ of α^* with the property that whenever $a_k < \alpha^* < b_k$ with a_k and $b_k \in \mathcal{N}$, C_k and D_k are bounded and $\|\mathbf{E}_{k+1}\| \le \lambda\|\mathbf{E}_k\|$. Consequently, the iteration (4.5) is convergent whenever $a_0 < \alpha^* < b_0$ with a_0 and $b_0 \in \mathcal{N}$.

Let $\bar{C}$ and $\bar{D}$ denote the maximum values for C_k and D_k, respectively, when a_k and $b_k \in \mathcal{N}$, and consider the following recurrence:

$$\begin{bmatrix} A_{k+1} \\ B_{k+1} \end{bmatrix} = \begin{bmatrix} \bar{C}A_kB_k \\ \bar{D}A_k^2B_k \end{bmatrix}, \quad \text{where} \quad \begin{bmatrix} A_0 \\ B_0 \end{bmatrix} = \begin{bmatrix} \alpha^* - a_0 \\ b_0 - \alpha^* \end{bmatrix}. \tag{4.7}$$

Since $C_k \le \bar{C}$ and $D_k \le \bar{D}$, it follows that $\alpha^* - a_k \le A_k$ and $b_k - \alpha^* \le B_k$ for each k. In other words, A_k and B_k generated by (4.7) bound the error in a_k and b_k, respectively.

Defining the variables

$$v_k = \log(A_k\sqrt{\bar{D}}) \quad \text{and} \quad w_k = \log(\bar{C}B_k),$$

we have

$$\begin{bmatrix} v_{k+1} \\ w_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} v_k \\ w_k \end{bmatrix}. \tag{4.8}$$

The solution is

$$\begin{bmatrix} v_k \\ w_k \end{bmatrix} = \frac{2v_0 + \sqrt{2}w_0}{4}(1+\sqrt{2})^k \begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix} + \frac{-2v_0 + \sqrt{2}w_0}{4}(1-\sqrt{2})^k \begin{bmatrix} -1 \\ \sqrt{2} \end{bmatrix}.$$

Observe that both v_0 and w_0 are negative when a_0 and b_0 are near α. Since $1+\sqrt{2} > |1-\sqrt{2}|$, we conclude that for k large enough,

$$\begin{bmatrix} v_k \\ w_k \end{bmatrix} \le \frac{2v_0 + \sqrt{2}w_0}{8}(1+\sqrt{2})^k \begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix}.$$

Hence, the root convergence order is $1+\sqrt{2}$. Since

$$b_k - a_k \le |b_k - \alpha^*| + |a_k - \alpha^*|,$$

$b_k - a_k$ converges to zero with root convergence order $1+\sqrt{2}$. □

5. Numerical comparisons. In this section we compare the CPU time performance of the new conjugate gradient method, denoted CG_DESCENT, to the L-BFGS limited memory quasi-Newton method of Nocedal [28] and Liu and Nocedal [25], and to other conjugate gradient methods as well. Comparisons based on other metrics, such as number of iterations or number of function/gradient evaluations, are given in [18], where extensive numerical testing of the methods is done. We considered both the PRP+ version of the conjugate gradient method, developed by Gilbert and Nocedal [13], where the β_k associated with the Polak–Ribière–Polyak conjugate gradient method [31, 32] is kept nonnegative, and versions of the conjugate gradient method developed by Dai and Yuan in [8, 10], denoted CGDY and DYHS, which achieve descent for any line search that satisfies the Wolfe conditions (2.2)–(2.3). The hybrid conjugate gradient method DYHS uses

$$\beta_k = \max\{0, \min\{\beta_k^{HS}, \beta_k^{DY}\}\},$$

where β_k^{HS} is the choice of Hestenes and Stiefel [22] and β_k^{DY} appears in [8]. The test problems are the unconstrained problems in the CUTE [4] test problem library.

The L-BFGS and PRP+ codes were obtained from Jorge Nocedal's Web page at http://www.ece.northwestern.edu/∼nocedal/software.html. The L-BFGS code is authored by Jorge Nocedal, while the PRP+ code is coauthored by Guanghui Liu, Jorge Nocedal, and Richard Waltz. In the documentation for the L-BFGS code, it is recommended that between 3 and 7 pairs of vectors be used for the memory. Hence, we chose 5 pairs of vectors for the memory. The line search in both codes is a modification of subroutine CSRCH of Moré and Thuente [27], which employs various polynomial interpolation schemes and safeguards in satisfying the strong Wolfe line search conditions.

We also manufactured a new L-BFGS code by replacing the Moré–Thuente line search with the new line search presented in our paper. We call this new code L-BFGS*. The new line search would need to be modified for use in the PRP+ code to ensure descent. Hence, we retained the Moré–Thuente line search in the PRP+ code. Since the conjugate gradient algorithms of Dai and Yuan achieve descent for any line search that satisfies the Wolfe conditions, we are able to use the new line search in our experiments with CGDY and with DYHS. All codes were written in Fortran and compiled with f77 (default compiler settings) on a Sun workstation.

For our line search algorithm, we used the following values for the parameters:

$$\delta = .1, \quad \sigma = .9, \quad \epsilon = 10^{-6}, \quad \theta = .5, \quad \gamma = .66, \quad \eta = .01.$$

Our rationale for these choices was the following: The constraints on δ and σ are $0 < \delta \le \sigma < 1$ and $\delta < .5$. As δ approaches 0 and σ approaches 1, the line search terminates more quickly. The chosen values $\delta = .1$ and $\sigma = .9$ represent a compromise between our desire for rapid termination and our desire to improve the function value. When using the approximate Wolfe conditions, we would like to achieve decay in the function value, if numerically possible. Hence, we made the small choice $\epsilon = 10^{-6}$ for the error tolerance in (4.3). When restricting β_k in (1.6), we would like to avoid truncation if possible, since the fastest convergence for a quadratic function is obtained when there is no truncation at all. The choice $\eta = .01$ leads to infrequent truncation of β_k. The choice $\gamma = .66$ ensures that the length of the interval $[a, b]$ decreases by a factor of 2/3 in each iteration of the line search algorithm. The choice $\theta = .5$ in the update procedure corresponds to the use of bisection. Our starting guess for step α_k in the line search was obtained by minimizing a quadratic interpolant.

In the first set of experiments, we stopped whenever

$$\text{(5.1)} \qquad \text{(a) } \|\nabla f(\mathbf{x}_k)\|_\infty \le 10^{-6} \quad \text{or} \quad \text{(b) } \alpha_k \mathbf{g}_k^\mathsf{T} \mathbf{d}_k \le 10^{-20} |f(\mathbf{x}_{k+1})|,$$

where $\|\cdot\|_\infty$ denotes the maximum absolute component of a vector. In all but three cases, the iterations stopped when (a) was satisfied—the second criterion essentially says that the estimated change in the function value is insignificant compared to the function value itself.

The CPU time in seconds and the number of iterations, function evaluations, and gradient evaluations for each of the methods are posted on the following Web site: http://www.math.ufl.edu/∼hager/papers/CG. In running the numerical experiments, we checked whether different codes converged to different local minimizers; we only provide data for problems in which all six codes converged to the same local minimizer. The numerical results are now analyzed.

The performance of the six algorithms, relative to CPU time, was evaluated using the profiles of Dolan and Moré [11]. That is, for each method, we plot the fraction P of problems for which the method is within a factor τ of the best time. In Figure 5.1, we compare the performance of the four codes CG_DESCENT, L-BFGS*, L-BFGS, and PRP+. The left side of the figure gives the percentage of the test problems for which a method is the fastest; the right side gives the percentage of the test problems that were successfully solved by each of the methods. The top curve is the method that solved the most problems in a time that was within a factor τ of the best time. Since the top curve in Figure 5.1 corresponds to CG_DESCENT, this algorithm is clearly the fastest for this set of 113 test problems with dimensions ranging from 50 to 10,000. In particular, CG_DESCENT is fastest for about 60% (68 out of 113) of the test problems, and it ultimately solves 100% of the test problems. Since L-BFGS* (fastest for 29 problems) performed better than L-BFGS (fastest for 17 problems), the new line search led to improved performance. Nonetheless, L-BFGS* was still dominated by CG_DESCENT.

In Figure 5.2 we compare the performance of the four conjugate gradient algorithms. Observe that CG_DESCENT is the fastest of the four algorithms. Since CGDY, DYHS, and CG_DESCENT use the same line search, Figure 5.2 indicates that the search direction of CG_DESCENT yields quicker descent than the search directions of CGDY and DYHS. Also, DYHS is more efficient than CGDY. Since each of these six codes differs in the amount of linear algebra required in each iteration and in the relative number of function and gradient evaluations, different codes will be superior in different problem sets. In particular, the fourth ranked PRP+ code in Figure 5.1 still achieved the fastest time in 6 of the 113 test problems.

In our next series of experiments, shown in Table 5.1, we explore the ability of the algorithms and line search to accurately solve the test problems.

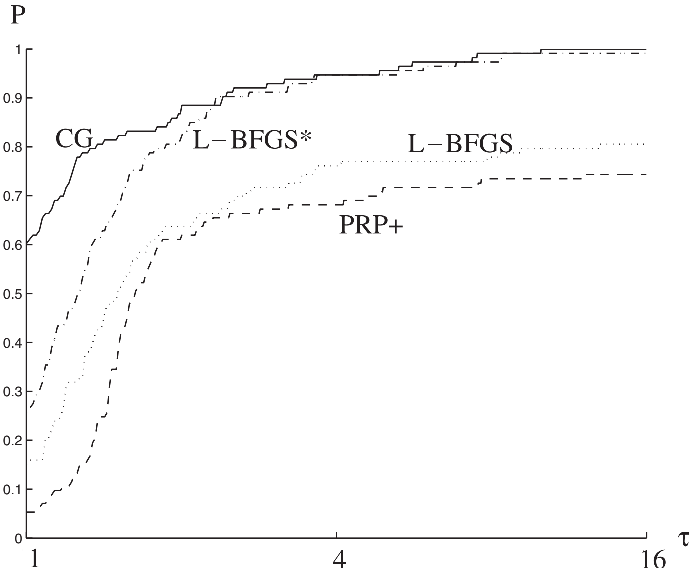


Fig. 5.1. *Performance profiles.*

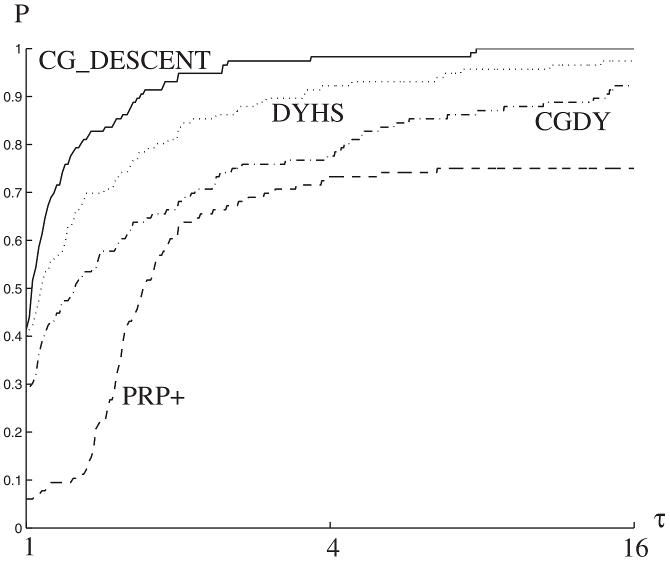


Fig. 5.2. *Performance profiles of conjugate gradient methods.*

In this series of experiments, we repeatedly solve six test problems, increasing the specified accuracy in each run. For the initial run, the stopping condition was $\|\mathbf{g}_k\|_\infty \le 10^{-2}$, and in the last run, the stopping condition was $\|\mathbf{g}_k\|_\infty \le 10^{-12}$. The test problems used in these experiments, and their dimensions, were the following:

Table 5.1
Solution time versus tolerance.

Tolerance	Algorithm	Problem					
$\|\mathbf{g}_k\|_\infty$		#1	#2	#3	#4	#5	#6
10^{-2}	CG_DESCENT	5.22	2.32	0.86	0.00	1.57	10.04
	L-BFGS*	4.19	1.57	0.75	0.01	1.81	14.80
	L-BFGS	4.24	2.01	0.99	0.00	2.46	16.48
	PRP+	6.77	3.55	1.43	0.00	3.04	17.80
10^{-3}	CG_DESCENT	9.20	5.27	2.09	0.00	2.26	17.13
	L-BFGS*	6.72	6.18	2.42	0.01	2.65	19.46
	L-BFGS	6.88	7.46	2.65	0.00	3.30	22.63
	PRP+	12.79	7.16	3.61	0.00	4.26	24.13
10^{-4}	CG_DESCENT	10.79	5.76	5.04	0.00	3.23	25.26
	L-BFGS*	11.56	10.87	6.33	0.01	3.49	31.12
	L-BFGS	12.24	10.92	6.77	0.00	4.11	33.36
	PRP+	15.97	11.40	8.13	0.00	5.01	F
10^{-5}	CG_DESCENT	14.26	7.94	7.97	0.00	4.27	27.49
	L-BFGS*	17.14	16.05	10.21	0.01	4.33	36.30
	L-BFGS	16.60	16.99	10.97	0.00	4.90	F
	PRP+	21.54	12.09	12.31	0.00	6.22	F
10^{-6}	CG_DESCENT	16.68	8.49	9.80	5.71	5.42	32.03
	L-BFGS*	21.43	19.07	14.58	9.01	5.08	46.86
	L-BFGS	21.81	21.08	13.97	7.78	5.83	F
	PRP+	24.58	12.81	15.33	8.07	7.95	F
10^{-7}	CG_DESCENT	20.31	11.47	11.93	5.81	5.93	39.79
	L-BFGS*	26.69	25.74	17.30	12.00	6.10	54.43
	L-BFGS	26.47	F	17.37	9.98	6.39	F
	PRP+	31.17	F	17.34	8.50	9.50	F
10^{-8}	CG_DESCENT	23.22	12.88	14.09	9.68	6.49	47.50
	L-BFGS*	28.18	33.19	20.16	16.58	6.73	63.42
	L-BFGS	32.23	F	20.48	14.85	7.67	F
	PRP+	33.75	F	19.83	F	10.86	F
10^{-9}	CG_DESCENT	27.92	13.32	16.80	12.34	7.46	56.68
	L-BFGS*	32.19	38.51	26.50	26.08	7.67	72.39
	L-BFGS	33.64	F	F	F	8.50	F
	PRP+	F	F	F	F	11.74	F
10^{-10}	CG_DESCENT	33.25	13.89	21.18	13.21	8.11	65.47
	L-BFGS*	34.16	50.60	29.79	33.60	8.22	79.08
	L-BFGS	39.12	F	F	F	9.53	F
	PRP+	F	F	F	F	13.56	F
10^{-11}	CG_DESCENT	38.80	14.38	25.58	13.39	9.12	77.03
	L-BFGS*	36.78	55.70	34.81	39.02	9.14	88.86
	L-BFGS	F	F	F	F	9.99	F
	PRP+	F	F	F	F	14.44	F
10^{-12}	CG_DESCENT	42.51	15.62	27.54	13.38	9.77	78.31
	L-BFGS*	41.73	60.89	39.29	43.95	9.97	101.36
	L-BFGS	F	F	F	F	10.54	F
	PRP+	F	F	F	F	15.96	F

1. FMINSURF (5625)
2. NONCVXU2 (1000)
3. DIXMAANE (6000)
4. FLETCBV2 (1000)
5. SCHMVETT (10000)
6. CURLY10 (1000)

These problems were chosen somewhat randomly; however, we did not include any problem for which the optimal cost was zero. When the optimal cost is zero while the minimizer $\mathbf{x}$ is not zero, the estimate $\epsilon|f(\mathbf{x}_k)|$ for the error in function value (which we used in the previous experiments) can be very poor as the iterates approach the minimizer (where f vanishes). These six problems all have nonzero optimal cost. The times reported in Table 5.1 differ slightly from the times reported at the Web site http://www.math.ufl.edu/∼hager/papers/CG due to timer errors and the fact that the computer runs were done at different times. In Table 5.1, F means that the line search terminated before the convergence tolerance for $\|\mathbf{g}_k\|$ was satisfied. According to the documentation for the line search in the L-BFGS and PRP+ codes, "Rounding errors prevent further progress. There may not be a step which satisfies the sufficient decrease and curvature conditions. Tolerances may be too small."

As can be seen in Table 5.1, the line search based on the Wolfe conditions (used in the L-BFGS and PRP+ codes) fails much sooner than the line search based on both the Wolfe and the approximate Wolfe conditions (used in CG_DESCENT and L-BFGS*). Roughly speaking, a line search based on the Wolfe conditions can compute a solution with accuracy on the order of the square root of the machine epsilon, while a line search that also includes the approximate Wolfe conditions can compute a solution with accuracy on the order of the machine epsilon.

6. Conclusions. We have presented a new conjugate gradient algorithm for solving unconstrained optimization problems. Although the update formulas (1.2)–(1.3) and (1.5)–(1.6) are more complicated than previous formulas, the scheme is relatively robust in numerical experiments. We prove that it satisfies the descent condition $\mathbf{g}_k^\mathsf{T}\mathbf{d}_k \le -\frac{7}{8}\|\mathbf{g}_k\|^2$, independent of the line search procedure, as long as $\mathbf{d}_k^\mathsf{T}\mathbf{y}_k \neq 0$. For (1.5)–(1.6), we prove global convergence under the standard (not strong) Wolfe conditions. A new line search was introduced that utilizes the "approximate Wolfe" conditions; this approximation provides a more accurate way to check the usual Wolfe conditions when the iterates are near a local minimizer. Our line search algorithm exploits a double secant step, denoted secant^2, shown in Figures 4.3 and 4.4, that is designed to achieve rapid decay in the width of the interval which brackets an acceptable step. The convergence order of secant^2, given in Theorem 4.2, is $1 + \sqrt{2}$. The performance profile for our conjugate gradient algorithm (1.5)–(1.6), implemented with our new line search algorithm, was higher than those of the well-established L-BFGS and PRP+ methods for a test set consisting of 113 problems from the CUTE library.

Acknowledgments. Initial experimentation with a version of the line search algorithm was done by Anand Ramasubramaniam in his Master's thesis [35]. The line search used for the numerical experiments of section 5 also exploited a result of Holly Hirst [23], which is roughly the following: For a line search done with "quadratic accuracy," the conjugate gradient method retains its n-step local quadratic convergence property [5]. By [16, Lem. 5], an iteration based on a quadratic interpolant achieves suitable accuracy. Hence, when possible, we always started our line search with a step based on quadratic interpolation. Constructive comments by the referees and by associate editor Jorge Nocedal are gratefully acknowledged.

REFERENCES

[1] M. Al-Baali, *Decent property and global convergence of the Fletcher–Reeves method with in exact line search*, IMA J. Numer. Anal., 5 (1985), pp. 121–124.

[2] M. Al-Baali and R. Fletcher, *An efficient line search for nonlinear least squares*, J. Optim. Theory Appl., 48 (1984), pp. 359–377.

[3] K. E. Atkinson, *An Introduction to Numerical Analysis*, John Wiley, New York, 1978.

[4] I. Bongartz, A. R. Conn, N. I. M. Gould, and P. L. Toint, *CUTE: Constrained and unconstrained testing environments*, ACM Trans. Math. Software, 21 (1995), pp. 123–160.

[5] A. I. Cohen, *Rate of convergence of several conjugate gradient algorithms*, SIAM J. Numer. Anal., 9 (1972), pp. 248–259.

[6] Y. Dai, J. Han, G. Liu, D. Sun, H. Yin, and Y.-X. Yuan, *Convergence properties of nonlinear conjugate gradient methods*, SIAM J. Optim., 10 (1999), pp. 345–358.

[7] Y. H. Dai and L. Z. Liao, *New conjugate conditions and related nonlinear conjugate gradient methods*, Appl. Math. Optim., 43 (2001), pp. 87–101.

[8] Y. H. Dai and Y. Yuan, *A nonlinear conjugate gradient method with a strong global convergence property*, SIAM J. Optim., 10 (1999), pp. 177–182.

[9] Y. H. Dai and Y. Yuan, *Nonlinear Conjugate Gradient Methods*, Shang Hai Science and Technology Publisher, Beijing, 2000.

[10] Y. H. Dai and Y. Yuan, *An efficient hybrid conjugate gradient method for unconstrained optimization*, Ann. Oper. Res., 103 (2001), pp. 33–47.

[11] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Programming, 91 (2002), pp. 201–213.

[12] R. Fletcher and C. Reeves, *Function minimization by conjugate gradients*, Comput. J., 7 (1964), pp. 149–154.

[13] J. C. Gilbert and J. Nocedal, *Global convergence properties of conjugate gradient methods for optimization*, SIAM J. Optim., 2 (1992), pp. 21–42.

[14] A. A. Goldstein, *On steepest descent*, SIAM J. Control, 3 (1965), pp. 147–151.

[15] W. W. Hager, *Applied Numerical Linear Algebra*, Prentice–Hall, Englewood Cliffs, NJ, 1988.

[16] W. W. Hager, *A derivative-based bracketing scheme for univariate minimization and the conjugate gradient method*, Comput. Math. Appl., 18 (1989), pp. 779–795.

[17] W. W. Hager, *Numerical Linear Algebra: Practical Insights and Applications*, in preparation.

[18] W. W. Hager and H. Zhang, *CG_DESCENT, A conjugate gradient method with guaranteed descent*, ACM Trans. Math. Software, to appear.

[19] W. W. Hager and H. Zhang *A survey of nonlinear conjugate gradient methods*, Pacific J. Optim., submitted.

[20] J. Han, G. Liu, D. Sun, and H. Yin, *Two fundamental convergence theorems for nonlinear conjugate gradient methods and their applications*, Acta Math. Appl. Sinica, 17 (2001), pp. 38–46.

[21] J. Y. Han, G. H. Liu, and H. X. Yin, *Convergence of Perry and Shanno's memoryless quasi-Newton method for nonconvex optimization problems*, OR Trans., 1 (1997), pp. 22–28.

[22] M. R. Hestenes and E. L. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 409–436.

[23] H. Hirst, *N-step Quadratic Convergence in the Conjugate Gradient Method*, Ph.D. dissertation, Department of Mathematics, Pennsylvania State University, State College, PA, 1989.

[24] C. Lemarechal, *A view of line-searches*, in Optimization and Optimal Control, Lecture Notes in Control and Inform. Sci. 30, A. Auslender, W. Oettli, and J. Stoer, eds., Springer-Verlag, Heidelberg, 1981, pp. 59–79.

[25] D. C. Liu and J. Nocedal, *On the limited memory BFGS method for large scale optimization*, Math. Programming Ser. B, 45 (1989), pp. 503–528.

[26] J. J. Moré and D. C. Sorensen, *Newton's method*, in Studies in Numerical Analysis, G. H. Golub, ed., Mathematical Association of America, Washington, DC, 1984, pp. 29–82.

[27] J. J. Moré and D. J. Thuente, *Line search algorithms with guaranteed sufficient decrease*, ACM Trans. Math. Software, 20 (1994), pp. 286–307.

[28] J. Nocedal, *Updating quasi-Newton matrices with limited storage*, Math. Comp., 35 (1980), pp. 773–782.

[29] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Classics in Appl. Math. 30, SIAM, Philadelphia, 2000.

[30] J. M. Perry, *A Class of Conjugate Gradient Algorithms with a Two Step Variable Metric Memory*, Discussion paper 269, Center for Mathematical Studies in Economics and Management Science, Northwestern University, Chicago, 1977.

[31] E. Polak and G. Ribière, *Note sur la convergence de méthodes de directions conjuguées*, Rev. Française Informat. Recherche Opérationnelle, 3 (1969), pp. 35–43.

[32] B. T. Polyak, *The conjugate gradient method in extreme problems*, USSR Comp. Math. Math. Phys., 9 (1969), pp. 94–112.

[33] M. J. D. Powell, *Nonconvex minimization calculations and the conjugate gradient method*, in Numerical Analysis, Lecture Notes in Math. 1066, D. F. Griffiths, ed., Springer, Berlin, 1984, pp. 122–141.

[34] M. J. D. Powell, *Restart procedures for the conjugate gradient method*, Math. Programming, 12 (1977), pp. 241–254.

[35] A. Ramasubramaniam, *Unconstrained Optimization by a Globally Convergent High Precision Conjugate Gradient Method*, Master's thesis, Department of Mathematics, University of Florida, Gainesville, FL, 2000.

[36] D. F. Shanno, *On the convergence of a new conjugate gradient algorithm*, SIAM J. Numer. Anal., 15 (1978), pp. 1247–1257.

[37] D. F. Shanno, *Globally convergent conjugate gradient algorithms*, Math. Programming, 33 (1985), pp. 61–67.

[38] C. Wang, J. Han, and L. Wang, *Global convergence of the Polak–Ribière and Hestenes–Stiefel conjugate gradient methods for the unconstrained nonlinear optimization*, OR Trans., 4 (2000), pp. 1–7.

[39] D. F. Shanno and K. H. Phua, *Remark on algorithm* 500, ACM Trans. Math. Software, 6 (1980), pp. 618–622.

[40] P. Wolfe, *Convergence conditions for ascent methods*, SIAM Rev., 11 (1969), pp. 226–235.

[41] P. Wolfe, *Convergence conditions for ascent methods.* II: *Some corrections*, SIAM Rev., 13 (1971), pp. 185–188.

[42] G. Zoutendijk, *Nonlinear programming, computational methods*, in Integer and Nonlinear Programming, J. Abadie, ed., North–Holland, Amsterdam, 1970, pp. 37–86.

SIAM J. Optim.
Vol. 16, No. 1, pp. 193–219

A NEW VERIFIED OPTIMIZATION TECHNIQUE FOR THE "PACKING CIRCLES IN A UNIT SQUARE" PROBLEMS*

MIHÁLY CSABA MARKÓT[†] [‡] AND TIBOR CSENDES[‡]

Abstract. This paper presents a new verified optimization method for the problem of finding the densest packings of nonoverlapping equal circles in a square. In order to provide reliable numerical results, the developed algorithm is based on interval analysis. As one of the most efficient parts of the algorithm, an interval-based version of a previous elimination procedure is introduced. This method represents the remaining areas still of interest as polygons fully calculated in a reliable way. Currently the most promising strategy of finding optimal circle packing configurations is to partition the original problem into subproblems. Still, as a result of the highly increasing number of subproblems, earlier computer-aided methods were not able to solve problem instances where the number of circles was greater than 27. The present paper provides a carefully developed technique resolving this difficulty by eliminating large groups of subproblems together. As a demonstration of the capabilities of the new algorithm the problems of packing 28, 29, and 30 circles were solved within very tight tolerance values. Our verified procedure decreased the uncertainty in the location of the optimal packings by more than 700 orders of magnitude in all cases.

Key words. interval arithmetic, branch-and-bound method, circle packing, optimality proof

AMS subject classifications. 52C15, 52C26, 65G30, 65K05, 90C30

DOI. 10.1137/S1052623403425617

1. Introduction.

The so-called optimal packing of equal circles into a square problem class is an interesting part of geometrical optimization. Circle packing has several real-life applications, e.g., cutting out a given number of identical circle-shaped objects from some kind of material with a minimal amount of waste. In addition, proving the optimality of a packing configuration is a serious theoretical challenge from both the mathematical and the computational points of view.

The paper is organized as follows: In section 2 we discuss the general definition of the problem and give a brief historical overview focusing mostly on the computer-aided methods which form the basis of the present algorithm. Section 3 contains the basic definitions and properties of interval analysis. In section 4 a general interval branch-and-bound (B&B) frame algorithm is introduced. After that, crucial parts of the B&B algorithm are discussed in detail, as those designed specifically for circle packing problems. As a specification step, section 5 introduces a new elimination procedure based on a prior noninterval one known from the literature. In section 6 the questions of finding global solutions are discussed and new methods for eliminating tile combinations are investigated. In section 7 we propose an efficient way of handling occurrences of free (not fixed) circles in optimal packings. Finally, in section 8 the results of the previous sections are applied by solving the packing problems of 28, 29, and 30 circles. The presented numerical results demonstrate how the algorithm works in the successive elimination steps.

*Received by the editors March 28, 2003; accepted for publication (in revised form) December 30, 2004; published electronically September 8, 2005.

http://www.siam.org/journals/siopt/16-1/42561.html

[†] European Space Agency, Advanced Concepts Team, Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands (Mihaly.Csaba.Markot@esa.int).

[‡] University of Szeged, Department of Applied Informatics, H-6701 Szeged, P.O. Box 652, Hungary (csendes@inf.u-szeged.hu).

2. Problem definition and history. The basic problem we consider is the following: *place a given number n of equal circles without overlapping into a unit square maximizing the diameter of the circles as the objective function.* With a slight modification, the problem can be formulated in the following way: *place a given number n of points into the unit square maximizing the minimal distance between the pairs of points.* It can be shown that for a fixed $n \geq 2$ the above two problems are equivalent in the sense that there is a bijective mapping between the feasible solutions of the circle and point packing problems, and, moreover, there exists a strictly monotonically increasing function transforming the objective function of the circle packing problem into the respective objective function of the point packing problem. The proof of the above statements is based on simple geometric transformations (for details see, e.g., [23]). This means that an optimal solution of the circle packing problem is determined by an optimal solution of the point packing problem, and vice versa. Thus, we can consider the more simple point packing problem:

$$\begin{aligned} &\text{maximize} \min_{1\leq i\neq j\leq n} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \\ &\text{s.t.} \quad 0 \leq x_i, y_i \leq 1, \qquad i = 1, 2, \dots, n, \end{aligned} \tag{2.1}$$

where the unit square is $[0,1]^2$, and the ith point is located at (x_i, y_i). The integer $n \geq 2$ is a parameter of the problem class; thus, one can refer to a particular point packing problem instance by specifying n. (Note that we still often call the problem circle packing.)

Due to the monotonicity of the square root function, in practice we solve the problem of maximizing

$$f_n : \mathbb{R}^{2n} \to \mathbb{R}, \quad f_n(x, y) = \min_{1\leq i\neq j\leq n} (x_i - x_j)^2 + (y_i - y_j)^2, \tag{2.2}$$

with $0 \leq x_i, y_i \leq 1,\ i = 1, 2, \dots, n$, saving the evaluation of the interval square root operations. In this way we compute distance values only if they are explicitly required.

Until now, the optimal packings of $2, \dots, 9, 14, 16, 25$, and 36 circles are proved in a theoretical way. On the other hand, computer-assisted optimality proofs exist for $n \leq 20$ [5, 6, 20] and for $21 \leq n \leq 27$ [19]. Recently, [11] reported that the objective function value of the currently known best packings are correct within the tolerance value of 10^{-5} for $n = 10, \dots, 35, 37, 38$ (without determining the location of all the optimizers). These computer methods use floating-point arithmetic and bound rounding errors only during the geometric steps of the algorithms. In contrast, the present paper introduces a fully interval arithmetic–based procedure providing the enclosures of both the possible optimizers and the optimum values with high accuracy. The source code and the control scripts of the algorithm are available at http://www.inf.u-szeged.hu/˜markot/packcirc.htm.

3. Interval analysis. Our algorithm uses interval computations to produce reliable numerical solutions. This requires a brief survey on the basic interval definitions and properties (for more details see [8, 16, 21]).

The set of compact *intervals* is denoted by $\mathbb{I}$, where for all $A \in \mathbb{I}$ intervals $A = [\underline{A}, \overline{A}] = \{a \in \mathbb{R} \mid \underline{A} \leq a \leq \overline{A}\}$. Here $\underline{A}, \overline{A} \in \mathbb{R}$ mean the *lower* and *upper bounds* of A, respectively. In case $\underline{A} = \overline{A}$ we call A a *point interval.* For a given set of reals $D \subseteq \mathbb{R}$, $\mathbb{I}(D)$ denotes the set of all intervals in D. The *width* of an interval is defined by $w(A) := \overline{A} - \underline{A}$.

The real *arithmetic operations* can be extended for intervals by applying the general definition $A \circ B := \{a \circ b \mid a \in A,\ b \in B\}$.

Let $\varphi : D \subseteq \mathbb{R} \to \mathbb{R}$ be a real *elementary function* which is continuous on all $A \in \mathbb{I}(D)$. The interval extension of φ is defined by $\Phi : \mathbb{I}(D) \to \mathbb{I}$, $\Phi(A) := \{\varphi(a) \mid a \in A\}$. The interval extension of a given elementary function can be determined, for example, by invoking monotonicity properties.

A vector of n intervals is called an *n-dimensional interval* (or simply a *box*): $X = (X_1, X_2, \ldots, X_n)$, $X \in \mathbb{I}^n$, and $X_i \in \mathbb{I}$ for $i = 1, 2, \ldots, n$. Moreover, for a given set of n-dimensional vectors $D \subseteq \mathbb{R}^n$, let $\mathbb{I}(D)$ denote the set of n-dimensional boxes in D. The extension of operations and functions for multidimensional intervals is defined componentwise, similarly as for real vectors.

In order to define interval extensions for compound real functions, we define the *interval inclusion functions*. We call $F : \mathbb{I}(D) \to \mathbb{I}$ an *inclusion function* of $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$ if $f(X) = \{f(x) \,|\, x \in X\} \subseteq F(X)$ holds for all $X \in \mathbb{I}(D)$. In the previous definition $f(X)$ denotes the range of f over X. One of the possible ways of constructing such interval functions is the so-called *natural interval extension*: in the real-type function expression the variables are replaced by intervals, and, moreover, the operators and elementary functions are replaced by the corresponding interval ones.

Beyond the theoretical reliability of the interval computations, the inclusion properties should be provided even in cases when finite precision floating-point computer arithmetic is used. Namely, nonrepresentable intervals should be handled to control rounding errors. This task is usually done by the computational environment using exactly representable floating-point numbers (also called machine numbers) and applying directed outward rounding procedures.

4. An interval B&B algorithm. In this section an interval B&B method is introduced for computing all the global maximizers and the f^* maximum value of the global optimization problem

$$\max_{z \in Z_0} f(z), \tag{4.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuous objective function and $Z_0 \in \mathbb{I}^n$ is the search interval. In the algorithm the interval inclusion function $F(Z)$ of $f(z)$ is used. At each iteration cycle (between step 2 and step 11 of Algorithm 1), choose a box Z from the boxes stored in the WorkTree and split it into two parts U^1 and U^2 (step 5). Then for both U^i try to delete some parts of U^i which cannot contain a global optimizer point (steps 7 to 9). If the remaining part of U^i (denoted also by U^i in the algorithm) fulfills the termination criterion, put it into the ResultList (step 10). Otherwise store the remaining part of U^i for further splittings (step 11).

In order to avoid repeated interval function evaluations, each box Z is stored together with $\overline{F}(Z)$. The ResultList is implemented as a single-linked list and is organized according to the first-in, first-out (FIFO) insertion strategy. The boxes still to be processed are placed in a balanced binary search tree, namely, in an AVL-tree called WorkTree. The tree elements are sorted in decreasing order of their $\overline{F}(Z)$ values. Moreover, elements having the same $\overline{F}(Z)$ values are stored at the same node of the tree in a single-linked list. The latter list type has the same properties as the ResultList. The search tree representation offers *Insert*() for insertion, *Delete*() for deletion, and *Head*() for returning the first element stored in the tree. That is, *Head*() implements an *interval selection method* choosing a box with maximal $\overline{F}(Z)$ for subdivision: this method is also known as the Moore–Skelboe selection rule.

Algorithm 1. *Global_Optimization.*

```
Inputs:   – $f$: the objective function,
          – $Z_0$: the search box,
          – $\varepsilon$: tolerance value for the stopping criterion.
Outputs:  – $Maximum$: enclosure of the global maximum value,
          – $ResultList$: set of candidates for a global maximizer.

1:  $Z := Z_0$; $WorkTree := \{(Z, \overline{F}(Z))\}$; Set an initial $\tilde f$ cutoff value.
2:  while ($WorkTree$ is not empty) do
3:     $(Z, \overline{F}(Z)) := Head(WorkTree)$;
4:     $Delete(Head(WorkTree))$;
5:     $Bisection(Z, U^1, U^2)$;
6:     for $i := 1$ to $2$ do
7:        Try to improve $\tilde f$;
8:        Apply accelerating devices for $U^i$;
9:        if ($U^i$ can be deleted as a whole) then continue with the next $i$;
10:       if $(w(F(U^i)) < \varepsilon)$ then $Insert(ResultList, (U^i, \overline{F}(U^i)))$;
11:       else $Insert(WorkTree, (U^i, \overline{F}(U^i)))$;
12: $Maximum := (\tilde f\,, \max\{\overline{F}(Z) \mid (Z, \overline{F}(Z)) \in ResultList\})$;
13: Return $Maximum$, $ResultList$;
```

In the following, we specify some further details of the algorithm such as the evaluation of $F(Z)$, the bisection strategy (step 5), and the accelerating devices (step 8).

An interval inclusion function for the packing circles problems. In [12] an inclusion function was already given, as follows.

THEOREM 4.1 (see [12]). *Let $(X, Y) \subseteq [0,1]^{2n}$, and let $D_{ij} = (X_i - X_j)^2 + (Y_i - Y_j)^2$ for all $i, j \in \{1, 2, \dots, n\}$, $i \neq j$. Then an inclusion function of $f_n(x, y)$ over the $2n$-dimensional box (X, Y) is given by*

$$F_n(X, Y) := \Big[\min_{1 \le i \neq j \le n} \underline{D}_{ij}, \min_{1 \le i \neq j \le n} \overline{D}_{ij}\Big].$$

Bisection step. In step 5 we split the leading box Z perpendicular to its widest component. If two or more components have the same width, we choose the one having the smallest index. This is the classical subdivision method. The investigation of some more sophisticated rules (see, e.g., [3, 15]) for circle packing problems can be a subject of further study.

Accelerating devices. In step 8 several tests are performed to delete some parts of U^i which cannot contain global maximizer points. In some cases the whole box can be rejected. In order to test whether the investigated regions contain a global maximizer, we assume that a guaranteed lower bound $\tilde f$ of the global maximum value exists. In general, e.g., $\tilde f := \underline{F}(Z_0)$ is computed. For practical considerations we will use the notation $\tilde f_0$ and $\tilde f$ as the lower bounds of the objective functions in (2.1) and (2.2), respectively. In the present algorithm the initial $\tilde f$ value was determined as follows: we performed an interval function evaluation (with natural interval extension) over a machine point representation of the currently best-known packing, and set $\tilde f$ as the lower bound of the result interval. The corresponding $\tilde f_0$ was obtained by an interval square root evaluation:

$$f_s := \sqrt{\tilde f} \in \mathbb{I}, \quad \tilde f_0 := \underline{f_s}. \tag{4.2}$$

In Algorithm 1 the following accelerating tests are applied:

(i) *Cutoff test for U^i.* After computing the inclusions $F(U^i)$ over the boxes $U^i, i = 1, 2$, U^i can be eliminated if $\overline{F}(U^i) < \tilde{f}$. Note that if $\tilde{f}$ is improved in step 7, then one can discard all the elements $(Z, \overline{F}(Z))$ of the WorkTree for which $\overline{F}(Z) < \tilde{f}$ holds. As the experiences show, the improvement of $\tilde{f}$ can be a very hard task for circle packing problems, requiring the verified results of some sophisticated search methods (see, e.g., those in [1, 4, 17]). Nevertheless, in cases when the goal is to verify the optimality of a given packing, the utilization of an appropriate initial $\tilde{f}$ can be enough.

(ii) *Elimination of inactive regions within U^i.* Assume we have a validated $\tilde{f}_0$ value. Considering U^i in the form of $(X, Y) \subseteq [0, 1]^{2n}$, one can consider $(X_i, Y_i) \subseteq [0, 1]^2$ as a rectangle in the unit square containing the ith point to be placed. Then from each (X_i, Y_i) we can iteratively delete those points which have a distance smaller than $\tilde{f}_0$ to *all* points of another rectangle $(X_j, Y_j),\ i \neq j$. This procedure may result in either the shrinking or the rejection of U^i. With regard to its importance, this method will be discussed in detail in section 5.

Further interval-based accelerating devices designed for circle packing problems can be found in [12], but those were not included in the present algorithm.

5. Method of active areas using polygons. This accelerating test is known from the literature (see, e.g., [5, 18, 19]) as a part of noninterval-type methods. The key idea of this method was already given in section 4. First, we outline a possible basis algorithm (Algorithm 2): Consider a box $(X, Y) \subseteq [0, 1]^{2n}$. The ith component of X and Y ($(X_i, Y_i) \subseteq [0, 1]^2$) is called the ith *initial active region*, $i = 1, \ldots, n$. During the procedure the R_i active regions of the different components are reduced iteratively, until either one of the active regions becomes empty or a pregiven iteration limit ($It_{\max}$) is reached. In the first case the whole box (X, Y) can be erased (step 5). In the latter case a new box (X', Y') containing the remaining regions will be stored (steps 7 and 8). The most important part of Algorithm 2 is step 4, in which we delete some points (forming a so-called *inactive region*) of R_i having distance smaller than $\tilde{f}$ from each point of R_j.

One crucial part of the algorithm is the representation of the intermediate active areas (i.e., the R_i regions). One can easily show that a set of points within a two-dimensional geometric object having a distance at least $\tilde{f}_0$ from all points of another object may be nonconvex or nonconnected. Nevertheless, a good approximation of the active and inactive point sets is vital to erasing as large inactive sets as possible. In [5] the initial active regions are quantized into many rectangular pieces, applying splittings both in horizontal and in vertical directions, and the set of eliminated and remaining pieces were representing the inactive and active point sets, respectively. In [18] a similar approach was applied but using only splittings in one direction. Until now, the most effective realization is the one of Nurmela and Östergård [19], which approximates the active and inactive regions by polygons. Although in a method working basically with multidimensional intervals the latter solution is more difficult to implement, we found that this extra effort resulted in an outstanding improvement in the computational efficiency as compared to the method using rectangular approximations. Moreover, the branching step of the current B&B frame algorithm generates rectangular splittings in a moderate way, mixing the advantages of the different approaches.

The method of [19] is based on the following lemma and theorem.

Lemma 5.1 (see [19]). *If a point p is at a distance less than $\tilde{f}_0$ from all the*

Algorithm 2. *Method_of_active_areas.*

```
Inputs:  –  $\tilde{f}_0$: a validated lower bound of the global maximum of (2.1),
         –  $(X,Y) \subseteq [0,1]^{2n}$: the box to be reduced,
         –  $It_{\max}$: the iteration limit.
Output:  –  $(X',Y') \subseteq [0,1]^{2n}$: a box containing the remaining areas.
1: for $i := 1$ to $n$ do $R_i := (X_i, Y_i)$;
2: for $i := 1$ to $It_{\max}$ do
3:    for all $(i,j)$, $1 \le i,j \le n$, $i \ne j$ do
4:       $R'_i := Diminish\_ij((R_i, R_j), \tilde{f}_0)$; {comment: reduce $R_i$ to $R'_i$}
5:       if $(R'_i = \emptyset)$ then return "$(X',Y')$ is empty";
6:       $R_i := R'_i$;
7: for $i := 1$ to $n$ do $(X'_i, Y'_i) := Rectangular\_enclosure(R_i)$;
8: return $(X',Y')$;
```

vertices of a polygon R, it is at a distance less than $\tilde{f}_0$ from all points of R.

THEOREM 5.2 (see [19]). *Assume that $p_1, \dots, p_k$ are distinct points on the boundary of a polygon R_i, that the line segments $\overline{p_l p_{l+1}}$ for $2 \le l \le k-2$ are edges of R_i, and that $\overline{p_1 p_2}$ and $\overline{p_{k-1} p_k}$ lay on the edges of R_i. If the points p_i, $1 \le i \le k$, are at a distance less than $\tilde{f}_0$ from all vertices of R_j, then the points in the polygon formed by $p_1, p_2, \dots, p_k$ are at a distance less than $\tilde{f}_0$ from all points of R_j.*

We introduce a numerically reliable area reduction method based on the above assertions. Due to the length and complexity of the algorithm, we will skip the technical details; these will be presented in a forthcoming paper [14]. (Keep in mind that we intend to present a *computer-aided proof*; that is, it is necessary to make the algorithmic details available and to perform a proof of correctness of the algorithms.)

Assuming *exact computations*, one can easily see that if the polygons $R_i, i = 1, \dots, n$, are initialized as convex sets (as it is in the current method; see Algorithm 2, step 1), then they remain convex after each elementary reduction made by Theorem 5.2. However, with finite precision arithmetic the points p_1 and p_k cannot be evaluated exactly. In the method of Nurmela and Östergård the evaluated points p_1 and p_k are corrected by estimating the possible computation error, while in the present method proper rectangles as the guaranteed enclosures of p_1 and p_k are computed. However, both methods may result in concave, or even self-intersecting, R_i polygons. To avoid the difficulty of representing and handling extremely irregular sets, we make some restrictions for the shape of the polygons; namely, we assume that the remaining regions satisfy the invariance criterion below (implicitly defining what we call "polygon" in what follows).

INVARIANCE CRITERION. *During the whole running of the interval implementation of Algorithm 2, each R_i, $i = 1, \dots, n$ (called remaining region or remaining polygon), is either a point p_1, a line segment $\overline{p_1 p_2}$, or a "simple" polygon $(p_1, \dots, p_k)$, $k \ge 3$, i.e., a polygon with edges $\overline{p_1 p_2}, \dots, \overline{p_{k-1} p_k}, \overline{p_k p_1}$, such that each pair of edges has at most one joint point as the joint endpoint of two consecutive edges.*

Notice that the above criterion allows also nonconvex polygons to form a remaining region. Algorithm 3 shows our designed interval algorithm for implementing an elementary area reduction method. In the algorithmic description we use the notation $d(x,y)$ for the euclidean distance between two points $x, y \in \mathbb{R}^2$.

In Algorithm 3 we consider several cases depending on the number of nodes of the polygon to be reduced: $s = 1$ is handled in steps 4 and 5, while $s = 2$ and

Algorithm 3. *Diminish_ij*—the reliable version.

```
Inputs:  – $R_i = R_i(b_1, b_2, \dots, b_s)$: the polygon to be reduced,
         – $R_j = R_j(a_1, a_2, \dots, a_t)$: the polygon used for reducing $R_i$,
         – $\tilde{f}_0$: a validated lower bound of the global maximum of (2.1).
Output:  – $R'_i$: the remaining polygon of $R_i$.
1:  for $l := 1$ to $s$ do
2:      if (it is guaranteed that $d(b_l, a_m) < \tilde{f}_0$ for all $m = 1, \dots, t$) then mark $b_l$ with a "$-$" flag;
3:      else mark $b_l$ with a "$+$" flag;
4:  if (all the $b_l$ have "$-$") then return "$R'_i$ is empty";
5:  else if (all the $b_l$ have "$+$") then return $R'_i := R_i$;
6:  Find a sequence of consecutive vertices $b_l$ with "$-$", denoted by $p_2, \dots, p_{k-1}$.
7:  if ($s = 2$) then {comment: $R_i$ is a line segment}
8:      Denote the node of $R_i$ differing from $p_2$ by $p_0$;
9:      Find an enclosure $P_1 \in \mathbb{I}^2$ of a point $p_1$ such that $p_1$ is on the line segment $\overline{p_0 p_2}$, and $d(p_1, a_m) < \tilde{f}_0$ for all $m = 1, \dots, t$.
10:     Build $R'_i$ from $P_1, p_2$;
11: else {comment: $s \geq 3$}
12:     Denote the preceding node of $p_2$ in $R_i$ by $p_0$;
13:     Denote the succeeding node of $p_{k-1}$ in $R_i$ by $p_{k+1}$;
14:     Find an enclosure $P_1 \in \mathbb{I}^2$ of a point $p_1$ such that $p_1$ is on the line segment $\overline{p_0 p_2}$, and $d(p_1, a_m) < \tilde{f}_0$ for all $m = 1, \dots, t$.
15:     Find an enclosure $P_k \in \mathbb{I}^2$ of a point $p_k$ such that $p_k$ is on the line segment $\overline{p_{k-1} p_{k+1}}$, and $d(p_k, a_m) < \tilde{f}_0$ for all $m = 1, \dots, t$.
16:     Let $d_1 = p_{k+1}, \dots, d_{s-k+2} = p_0$ be the consecutive vertices of $R_i$ not chosen in step 6;
17:     Build $R'_i$ from $P_1, P_k, d_1, \dots, d_{s-k+2}$;
18: return $R'_i$;
```

$s \geq 3$ are considered in steps 7 to 10 and steps 11 to 17, respectively. We represent polygons commonly as a sequence of consecutive vertices, but we assume that the coordinates of the vertices are machine numbers. (We start the elimination procedure with such polygons; see Algorithm 2, step 1.) Each execution of the interval version of Algorithm 3 results in either an empty polygon (step 4) if we can *provide a guarantee* that each vertex of R_i is at a distance less than $\tilde{f}_0$ from R_j; or a polygon (step 5 or 18) which *contains the polygon that would be obtained assuming exact arithmetic.*

Note that $p_0 = p_{k+1}$ may hold in steps 12 and 13; in this case we construct R'_i without duplicating this point in the result polygon.

There are several crucial parts of Algorithm 3 making the procedure reliable. We give only a short overview on them; for details see [14]. The first problem to be resolved is to mark the vertices of R_i correctly, i.e., to verify the validity of the **if** condition of step 2. The second is to compute inclusion rectangles for some appropriate p_1 and p_k points (steps 9, 14, 15) we would get if we used exact arithmetic. Finally, we have to build resulting polygons R'_i in steps 10 and 17 satisfying the invariance criterion and enclosing all points we must keep.

The invariance criterion of the resulting polygon is reached basically during the latter task by testing several separation properties. Consider Figure 5.1, which shows an example of executing Algorithm 3 for $s = 2$ and $s \geq 3$, respectively. (Now we

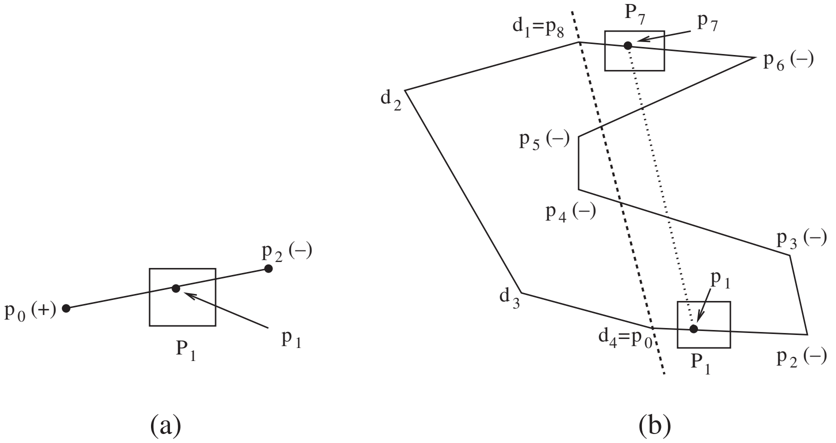


Fig. 5.1. *The polygon R_i and the enclosures P_1, P_2 in the method of active areas with interval computations. Cases $s = 2$ (left) and $s \geq 3$ with $s = 9,\ k = 7$ (right).*

sketch the key idea only when $s \geq 3$, since $s = 2$ is merely a special case of this.) After creating P_1 and P_2 we can test whether the convex hull $conv(P_1, P_k)$ of them is strictly separated from the set of vertices $d_2, \ldots, d_{s-k+1}$ by the line given by p_0 and p_{k+1} (we do not allow even touching the line). If so, then the polygons formed by $d_1, \ldots, d_{s-k+2}$ and by $conv(p_0, p_{k+1}, P_1, P_k)$ both satisfy the invariance criterion. Moreover, their union (which can be obtained by concatenating the corresponding vertices) also satisfies the criterion and forms a remaining polygon.

After making a more elaborative analysis on Algorithm 3, we can prove its correctness as a reliable method of eliminating inactive regions.

Theorem 5.3 (see [14]). *The interval implementation of Algorithm* 3 *eliminates only those points from R_i which are guaranteed to be at a distance less than $\tilde{f}_0$ from all points of R_j.*

Corollary 5.4 (see [14]). *Algorithm* 2 *deletes only those $(x, y) \in \mathbb{R}^{2n}$ feasible points for which $f_n(x, y) < \tilde{f}$ holds.*

6. A global elimination procedure. Now we have a reliable B&B method for solving circle packing problem instances as global optimization problems. However, some difficulties would arise if one ran Algorithm 1 on the whole initial search box $[0, 1]^{2n}$. The result would consist of at least $n!$ solutions showing the same packing pattern, and differing from each other only in the indexing of the result components. In other words, in the optimization frame procedure the points will be permuted, but these permutations give the same solution in the geometrical sense. The other possible problem is that each solution can be transformed into symmetric, but numerically different, solutions (by reflections and rotations), which also increase the number of obtained result elements unnecessarily. The predecessor [12] of the currently used interval method (which approximated the active areas by rectangular cells instead of polygons) was able to solve packing problems only up to $n = 5$ within several hours of CPU time when the permutation and symmetry problems were not resolved.

One possible solution for the above difficulties can be reached, for example, by the lexicographical ordering of the components. This was realized in an imposed form of the earlier introduced noninterval-based computer-aided proofs (see [5, 18, 19]). The method is called *tiling.* In what follows we give an overview on the tiling method, and after that we make a proposal to overcome its drawback for large numbers of circles and extend the optimality proof for previously unsolved problem instances.

6.1. Tiling methods. Assuming that a lower bound $\tilde{f}$ for the maximum value of the considered point packing problem instance is given, split the unit square into regions (tiles) in such a way that the squared distance between any two points in each tile is less than $\tilde{f}$ (or the distance between any two points in each tile is less than $\tilde{f}_0$). Then for each packing configuration attaining objective function value greater than or equal to $\tilde{f}$, each tile can contain obviously at most one point of the packing. The optimality of a given packing can be proved by running the search procedure on all possible tile combinations.

Remark 1. For many optimization problems only an approximation or a bound of the optimum value is required by the user. We can validate such pregiven approximations when we use tiling: if all the tile combinations can fully be eliminated considering a hypothetical bound as a cutoff value, then it can be regarded as an upper bound of the global optimum.

An interesting question arising is how to split the unit square to produce combinations which are easy to represent and handle and, in addition, to keep the number of combinations as small as possible. Our interval-based method prefers rectangular tiles applied also by the earlier strategies. When splitting the unit square into $k \times l$ rectangles (in a regular way), the minimal number of initial tile combinations is determined by

$$\min\left\{\binom{k\cdot l}{n} \mid k, l \geq 1 \text{ integers}, \ (1/k^2 + 1/l^2)^{1/2} < \tilde{f}_0\right\}.$$

In the previous studies the initial tile combinations were eliminated sequentially one after the other. However, the highly increasing number of those combinations made problem instances $n \geq 28$ unsolvable with this strategy in an acceptable time limit. For example, for $n = 27$ a 6×6 tiling can be applied, resulting in $\binom{36}{27} \approx 9.4 \cdot 10^7$ initial time combinations. The paper [19] does not present computational details, but the optimality proof for this case was completed in about one month of CPU time (performing simple floating-point computations instead of the about 4–35 times slower interval arithmetic). For $n = 28$ at least a 7×6 tiling is needed, thus, $\binom{42}{28} \approx 5.3 \cdot 10^{10}$ combinations must be checked. As in earlier studies, symmetry properties would help to reduce this number, but afterwards we would still have more than $1.3 \cdot 10^{10}$ cases.

The main idea of the proposed procedure is the following: the largest part of the combinations may be eliminated in a relatively easy way considering only local relations within the combinations. In detail, if one can discover patterns of tiles which cannot contain components of an optimal solution, then several tile combinations (i.e., higher-dimensional subproblems) containing any of these patterns can be discarded quickly. Let us denote by $P(n, X_1, \dots, X_n, Y_1, \dots, Y_n)$ a point packing problem instance where n is the number of points to be packed, $X_i, Y_i \in \mathbb{I}$, $i = 1, \dots n$, are the components of the starting box, and the objective function is given by (2.2). The simple theorem below shows how to apply a result achieved on a $2m$-dimensional packing problem for a higher-dimensional problem with $2n$ dimensions, $n \geq m \geq 2$.

THEOREM 6.1. *Let $n \geq m \geq 2$ be integers and let*

$$P_m = P(m, Z_1, \dots, Z_m, W_1, \dots, W_m) = P(m, (Z, W)),$$
$$P_n = P(n, X_1, \dots, X_n, Y_1, \dots, Y_n) = P(n, (X, Y))$$

be point packing problem instances ($X_i, Y_i, Z_i, W_i \in \mathbb{I}; X_i, Y_i, Z_i, W_i \subseteq [0, 1]$). Run Algorithm 1 on P_m using a hypothetic $\tilde{f}$ cutoff value in the accelerating devices and *skipping step* 7, *and stop after an arbitrary, preset number of iteration steps. Denote by* $(Z'_1, \dots, Z'_m, W'_1, \dots, W'_m) := (Z', W')$ *the componentwise hull of all the elements placed on the WorkTree and on the ResultList. Assume that there exists an invertible, distance-preserving geometrical transformation* φ *with* $\varphi(Z_i) = X_i$ *and* $\varphi(W_i) = Y_i$ *for all* $i = 1, \dots, m$*. Then for each point packing* $(x, y) \in \mathbb{R}^{2n}$ *satisfying* $(x, y) \in (X, Y)$ *and* $f_n(x, y) \geq \tilde{f}$*, the statement*

$$(x, y) \in (\varphi(Z'_1), \dots, \varphi(Z'_m), X_{m+1}, \dots, X_n, \\ \varphi(W'_1), \dots, \varphi(W'_m), Y_{m+1}, \dots, Y_n) := (X', Y')$$

also holds.

Proof. Perform an indirect proof and assume that a feasible solution $(x, y) \in \mathbb{R}^{2n}$ of P_n with $f_n(x, y) \geq \tilde{f}$ is discarded when modifying the starting box of P_n, i.e., $(x, y) \in (X, Y)$, but $(x, y) \notin (X', Y')$. Additionally, let $(z, w) = (\varphi^{-1}(x_1), \dots, \varphi^{-1}(x_m), \varphi^{-1}(y_1), \dots \varphi^{-1}(y_m)) \in \mathbb{R}^{2m}$. Notice that $(x, y) \notin (X', Y')$ implies $(z, w) \notin (Z', W')$ by the indirect assumption, and, moreover,

$$\tilde{f} \leq f_n(x, y) \leq f_m(x_1, \dots, x_m, y_1, \dots y_m) = f_m(z, w). \tag{6.1}$$

The second inequality of (6.1) follows from (2.2), and (6.1) is implied by the distance-preserving property of φ^{-1}. Consequently, Algorithm 1 deleted the feasible solution (z, w) of P_m for which $\tilde{f} \leq f_m(z, w)$ holds. But this is a contradiction, since all the accelerating devices of Algorithm 1 erase only those feasible solutions for which the objective function value is less than $\tilde{f}$. This completes the proof. □

The meaning of Theorem 6.1 is the following: assume that we are able to reduce some search regions on a tile set S'. When processing a higher-dimensional subproblem on a tile set S containing the image of S', it is enough to consider *the image of those of the remaining regions of* S' as the particular components of S. Mostly we apply the theorem in the special case when φ is the identity, i.e., when the tile set of the higher-dimensional problem is the superset of the tile set of the smaller problem.

As a consequence of the theorem, if it is proved that S' cannot contain point packings attaining at least $\tilde{f}$ function value, then all the higher-dimensional problems with tile set S, $S' \subseteq S$ can be eliminated at once (when using the same $\tilde{f}$). The latter fact is formalized below with the notation of Theorem 6.1.

COROLLARY 6.2. *Let* φ *be the identity transformation and assume that Algorithm* 1 *stops with an empty WorkTree and with an empty ResultList; i.e., the whole search region* $(Z, W) = (Z_1, \dots, Z_m, W_1, \dots, W_m) = (X_1, \dots, X_m, Y_1, \dots, Y_m)$ *is eliminated by the accelerating devices using (the same)* $\tilde{f}$*. Then* (X, Y) *does not contain any* $(x, y) \in \mathbb{R}^{2n}$ *vectors for which* $f_n(x, y) \geq \tilde{f}$ *holds.*

Remark 2. In order to apply Theorem 6.1 correctly, we have to take care of the following important facts:

(i) $w(Z_i) = w(X_i)$ and $w(W_i) = w(Y_i)$ for all $i = 1, \dots, m$ should be provided. This assumption does not necessarily hold if one tries to split the unit square in a regular way using floating-point numbers. The solution is the enlargement of the unit square in such a way that the resulting bounds of the tiles are exactly representable machine numbers (e.g., small integers).

(ii) We use shifting and rotating operators as φ, but even in these simple cases we have to apply the transformations in an interval way to obtain a guaranteed enclosure of the transformed objects.

6.2. Basic algorithms for the optimality proofs. Our optimality proofs are based on Theorem 6.1 and have the following basic idea: first we find feasible patterns of tiles and remaining areas on some small subsets of the whole set of tiles and then we process bigger and bigger subsets while using the results of the previous steps. Thus, our method is not fully automated, it consists of several phases, and each phase depends on the result of the previous phases. (However, the whole method is easily controlled by short command scripts associated to each phase.)

Obviously, there are several ways of "growing" the considered search regions until we reach the enclosures of the global maximizers as our final destination. We introduce two basic algorithms executed in our computer-aided proofs. We discuss them in general, i.e., for an arbitrary n and for a regular $k \times l$ splitting (with k rows and l columns) of the square. We assume that the columns are numbered $1, \dots, l$ from left to right.

To discuss the algorithms in detail let us introduce some new notation and abbreviations. In the beginning of each phase we determine some sets of tile combinations consisting either of the remaining areas of the previous phases (or its transformations) or of full tiles. These sets are denoted by $S^m_{s..f}$, where $s \le f$, $s, f \in \{1, \dots, l\}$, $m \in \{0, 1, \dots, n\}$, symbolizing that a number of m remaining or full tile regions are considered and the regions are chosen from columns $s, s+1, \dots, f$. (In the example of $n = 29$, we first generate $S^{15}_{1..3}$, that is, all the combinations of 15 tiles where the tiles are chosen from the leftmost 3 columns of the square.) Moreover, for a set $M = \{m_1, \dots, m_k\} \subseteq \{0, 1, \dots, n\}$, let $S^M_{s..f} = \cup_{i=1}^k S^{m_i}_{s..f}$. The elements of the sets $S^m_{s..f}$ are processed sequentially by the optimization algorithm using $\tilde{f}$. With the exception of the final phase, we stop the optimization algorithm after a certain number of iterations. As a result, some combinations can be fully or partly eliminated. The resulting new sets are denoted by $\bar{S}^m_{s..f}$. The sets $\bar{S}^m_{s..f}$ form the input components of the subsequent phases. We will use the notation ${}^{m_1}_{s_1..f_1}S^m_{s..f}$, $1 \le s \le s_1 \le f_1 \le f \le l$, $0 \le m_1 \le m \le n$, as that of the subset of $S^m_{s..f}$ in which each element contains exactly m_1 tile regions from columns $s_1, \dots, f_1$. Note that the elements of the sets $S^m_{s..f}$, $\bar{S}^m_{s..f}$, ${}^{m_1}_{s_1..f_1}S^m_{s..f}$, and ${}^{m_1}_{s_1..f_1}\bar{S}^m_{s..f}$ are in general only *enclosures* of the corresponding exact remaining areas.

Remark 3. In each phase of our optimality proof we accelerate Algorithm 1 by calling the method of active areas immediately to the initial search region. In many cases, this results in the elimination of the whole search box before doing any bisection. (Obviously, this modification has no effect on the correctness of Theorem 6.1.)

Algorithm 4 (called *Grow*) takes the set $\bar{S}^m_{1..c}$ for a given pair (c, m), i.e., all the current remaining areas where m tiles are chosen from the first c columns and produces a new set ${}^{m}_{1..c}S^{m+i}_{1..(c+1)}$. The procedure takes all the combinations $A \in \bar{S}^m_{1..c}$ (step 3) and adds a suitable (see Remark 4) number i of new tiles to them in all the possible ways. With the aid of the sets $\bar{S}^{n-m}_{c+1..l}$, in step 6 we test whether pattern B can be valid in column $(c+1)$. If this is the case, each result combination A' (step 7) can be tested further since we have additional information on the possible remaining areas in columns $2, \dots, (c+1)$. Namely, $\bar{S}^{i+j}_{2..(c+1)}$ can be generated from the given $\bar{S}^{i+j}_{1..c}$ by applying Theorem 6.1 (performing a shifting transformation). In case there is no remaining combination with pattern P in $\bar{S}^{i+j}_{2..(c+1)}$ (tested in step 9), we can immediately ignore A' by Corollary 6.2. Otherwise we intersect A' with $C \in \bar{S}^{i+j}_{2..(c+1)}$ on the tile positions corresponding to P (step 10) and apply Theorem 6.1 again to the result A'': we store A'' in the output set only if none of its components are empty (steps 11 and 12).

Algorithm 4. *Grow*: add one column.

```
Inputs:  – $n$: the number of points to be packed,
         – $k$, $l$: the square is divided into $k$ rows and $l$ columns regularly,
         – $c$, $1 \leq c \leq l$: a column index,
         – $\bar{S}^{t}_{1..c}$ for all $0 \leq t \leq ck$,
         – $\bar{S}^{t}_{c+1..l}$ for all $0 \leq t \leq (l-c)k$ (if available),
         – $m$, $1 \leq m \leq ck$: the elements of $\bar{S}^{m}_{1..c}$ will be extended by adding
           one column,
         – $i$: the number of tiles to be added from column $c+1$.
Output:  – ${}^{m}_{1..c}S^{m+i}_{1..(c+1)}$: the output set.

Initialization: ${}^{m}_{1..c}S^{m+i}_{1..(c+1)} := \emptyset$;
Generate $S^{i}_{(c+1)..(c+1)}$ from full tiles;
for all ($A \in \bar{S}^{m}_{1..c}$) do
  Let $j$ be the number of regions in columns $2, \dots, c$ of $A$;
  for all ($B \in S^{i}_{(c+1)..(c+1)}$) do
    if ($\bar{S}^{n-m}_{c+1..l}$ is available and there exists an element of it having the same tile
    pattern as B in column $c+1$) then
      Let $A'$ be the concatenation of $A$ and $B$;
      Let $P$ be the set of tile positions in columns $2, \dots, (c+1)$ of $A'$;
      if (a $C \in \bar{S}^{i+j}_{2..(c+1)}$ exists with tile positions $P$) then
        Intersect $A'$ and $C$ on positions $P$ resulting $A''$;
        if (none of the components of $A''$ is empty) then
          Add $A''$ to ${}^{m}_{1..c}S^{m+i}_{1..(c+1)}$.
```

Figure 6.1 demonstrates how the algorithm works for $k = l = 4,\ c = 2,\ m = 5,\ i = 3$ and for $j = 3$ in the main loop. In each part of the figure solid lines denote the boundaries of the considered combinations.

Remark 4. In practice, Algorithm 4 is performed on some relevant m values. In order to cover all the valid tile combinations on the first $c+1$ columns, we have to compute a lower m_l and an upper m_u bound on the number of tiles $m+i$ of the output, i.e., to associate a set of i values to each m. A possible choice for a lower bound is m; however, we may increase this number if we have a kind of complementary information: assume that it is guaranteed that all the point packings $x \in \mathbb{R}^{2n}$, $f_n(x) \geq \tilde{f}$ can contain at most t points in columns $(c+2), \dots, l$. In other words, we know that $S^{t_0}_{c+2..l} = \emptyset$ (or $\bar{S}^{t_0}_{c+2..l} = \emptyset$) for each $t_0 > t$. Then m_l can be set to $\max(m, n-t)$. An upper bound on $m+i$ can be determined by $m_u := \min(m+k, n)$. Summarizing these results, Algorithm 4 must be performed to all i with $\max(0, n-t-m) \leq i \leq \min(k, n-m)$ for each m.

Algorithm 5, called *Join*, joins the elements of two sets of tile combinations pairwise. This algorithm also inputs a given parameter pair (c, m). It is assumed that the possible remaining regions are known when locating m points in the first c columns. Moreover, we have all the possibilities to add some points in the $(c+1)$th column for all the elements of $\bar{S}^{m}_{1..c}$; thus, we know the sets ${}^{m}_{1..c}\bar{S}^{m_1}_{1..(c+1)}$, e.g., by Algorithm 4. Similarly, the sets ${}^{n-m}_{(c+1)..l}\bar{S}^{m_2}_{c..l}$ are also assumed to be known. (In practice, the latter sets are evaluated from ${}^{n-m}_{1..(l-c)}\bar{S}^{m_2}_{1..(l-c+1)}$ applying a rotation by 180 degrees around the midpoint of the search square for each element, since we "grow" tile combina-

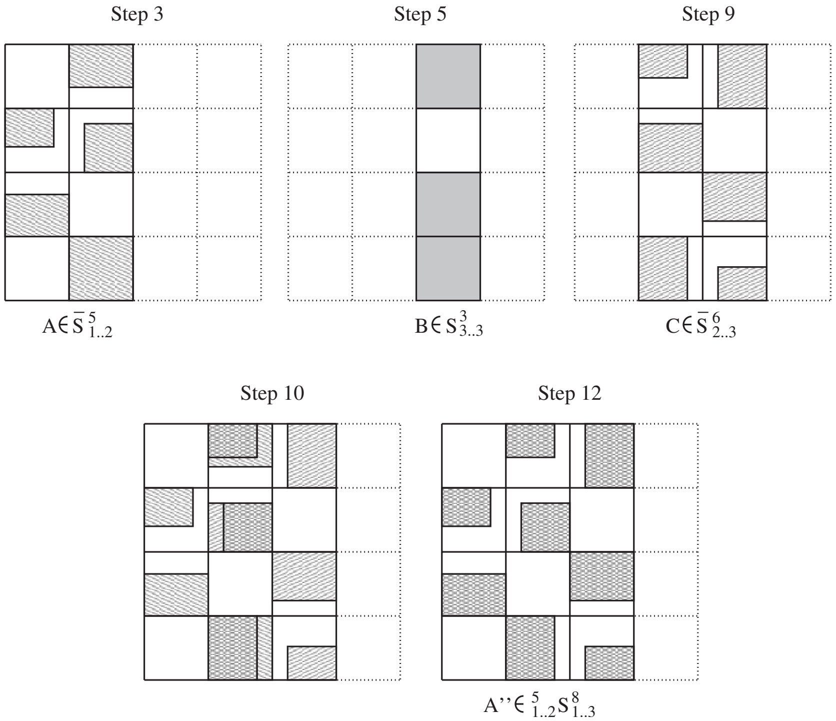


FIG. 6.1. *An example of adding a new column for* $k = l = 4,\ c = 2,\ m = 5,\ i = 3,\ j = 3$.

Algorithm 5. *Join*: join remaining areas.

```
Inputs:  – $n$: the number of points to be packed,
         – $k$, $l$: the square is divided into $k$ rows and $l$ columns regularly,
         – $c$, $1 \le c \le l-1$: a column index,
         – $m$, $0 \le m \le ck$,
         – ${}^{m}_{1..c}\bar{S}^{m_1}_{1..(c+1)}$ for all $m_1$, $m \le m_1 \le n$,
         – ${}^{n-m}_{(c+1)..l}\bar{S}^{m_2}_{c..l}$ for all $m_2$, $n-m \le m_2 \le n$.
Output:  – ${}^{m}_{1..c}S^{n}_{1..l}$: the output set.
Initialization: ${}^{m}_{1..c}S^{n}_{1..l} := \emptyset$;
for $m_1 := m$ to $n$ do
  for all ($A \in {}^{m}_{1..c}\bar{S}^{m_1}_{1..(c+1)}$) do
    Let $j$ be the number of regions in column $c$ of $A$;
    Let $P$ be the set of tile positions in columns $c$ and $(c+1)$ of $A$;
    for all ($C \in {}^{n-m}_{(c+1)..l}\bar{S}^{n-m+j}_{c..l}$) do
      Let $P'$ be the set of tile positions in columns $c$ and $(c+1)$ of $C$;
      if ($P = P'$) then
        Intersect $A$ and $C$ on positions $P$ resulting $A'$;
        if (none of the components of $A'$ are empty) then
          Add $A'$ to ${}^{m}_{1..c}S^{n}_{1..l}$.
```

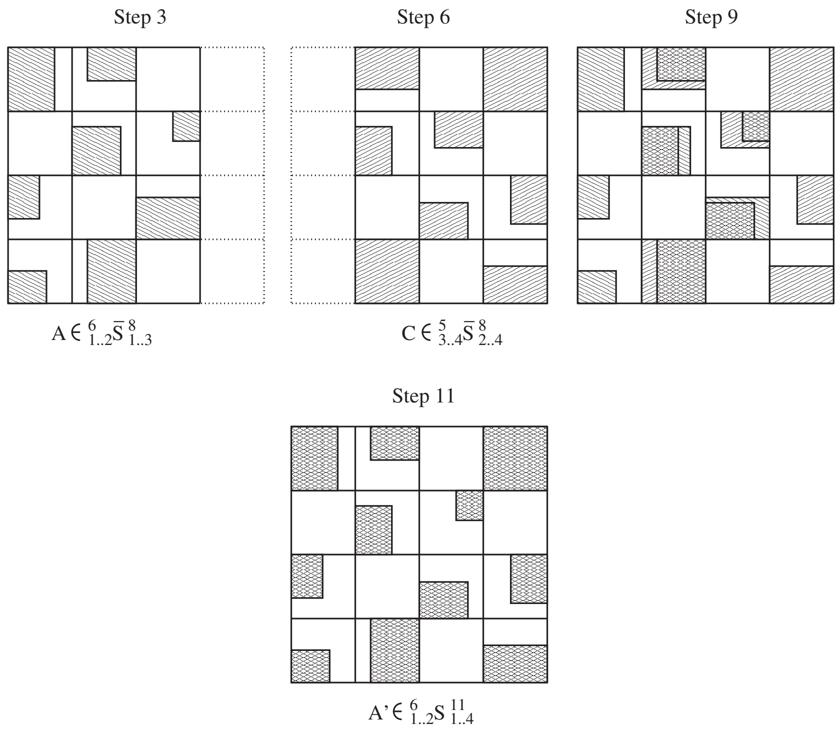

Step 3
Step 6
Step 9
$A \in {}^{6}_{1..2}\bar{S}^{8}_{1..3}$
$C \in {}^{5}_{3..4}\bar{S}^{8}_{2..4}$
Step 11
$A' \in {}^{6}_{1..2}S^{11}_{1..4}$

FIG. 6.2. *An example of joining the remaining areas for $n = 11,\ k = l = 4,\ c = 2,\ m = 6$ and for $m_1 = 8,\ j = 3$.*

tions from left to right, starting from column 1.) Consequently, the input sets can be intersected in tile positions from columns c and $c + 1$.

Algorithm 5 works as follows: we take all the elements A from ${}^{m}_{1..c}\bar{S}^{m_1}_{1..(c+1)}$ (step 3). Then for each A we determine *all* those combinations $C \in {}^{n-m}_{(c+1)..l}\bar{S}^{n-m+j}_{c..l}$ which have the same tile positions in columns c and $c+1$ as A (steps 5–8). If the condition of step 8 holds, then we join A and C by Theorem 6.1 (step 9), and finally, similarly to Algorithm 4, we store the result only if all of its components are nonempty (steps 10 and 11). The output set contains all those remaining areas of the considered packing problem for which exactly m tiles are chosen from the leftmost c columns. Figure 6.2 shows an example of executing Algorithm 5 for $n = 11,\ k = l = 4,\ c = 2,\ m = 6$ and for $m_1 = 8,\ j = 3$.

Notice that Algorithm 5 can be generalized on the base of the number i of intersecting columns. Similar representations with $i = 0, \dots, l - 1$ are possible. In the present paper only $i = 2$ is considered since it proved to be enough for the discussed problem instances. However, in general it may be worth it to apply many different i values.

7. A method for handling free circles.

Let us introduce the following definition.

DEFINITION 7.1. *Consider the point packing problem class with the distance function and objective function given by* (2.2). *Consider an optimal point packing, i.e., a vector $(x, y) = (x_1, \dots, x_n, y_1, \dots, y_n) \in \mathbb{R}^{2n}$ for which f_n is maximal. We call a point $p_k = (x_k, y_k),\ k \in \{1, \dots, n\}$, of this optimal packing to be a* free point*, or,* *equivalently,* a free circle center*, if there exists a half line H with the endpoint p_k and there exists a positive number ε, such that*

$$f_n(x, y) = f_n(x_1, \dots, x'_k, \dots, x_n, y_1, \dots, y'_k, \dots, y_n)$$

for all $(x'_k, y'_k) \in H \cap N_\varepsilon(p_k)$, where $N_\varepsilon(p_k)$ denotes the ε-neighborhood of p_k.

Note that by the equivalence of the point packing and circle packing problems each free point in an optimal point packing is identified by the center of a *free circle* of the corresponding optimal packing of circles and vice versa. The meaning of the above definition is straightforward: we say a circle is free in the optimal packing if the center of this circle can be slightly perturbed without losing the optimality of the packing. It is also obvious that the handling of free circles is crucial when solving circle packing problems, since they pose a continuum (and usually a positive measure) set of equivalent global optimizers.

In advanced phases of the B&B algorithm, the enclosure of such a set usually appears as a region extremely bigger than the others (and it is subdivided probably unnecessarily), but it is almost useless as a reducing region. It is apparent that one can consider only a single canonical value instead of such a point set. The following method shows how to apply a simple criterion to temporarily replace some parts of the search region with single points (without losing any optimal solutions).

1. Let $(X_1, \dots, X_n, Y_1, \dots, Y_n) = (X, Y) \in \mathbb{I}^{2n}$ enclose all the remaining boxes (i.e., all the candidates stored either in the WorkTree or in the ResultList) after a certain number of iteration loops when executing Algorithm 1 (or its accelerated version; see Remark 3). We assume that either the WorkTree or the ResultList is not empty. Moreover, let $\tilde{f}$ be the current cutoff value used in the algorithm.

2. Assume that there exist machine representable points $p_{k_1}, \dots, p_{k_t},\ p_{k_s} \in (X_{k_s}, Y_{k_s}),\ s \in \{1, \dots, t\}$, in t different components of (X, Y), such that for each k_s

$$\underline{D}(p_{k_s}, (X_j, Y_j)) > \overline{F}(X, Y) \geq \tilde{f} \tag{7.1}$$

holds for all $j \in \{1, \dots, n\},\ j \neq k_s$. Let K denote the set of indices $\{k_1, \dots, k_t\}$.

3. Replace the components (X_i, Y_i) of (X, Y) with the point intervals p_i for each $i \in K$. Run Algorithm 1 on the resulting (X', Y') box, ignoring the improvement step of $\tilde{f}$, i.e., using the earlier found best lower bound, and stop it after a certain number of iteration steps.

4. Let $(X'', Y'') \in \mathbb{I}^{2n}$ enclose all the remaining boxes. The components of the *output box* of the procedure are then given by (X_i, Y_i) for $i \in K$ and (X''_j, Y''_j) for each $j \notin K$; i.e., the latter components can be diminished as compared to (X_j, Y_j).

The idea behind the proposed method is the following: Assume that optimal point packings exist in (X, Y). Then clearly $\overline{F}(X, Y) \geq f^*$ holds. Consider a set of points $\{p_j \,|\, p_j \in (X_j, Y_j), j \notin K\}$ with a minimal pairwise distance f^*. Notice that by (7.1) this set can be expanded with $\{p_i, i \in K\}$ to obtain an optimal packing. Moreover, for each $i \in K$, $d(p_i, p_j) > \overline{F}(X, Y) \geq f^*$ occurs for all $j \in \{1, \dots, n\},\ j \neq i$, which implies the existence of an $\varepsilon > 0$ such that $d(p'_i, p_j) > f^*$ for all $p'_i \in N_\varepsilon(p_i)$. In other words, each p_i is a free point of each such packing.

THEOREM 7.2. *The above procedure for shrinking the remaining regions is correct in the sense that all the optimal packings in (X, Y) are also present in the output box of the procedure.*

Proof. Let us use the abbreviations $Z := (X, Y)$ and $Z' := (X', Y')$. Consider a box $V' \subseteq Z'$ eliminated by Algorithm 1 using $\tilde{f}$ in step 3 of the above method. Recall that $V'_i = p_i$ for all $i \in K$. Let $V \subseteq Z$ be given by $V_j := V'_j$ for $j \notin K$ and by $V_i := Z_i$

for the other components. It is enough to show that V would also be eliminated by Algorithm 1 using $\tilde{f}$ (i.e., that V cannot contain global optimizers).

First, assume that V' is eliminated by the method of active areas. Notice that from (7.1) each area reduction is made only between components $j_1, j_2 \notin K$. Clearly, if a component V'_s can be fully erased by such regions, then it could also be eliminated in V by the same test, since in V all the particular reducing regions of V' are also present.

Second, consider the case when V' is deleted by the cutoff test. We first prove two simple consequences of (7.1):

$$\underline{D}(p_i, Z_j) > \tilde{f} \ \Rightarrow\ \overline{D}(p_i, V'_j) = \overline{D}(V'_i, V'_j) > \tilde{f}, \tag{7.2}$$

$$\underline{D}(p_i, Z_j) > \tilde{f} \ \Rightarrow\ \overline{D}(Z_i, V_j) = \overline{D}(V_i, V_j) > \tilde{f} \tag{7.3}$$

for all $i \in K,\ j \in \{1, \dots, n\},\ i \neq j,\ V'_j \subseteq Z_j$, and $V_j \subseteq Z_j$.

Both statements can easily be proved in an indirect way by applying the fact that D is an inclusion function of the distance function d. Assuming the converse of the conclusions of (7.2) and (7.3), respectively, we could state $d(p_i, (x_j, y_j)) \leq \tilde{f}$, where (x_j, y_j) is arbitrarily chosen from V'_j for (7.2) and from V_j for (7.3). But due to $(x_j, y_j) \in Z_j$, this contradicts the premise of both (7.2) and (7.3).

Since V' is deleted by the cutoff test, we obtain

$$\begin{aligned}\tilde{f} > \overline{F}(V') &= \min_{1\leq i\neq j\leq n} \overline{D}(V'_i, V'_j) = \min_{\substack{1\leq i\neq j\leq n\\ i\notin K, j\notin K}} \overline{D}(V'_i, V'_j) = \min_{\substack{1\leq i\neq j\leq n\\ i\notin K, j\notin K}} \overline{D}(V_i, V_j)\\ &= \min_{1\leq i\neq j\leq n} \overline{D}(V_i, V_j) = \overline{F}(V).\end{aligned}$$

Here the conclusions of (7.2) and (7.3) are applied in the second and fourth equations, respectively, while the third equation follows from the construction of V. Thus, from the first and last statements of the chain we obtained that V can also be deleted using the same cutoff value. This completes the proof. □

According to our computational experience, the use of single values instead of regions was indispensable in achieving high precision verified solutions for the circle packing problems. The above method is performed in practice in the following way: for each component $(X_i, Y_i),\ i \in \{1, \dots, n\}$, a stochastic search is applied for finding a machine representable point $p_i \in (X_i, Y_i)$ for which $g(i) = \min_{1\leq j\leq n, j\neq i} \underline{D}(p_i, (X_j, Y_j))$ is maximal. The search procedure we used was the stochastic optimization method GLOBAL [2]. If $g(i) > \overline{F}(X, Y) \geq \tilde{f}$ holds for the best $g(i)$ value found, then it is ensured that (7.1) also holds. (Notice, that the obtained p_i is reliable since $g(i)$ is computed by interval arithmetic.)

8. Optimality for $n = 28, 29$, and 30.

8.1. Hardware and software environment. The optimization procedure was carried out on a PC with an Intel Pentium IV 1400 MHz processor and 1 Gbyte RAM running under the Linux operating system. The global optimization frame algorithm is based on the Toolbox for C–XSC libraries [7], while the low-level interval arithmetic routines were implemented by PROFIL/BIAS [9]. Additionally, the multiple precision extension of PROFIL [10] is also built in to determine $\tilde{f}$ and to have correctly rounded decimal numbers in those I/O routines which communicate with the user. (During the intermediate phases of our algorithm we used binary I/O routines.)

8.2. The global procedure. The currently known best packings for the investigated cases were published in [4] (for $n = 28$ points) and in [17] (for $n = 29$ and 30, respectively). The solution points and the corresponding function values can be found in [22], where the result components are rounded to 13 decimal digits. In order to obtain higher precision for the initial values of $\tilde{f}$, the *structure*s of the best packings were investigated. These structures describe which points are located on the sides of the square and which pairs of points have minimal distances. The structures are formally given by systems of equations. Since the exact solutions of these systems are not known for $n = 28, 29$, we solved them numerically by Maple to a precision of 20 digits. The guaranteed enclosures of the objective function values on these more precise approximate results were then determined by a simple function evaluation with multiple precision intermediate interval data.

In contrast, the coordinates of the best known packing of 30 points have the exact form of $(qd + \text{par}(p+1)(1-4d), pa)$, where $d = (20 - \sqrt{10})/75$, $a = 1/5$, par denotes the parity function, and $p \in \{0, \dots, 5\}$, $q \in \{0, \dots, 4\}$. Moreover, the maximum value is exactly d. An enclosure of d was determined by a simple multiple-precision expression evaluation.

From the above function enclosures we obtained the following $\tilde{f}_0$ lower bounds for the objective function (2.1):

$$\tilde{f}_{0,28} = 0.2305354936426673,$$

$$\tilde{f}_{0,29} = 0.2268829007442089,$$

$$\tilde{f}_{0,30} = 0.2245029645310881.$$

These allowed us to split the search space into 7×6 uniform cells as initial tiles for all three problems.

The original point packing problem was modified in two steps: first, we considered squared distances instead of distances between points (see section 1) and then we enlarged the search space from $[0,1] \times [0,1]$ to $[0,42] \times [0,42]$—as was suggested by the first part of Remark 2. At the same time we evaluated the $\tilde{f}$ values

$$\tilde{f}_{28} = 93.7506267944766227,$$

$$\tilde{f}_{29} = 90.8034005467881010,$$

$$\tilde{f}_{30} = 88.9083890308478496$$

used during the whole computation.

Since the global procedure runs almost identically for the three problem instances, we introduce it only in the example of $n = 29$. (In [13] a similar report is given for $n = 28$.) In any case, *detailed computational results are given for all three problems* in Tables 8.1, 8.2, and 8.3, respectively. The results are grouped by the successive processing phases.

According to our previous suggestions, with the exception of the last phase we stopped the B&B algorithm after a preset number of iterations on each input combination. This number was 3 in the first three phases and 10,000 in the fourth (refining) phase.

Phase 1. First we evaluated $\bar{S}^m_{1..3}$ for some m. The initial sets $S^m_{1..3}$ were built from full tile combinations, since we had no previous information about the possible configurations. Notice that it is not worth it to evaluate the above sets when m is

TABLE 8.1

Computational details of solving the packing problem of $n = 28$ points. The two columns on the left show the name and the size of the input sets of the global optimization process: these sets are full tile combinations in Phase 1*, the resulting sets of Algorithm* 4 *in Phase* 2*, the resulting sets of Algorithm* 5 *in Phase* 3*, the aggregated resulting sets of Phase* 3 *in Phase* 4*, and the only box to be refined in Phase* 5*, respectively. The middle group of columns contains the CPU time (CPUt, in seconds), the number of function evaluations (NFE), and the number of executions of the method of active areas (NMAA) while processing the elements of the corresponding S sets. The column set on the right shows the name and the size of the output sets of the optimization method.*

S	$\|S\|$	CPUt	NFE	NMAA	$\bar{S}$	$\|\bar{S}\|$
Phase 1						
$S^{14}_{1..3}$	116 280	958	54 846	60 249	$\bar{S}^{14}_{1..3}$	17 799
$S^{15}_{1..3}$	54 264	139	4 401	16 658	$\bar{S}^{15}_{1..3}$	1 082
$S^{16}_{1..3}$	20 349	24	27	5 253	$\bar{S}^{16}_{1..3}$	0
$S^{13}_{1..3}$	203 490	2 526	232 268	174 995	$\bar{S}^{13}_{1..3}$	77 852
Phase 2						
${}^{13}_{1..3}S^{17}_{1..4}$	588 703	57 072	4 641 519	3 004 399	${}^{13}_{1..3}\bar{S}^{17}_{1..4}$	390 402
${}^{13}_{1..3}S^{18}_{1..4}$	632 289	47 620	2 843 863	2 214 934	${}^{13}_{1..3}\bar{S}^{18}_{1..4}$	226 564
${}^{13}_{1..3}S^{19}_{1..4}$	43 729	1 174	56 620	81 812	${}^{13}_{1..3}\bar{S}^{19}_{1..4}$	3 044
${}^{13}_{1..3}S^{20}_{1..4}$	0	-	-	-	${}^{13}_{1..3}\bar{S}^{20}_{1..4}$	0
${}^{14}_{1..3}S^{17}_{1..4}$	438 207	35 565	2 734 521	1 894 823	${}^{14}_{1..3}\bar{S}^{17}_{1..4}$	231 820
${}^{14}_{1..3}S^{18}_{1..4}$	354 167	22 466	1 312 522	1 090 923	${}^{14}_{1..3}\bar{S}^{18}_{1..4}$	103 582
${}^{14}_{1..3}S^{19}_{1..4}$	88 508	2 909	143 263	177 416	${}^{14}_{1..3}\bar{S}^{19}_{1..4}$	9 478
${}^{14}_{1..3}S^{20}_{1..4}$	2 540	13	12	2 550	${}^{14}_{1..3}\bar{S}^{20}_{1..4}$	0
${}^{14}_{1..3}S^{21}_{1..4}$	0	-	-	-	${}^{14}_{1..3}\bar{S}^{21}_{1..4}$	0
${}^{15}_{1..3}S^{17}_{1..4}$	21 844	1 276	95 333	74 306	${}^{15}_{1..3}\bar{S}^{17}_{1..4}$	7 816
${}^{15}_{1..3}S^{18}_{1..4}$	26 904	1 230	66 376	65 169	${}^{15}_{1..3}\bar{S}^{18}_{1..4}$	4 962
${}^{15}_{1..3}S^{19}_{1..4}$	14 134	312	13 370	22 543	${}^{15}_{1..3}\bar{S}^{19}_{1..4}$	853
${}^{15}_{1..3}S^{20}_{1..4}$	2 870	24	530	3 217	${}^{15}_{1..3}\bar{S}^{20}_{1..4}$	31
${}^{15}_{1..3}S^{21}_{1..4}$	0	-	-	-	${}^{15}_{1..3}\bar{S}^{21}_{1..4}$	0
${}^{15}_{1..3}S^{22}_{1..4}$	0	-	-	-	${}^{15}_{1..3}\bar{S}^{22}_{1..4}$	0
Phase 3						
${}^{13}_{1..3}S^{28}_{1..6}$	243 762	2 147	6 910	249 347	${}^{13}_{1..3}\bar{S}^{28}_{1..6}$	56
${}^{14}_{1..3}S^{28}_{1..6}$	998 204	9 107	38 614	1 028 665	${}^{14}_{1..3}\bar{S}^{28}_{1..6}$	506
Phase 4						
$S^{28}_{1..6}$	562	4 248	213 692	125 743	$\bar{S}^{28}_{1..6}$	6
Phase 5						
$S^{28}_{1..6}$	1	1 854	139 334	72 982	$\bar{S}^{28}_{1..6}$	1
$\sum$	3 850 807	190 664	12 598 021	10 365 984		1 075 854

"small," since we may not achieve significant reductions of the active areas for such m values. On the other hand, $\bar{S}^{m_0}_{1..3} = \emptyset$ implies that $\bar{S}^{m}_{1..3} = \emptyset$ for all $m \geq m_0$ by Corollary 6.2. We can utilize the previous observations if we evaluate the sequence $\bar{S}^{\lceil n/2 \rceil}_{1..3}, \bar{S}^{\lceil n/2 \rceil + 1}_{1..3}, \ldots, \bar{S}^{\lceil n/2 \rceil + t}_{1..3}$ until we obtain $\bar{S}^{\lceil n/2 \rceil + t}_{1..3} = \emptyset$. Then we evaluate the

Table 8.2

Computational details of solving the packing problem of $n = 29$ points. The table is organized in the same way as Table 8.1.

S	$\lvert S\rvert$	CPUt	NFE	NMAA	$\bar{S}$	$\lvert\bar{S}\rvert$
Phase 1						
$S^{15}_{1..3}$	54 264	305	14 928	22 758	$\bar{S}^{15}_{1..3}$	4 194
$S^{16}_{1..3}$	20 349	50	395	5 518	$\bar{S}^{16}_{1..3}$	40
$S^{17}_{1..3}$	5 985	8	0	1 574	$\bar{S}^{17}_{1..3}$	0
$S^{14}_{1..3}$	116 280	1 618	104 620	86 973	$\bar{S}^{14}_{1..3}$	33 794
$S^{13}_{1..3}$	203 490	3 454	338 771	230 121	$\bar{S}^{13}_{1..3}$	112 824
Phase 2						
${}^{13}_{1..3}S^{18}_{1..4}$	318 042	42 945	2 660 219	1 712 120	${}^{13}_{1..3}\bar{S}^{18}_{1..4}$	221 087
${}^{13}_{1..3}S^{19}_{1..4}$	111 682	8 323	475 178	391 266	${}^{13}_{1..3}\bar{S}^{19}_{1..4}$	35 833
${}^{13}_{1..3}S^{20}_{1..4}$	0	-	-	-	${}^{13}_{1..3}\bar{S}^{20}_{1..4}$	0
${}^{14}_{1..3}S^{18}_{1..4}$	691 320	77 284	4 638 570	3 182 171	${}^{14}_{1..3}\bar{S}^{18}_{1..4}$	383 148
${}^{14}_{1..3}S^{19}_{1..4}$	248 188	18 399	1 049 402	848 878	${}^{14}_{1..3}\bar{S}^{19}_{1..4}$	80 553
${}^{14}_{1..3}S^{20}_{1..4}$	20 685	492	15 298	31 861	${}^{14}_{1..3}\bar{S}^{20}_{1..4}$	520
${}^{14}_{1..3}S^{21}_{1..4}$	0	-	-	-	${}^{14}_{1..3}\bar{S}^{21}_{1..4}$	0
${}^{15}_{1..3}S^{18}_{1..4}$	113 761	10 142	605 305	448 297	${}^{15}_{1..3}\bar{S}^{18}_{1..4}$	49 428
${}^{15}_{1..3}S^{19}_{1..4}$	73 971	4 357	241 322	216 327	${}^{15}_{1..3}\bar{S}^{19}_{1..4}$	18 064
${}^{15}_{1..3}S^{20}_{1..4}$	20 303	605	21 069	34 290	${}^{15}_{1..3}\bar{S}^{20}_{1..4}$	1 137
${}^{15}_{1..3}S^{21}_{1..4}$	652	3	2	654	${}^{15}_{1..3}\bar{S}^{21}_{1..4}$	0
${}^{15}_{1..3}S^{22}_{1..4}$	0	-	-	-	${}^{15}_{1..3}\bar{S}^{22}_{1..4}$	0
${}^{16}_{1..3}S^{18}_{1..4}$	689	33	1 839	1 742	${}^{16}_{1..3}\bar{S}^{18}_{1..4}$	130
${}^{16}_{1..3}S^{19}_{1..4}$	855	20	999	1 471	${}^{16}_{1..3}\bar{S}^{19}_{1..4}$	60
${}^{16}_{1..3}S^{20}_{1..4}$	557	4	55	601	${}^{16}_{1..3}\bar{S}^{20}_{1..4}$	0
${}^{16}_{1..3}S^{21}_{1..4}$	114	0	0	114	${}^{16}_{1..3}\bar{S}^{21}_{1..4}$	0
${}^{16}_{1..3}S^{22}_{1..4}$	0	-	-	-	${}^{16}_{1..3}\bar{S}^{22}_{1..4}$	0
${}^{16}_{1..3}S^{23}_{1..4}$	0	-	-	-	${}^{16}_{1..3}\bar{S}^{23}_{1..4}$	0
Phase 3						
${}^{13}_{1..3}S^{29}_{1..6}$	2 349	17	67	2 400	${}^{13}_{1..3}\bar{S}^{29}_{1..6}$	1
${}^{14}_{1..3}S^{29}_{1..6}$	860 709	6 875	21 410	878 061	${}^{14}_{1..3}\bar{S}^{29}_{1..6}$	180
Phase 4						
$S^{29}_{1..6}$	181	3 785	157 168	80 731	$\bar{S}^{29}_{1..6}$	4
Phase 5						
$S^{29}_{1..6}$	1	3	166	107	$\bar{S}^{29}_{1..6}$	1
$\sum$	2 864 427	178 722	10 346 783	8 178 035		940 998

sets $\bar{S}^{\lfloor n/2\rfloor}_{1..3}$ (only if n is odd), $\bar{S}^{\lfloor n/2\rfloor-1}_{1..3}, \ldots, \bar{S}^{\lfloor n/2\rfloor-t+1}_{1..3}$. For $n = 29$, $\lceil n/2\rceil = 15$ and $\lfloor n/2\rfloor = 14$; thus, only $\bar{S}^{M_1}_{1..3}, M_1 = \{15, 16, 17, 14, 13\}$, was computed because $\bar{S}^{17}_{1..3}$ has proved to be empty. During the subsequent phases the sets $\bar{S}^{m}_{1..3}, m \leq 12$, were considered to be consisting of full tiles.

Acceleration of Phase 1. We reduced the necessary computational effort by filter-

TABLE 8.3
Computational details of solving the packing problem of $n = 30$ points. The table is organized in the same way as Table 8.1.

S	$\lvert S\rvert$	CPUt	NFE	NMAA	$\bar{S}$	$\lvert\bar{S}\rvert$
Phase 1						
$S^{15}_{1..3}$	54 264	499	27 566	30 132	$\bar{S}^{15}_{1..3}$	7 974
$S^{16}_{1..3}$	20 349	91	1 561	6 318	$\bar{S}^{16}_{1..3}$	255
$S^{17}_{1..3}$	5 985	13	20	1 590	$\bar{S}^{17}_{1..3}$	0
$S^{14}_{1..3}$	116 280	2 248	146 498	109 752	$\bar{S}^{14}_{1..3}$	47 003
Phase 2						
${}^{14}_{1..3}S^{18}_{1..4}$	0	-	-	-	${}^{14}_{1..3}\bar{S}^{18}_{1..4}$	0
${}^{14}_{1..3}S^{19}_{1..4}$	414 511	40 089	2 298 205	1 685 461	${}^{14}_{1..3}\bar{S}^{19}_{1..4}$	182 441
${}^{14}_{1..3}S^{20}_{1..4}$	53 774	3 110	117 194	129 588	${}^{14}_{1..3}\bar{S}^{20}_{1..4}$	7 146
${}^{14}_{1..3}S^{21}_{1..4}$	0	-	-	-	${}^{14}_{1..3}\bar{S}^{21}_{1..4}$	0
${}^{15}_{1..3}S^{18}_{1..4}$	53 490	7 289	432 161	281 500	${}^{15}_{1..3}\bar{S}^{18}_{1..4}$	37 167
${}^{15}_{1..3}S^{19}_{1..4}$	191 328	14 536	814 390	653 867	${}^{15}_{1..3}\bar{S}^{19}_{1..4}$	63 213
${}^{15}_{1..3}S^{20}_{1..4}$	58 757	3 602	133 421	141 579	${}^{15}_{1..3}\bar{S}^{20}_{1..4}$	8 812
${}^{15}_{1..3}S^{21}_{1..4}$	3 899	45	564	4 357	${}^{15}_{1..3}\bar{S}^{21}_{1..4}$	7
${}^{15}_{1..3}S^{22}_{1..4}$	0	-	-	-	${}^{15}_{1..3}\bar{S}^{22}_{1..4}$	0
${}^{16}_{1..3}S^{18}_{1..4}$	3 055	311	17 255	12 708	${}^{16}_{1..3}\bar{S}^{18}_{1..4}$	1 359
${}^{16}_{1..3}S^{19}_{1..4}$	7 170	368	19 367	18 519	${}^{16}_{1..3}\bar{S}^{19}_{1..4}$	1 359
${}^{16}_{1..3}S^{20}_{1..4}$	4 697	153	4 878	7 983	${}^{16}_{1..3}\bar{S}^{20}_{1..4}$	263
${}^{16}_{1..3}S^{21}_{1..4}$	1 186	13	206	1 342	${}^{16}_{1..3}\bar{S}^{21}_{1..4}$	7
${}^{16}_{1..3}S^{22}_{1..4}$	20	0	0	20	${}^{16}_{1..3}\bar{S}^{22}_{1..4}$	0
${}^{16}_{1..3}S^{23}_{1..4}$	0	-	-	-	${}^{16}_{1..3}\bar{S}^{23}_{1..4}$	0
Phase 3						
${}^{14}_{1..3}S^{30}_{1..6}$	16 799	104	10	16 807	${}^{14}_{1..3}\bar{S}^{30}_{1..6}$	0
${}^{15}_{1..3}S^{30}_{1..6}$	239 430	2 068	3 197	242 002	${}^{15}_{1..3}\bar{S}^{30}_{1..6}$	36
Phase 4						
$S^{30}_{1..6}$	36	1 391	79 916	39 935	$\bar{S}^{30}_{1..6}$	2
Phase 5						
$S^{30}_{1..6}$	1	0	18	7	$\bar{S}^{30}_{1..6}$	1
$\sum$	1 245 031	75 930	4 096 427	3 383 467		357 045

ing out symmetric cases from each set $S^m_{1..3}$. We applied the following transformations: reflections to the axes $x = 10.5$ and $y = 21$ and reflection to the point $(10.5, 21)$. Each full tile combination is identified by a binary string; thus, the filtering procedure can be done by simple transformations on these strings. After the elimination procedure made on $S^m_{1..3}$, the inverse transformations have to be applied for the remaining areas to generate the correct $\bar{S}^m_{1..3}$.

Phase 2. At this point we had $\bar{S}^m_{1..3}$ for $0 \leq m \leq 21$. We evaluated ${}^{m}_{1..3}\bar{S}^{m+i}_{1..4}$ for each necessary (m, i) pairs in two steps: first, we computed ${}^{m}_{1..3}S^{m+i}_{1..4}$ by Algorithm 4 and then ran the optimization algorithm on these sets to obtain ${}^{m}_{1..3}\bar{S}^{m+i}_{1..4}$. The parameters m and i were determined in the following way (cf. Remark 4).

We had to deal only with $m \in M_2 = \{13, 14, 15, 16\}$, where M_2 was the result of Phase 1; i.e., we added one column to each element of the set $\bar{S}^{M_2}_{1..3}$. According to Remark 4, a lower bound m_l of the number of output tiles $m + i$ could be determined by a process similar to Phase 1: $\bar{S}^{12}_{5..6}$ proved to be empty (but $\bar{S}^{11}_{5..6}$ did not). Thus, we had to choose at least $m_l = 18$ tiles from the first 4 columns. The upper bounds for $m + i$ were given by $m + 7$ for each m.

Phase 3. It is easy to see that we could proceed with adding other columns to the result patterns of Phase 2. Instead, we found that already at this time we had enough local information to join our remaining combinations by Algorithm 5 and to obtain a relatively small set of combinations consisting of 29 tiles.

As a result of Phase 2, we knew ${}^{\;i}_{1..3}\bar{S}^{m_i}_{1..4}$ for all $18 \le m_i \le 29$, $i \in M_2$. (Recall that we did not need to consider m_i values with $i \le m_i < 18$.) Since the above sets are nonempty only when $m_{13} \in M_{13} = \{18, 19\}$, $m_{14} \in M_{14} = \{18, 19, 20\}$, $m_{15} \in M_{15} = \{18, 19, 20\}$, and $m_{16} \in M_{16} = \{18, 19\}$, respectively, it was enough to consider these m_i values when applying Algorithm 5. As was suggested in the discussion of Algorithm 5, the other input sets ${}^{\;i}_{4..6}\bar{S}^{M_i}_{3..6}$, $i \in M_2$, were evaluated from ${}^{\;i}_{1..3}\bar{S}^{M_i}_{1..4}$ by rotating each element 180 degrees around the midpoint $(21, 21)$ of the square.

Thus, in Phase 3 we had to join ${}^{13}_{1..3}\bar{S}^{M_{13}}_{1..4}$ with ${}^{16}_{4..6}\bar{S}^{M_{16}}_{3..6}$ (this process is called $\mathit{Join}(13, 16)$ in what follows) and, similarly, join ${}^{14}_{1..3}\bar{S}^{M_{14}}_{1..4}$ with ${}^{15}_{4..6}\bar{S}^{M_{15}}_{3..6}$ ($\mathit{Join}(14, 15)$), then join ${}^{15}_{1..3}\bar{S}^{M_{15}}_{1..4}$ with ${}^{14}_{4..6}\bar{S}^{M_{14}}_{3..6}$ ($\mathit{Join}(15, 14)$), and, finally, join ${}^{16}_{1..3}\bar{S}^{M_{16}}_{1..4}$ with ${}^{13}_{4..6}\bar{S}^{M_{13}}_{3..6}$ ($\mathit{Join}(16, 13)$). The resulting sets ${}^{M_2}_{1..3}S^{29}_{1..6}$ had to be processed by the optimization algorithm sequentially. Since we stated that for a solution having objective function value greater than or equal to $\tilde{f}$ only 13, 14, 15, or 16 tiles can be chosen from each half of the search region, the union of the resulting sets

$${}^{13}_{1..3}\bar{S}^{29}_{1..6} \cup {}^{14}_{1..3}\bar{S}^{29}_{1..6} \cup {}^{15}_{1..3}\bar{S}^{29}_{1..6} \cup {}^{16}_{1..3}\bar{S}^{29}_{1..6}$$

can be considered as $S^{29}_{1..6}$, i.e., a set which contains all the global maximizers.

Acceleration of Phase 3. Again, symmetry properties were applied to reduce the necessary computations. Namely, without loss of generality we can assume that in each element of $S^{29}_{1..6}$ the left half of the search square contains more active regions than the right one (keeping in mind that points located on the vertical halving line segment are considered to be contained in both half squares). Consequently, we need not execute $\mathit{Join}(15, 14)$ and $\mathit{Join}(16, 13)$ when $\mathit{Join}(14, 15)$ and $\mathit{Join}(13, 16)$ are already done. Another possibility of exploiting symmetry is applied for even n numbers: for example, in $\mathit{Join}(14, 14)$ for $n = 28$ we assumed that column 3 does not contain fewer regions than column 4; i.e., $\mathit{Join}(14, 14)$ could be restricted to join each ${}^{14}_{1..3}\bar{S}^{i}_{1..4}$ with each ${}^{14}_{4..6}\bar{S}^{j}_{3..6}$ only when $j \ge i$. A similar acceleration was done for $n = 30$ in $\mathit{Join}(15, 15)$. (We presented only two very simple cases for filtering symmetric combinations, which can easily be controlled in our implementation. Obviously, one can develop more sophisticated methods resulting in larger filtering improvements.)

Phase 4. According to the previous remarks, only the two resulting sets ${}^{13}_{1..3}\bar{S}^{29}_{1..6}$ and ${}^{14}_{1..3}\bar{S}^{29}_{1..6}$ were put into a set $S^{29}_{1..6}$ to be refined. Since this set may contain a lot of equivalent solutions, we still did not use the original stopping criterion of Algorithm 1. Instead, we increased the maximal number of allowed iteration steps from 3 to 10,000. As a result, we managed to reduce the number of remaining combinations to 6 for $n = 28$, to 4 for $n = 29$, and to 2 for $n = 30$.

Phase 5. After Phase 4 it was possible to check all the remaining regions one by one. (However, for some n values, a much bigger number of combinations may remain; thus, we are planning to automatize this step in the future.) For the problem of packing 29 points the first two combinations were leading to symmetric packings (since the initial tile combinations of them were symmetric), and so did the other two combinations. But we found that the initial tile sets of the first group could not be transformed to the tile sets of the second group of combinations. The reason of this is that the 7×6 splitting is not invariant to rotations by ± 90 degrees around the midpoint of the square.

However, it was easy to see that if we split the square into 6×7 regular tiles, each remaining region of one of the latter two combinations was located in different tiles of the new splitting, and the new tile set corresponding to this combination was symmetric to the patterns of the first two combinations. Consequently, it was enough to consider one of the four combinations in the further investigations. (One can guess that four combinations should exist for both groups of remaining subproblems. Indeed, it is easy to check that the additional combinations were filtered at Phase 3 by the mentioned symmetry rules.)

In the cases of 28 and 30 points, similar observations were made, resulting in the same consequence, i.e., that one can deal only with one combination. We obtained the first main result of our study: *Let* $n \in \{28, 29, 30\}$. *Apart from symmetric cases, one initial tile combination contains all the globally optimal solutions of the packing problem of* n *points.* We shortly improve this statement by proving that all the global solutions are located in a very small area within the particular tile combination.

As a further refinement of the only remaining box, the method for guessing free points and shrinking several components of the remaining regions was applied (see section 7). We obtained that for $n = 28$ only the 12th component could be replaced by a point, for $n = 29$ only the 5th component, while for $n = 30$ no components could be replaced. (These facts are in accordance with the location of the free points of the best-known packings.) The only remaining regions were then refined by Algorithm 1 using the stopping criterion parameter of $\varepsilon = 10^{-10}$. As was also stated in [12], the cost of this local investigation was relatively small as compared to the global part of our whole procedure. Finally, after transforming the enclosures of the result boxes back to the original point packing problem, we obtained the tightest currently known enclosures of the global maximum values, which are

$$F_{28}^* = [\underline{0.23053549364266}73, \underline{0.23053549364267}43], \quad w(F_{28}^*) \approx 7 \cdot 10^{-15},$$

$$F_{29}^* = [\underline{0.2268829007442}089, \underline{0.2268829007442}240], \quad w(F_{29}^*) \approx 2 \cdot 10^{-14},$$

$$F_{30}^* = [\underline{0.22450296453108}81, \underline{0.22450296453109}03], \quad w(F_{30}^*) \approx 2 \cdot 10^{-15}.$$

The enclosure of the particular global maximizers $(X, Y)_n^*$ are reported in Tables 8.4, 8.5 and 8.6, respectively. The precision of the components are highlighted by underscores. Our conclusions are the following: *Let* $n \in \{28, 29, 30\}$. *Apart from symmetric cases, all the global optimizers of the packing problem of* n *points are located in the reported boxes having very tight components (with the exception of the components enclosing possibly free points for* $n = 28, 29$*). Moreover, the reported enclosures confirm that the structure of the currently known best packings are optimal within the precision of* $w(F_n^*)$. *In other words, all the exact global optimizers are located in* $(X, Y)_{28}^*$, $(X, Y)_{29}^*$, *and* $(X, Y)_{30}^*$, *and the exact global optima differ from the currently best known function values by at most* $w(F_n^*)$.

Table 8.4
The enclosure of the global maximizer point for packing 28 *points.*

i	X_i	Y_i
1.	$[\underline{0.0000000000000}000, \underline{0.0000000000000}142]$,	$[\underline{0.0000000000000}000, \underline{0.0000000000000}182]$
2.	$[\underline{0.0000000000000}000, \underline{0.0000000000000}723]$,	$[\underline{0.283335701951}1142, \underline{0.283335701951}2377]$
3.	$[\underline{0.0000000000000}000, \underline{0.0000000000000}617]$,	$[\underline{0.513871195593}7813, \underline{0.513871195593}9050]$
4.	$[\underline{0.0000000000000}000, \underline{0.0000000000000}689]$,	$[\underline{0.744406689236}4485, \underline{0.744406689236}5722]$
5.	$[\underline{0.00000000000000}00, \underline{0.00000000000000}94]$,	$[\underline{0.9999999999999}859, \underline{1.0000000000000}000]$
6.	$[\underline{0.1818703764471}228, \underline{0.1818703764471}709]$,	$[\underline{0.1416678509755}570, \underline{0.1416678509756}246]$
7.	$[\underline{0.199649593968}5054, \underline{0.199649593968}7475]$,	$[\underline{0.398603448772}3758, \underline{0.398603448772}5656]$
8.	$[\underline{0.199649593968}5047, \underline{0.199649593968}6531]$,	$[\underline{0.629138942415}0430, \underline{0.629138942415}2327]$
9.	$[\underline{0.1918713858351}473, \underline{0.1918713858351}900]$,	$[\underline{0.8722033446182}193, \underline{0.8722033446182}866]$
10.	$[\underline{0.36374075289}42488, \underline{0.36374075289}64056]$,	$[\underline{0.00000000000}00000, \underline{0.00000000000}15377]$
11.	$[\underline{0.381519970415}6590, \underline{0.381519970415}7629]$,	$[\underline{0.256935597796}7746, \underline{0.256935597796}9572]$
12.	$[\underline{0.39}92990052060252, \underline{0.40}23815131388759]$,	$[\underline{0.50}89492733445356, \underline{0.51}38712298523256]$
13.	$[\underline{0.3915209798036}835, \underline{0.3915209798037}213]$,	$[\underline{0.7569355977968}880, \underline{0.7569355977969}508]$
14.	$[\underline{0.3837427716702}937, \underline{0.3837427716703}746]$,	$[\underline{0.9999999999999}370, \underline{1.0000000000000}000]$
15.	$[\underline{0.56339034686}27852, \underline{0.56339034686}41770]$,	$[\underline{0.11526774682}08971, \underline{0.11526774682}28100]$
16.	$[\underline{0.5839312817244}451, \underline{0.5839312817244}956]$,	$[\underline{0.3672817571470}099, \underline{0.3672817571470}896]$
17.	$[\underline{0.5911705737722}300, \underline{0.5911705737722}556]$,	$[\underline{0.6416678509755}755, \underline{0.6416678509756}131]$
18.	$[\underline{0.5833923656388}268, \underline{0.5833923656389}046]$,	$[\underline{0.884732253178}6048, \underline{0.884732253178}7316]$
19.	$[\underline{0.763039940831}3210, \underline{0.763039940831}8696]$,	$[\underline{0.000000000000}0000, \underline{0.000000000000}1518]$
20.	$[\underline{0.7694645063572}478, \underline{0.7694645063573}329]$,	$[\underline{0.230445956325}7084, \underline{0.230445956325}8553]$
21.	$[\underline{0.77272034313109}99, \underline{0.77272034313110}68]$,	$[\underline{0.4995893688792}053, \underline{0.4995893688792}224]$
22.	$[\underline{0.7830419596073}901, \underline{0.7830419596074}357]$,	$[\underline{0.7694645063572}883, \underline{0.7694645063573}327]$
23.	$[\underline{0.7830419596073}660, \underline{0.7830419596074}357]$,	$[\underline{0.9999999999999}551, \underline{1.0000000000000}000]$
24.	$[\underline{0.993575434473}9882, \underline{0.993575434474}5461]$,	$[\underline{0.0000000000000}000, \underline{0.0000000000000}352]$
25.	$[\underline{0.9999999999999}152, \underline{1.0000000000000}000]$,	$[\underline{0.2304459563257}084, \underline{0.2304459563257}429]$
26.	$[\underline{0.99999999999999}47, \underline{1.00000000000000}00]$,	$[\underline{0.4609814499683}757, \underline{0.4609814499684}068]$
27.	$[\underline{0.9999999999999}551, \underline{1.0000000000000}000]$,	$[\underline{0.691516943611}0431, \underline{0.691516943611}1594]$
28.	$[\underline{0.9999999999999}316, \underline{1.0000000000000}000]$,	$[\underline{0.922052437253}7104, \underline{0.922052437253}9002]$

It is also worth mentioning that our verified procedure decreased the uncertainty in the location of the optimal packing by as much as

$$\log\left(w([0,1])^{2n}/\prod_{i=1}^{n} w(X_n^*)w(Y_n^*)\right),$$

i.e., more than 711, 764, and 872 orders of magnitude, respectively.

As Tables 8.1 to 8.3 show, the total time complexity of the optimality proof was approximately 53, 50, and 21 hours for the packing of 28, 29, and 30 points, respectively. Notice that the CPU time of those parts of the method not indicated

Table 8.5

The enclosure of the global maximizer point for packing 29 *points.*

i	X_i	Y_i
1.	$[\underline{0.0000000000000}000, \underline{0.0000000000000}246]$,	$[\underline{0.0000000000000}000, \underline{0.0000000000000}393]$
2.	$[\underline{0.0000000000000}000, \underline{0.0000000000000}217]$,	$[\underline{0.2268829007442}089, \underline{0.2268829007442}435]$
3.	$[\underline{0.0000000000000}000, \underline{0.0000000000000}199]$,	$[\underline{0.4537658014884}179, \underline{0.4537658014884}444]$
4.	$[\underline{0.0000000000000}000, \underline{0.0000000000000}136]$,	$[\underline{0.6806487022326}268, \underline{0.6806487022326}528]$
5.	$[\underline{0.0}000000000000000, \underline{0.0}532481705058822]$,	$[\underline{0.9}073423366158249, \underline{1.0}000000000000000]$
6.	$[\underline{0.2009548756284}472, \underline{0.2009548756284}715]$,	$[\underline{0.1053232576939}058, \underline{0.1053232576939}296]$
7.	$[\underline{0.2009548756284}483, \underline{0.2009548756284}689]$,	$[\underline{0.3322061584381}170, \underline{0.3322061584381}365]$
8.	$[\underline{0.2009548756284}484, \underline{0.2009548756284}683]$,	$[\underline{0.5590890591823}258, \underline{0.5590890591823}453]$
9.	$[\underline{0.2009548756284}486, \underline{0.2009548756284}655]$,	$[\underline{0.7859719599265}345, \underline{0.7859719599265}537]$
10.	$[\underline{0.2762399917698}153, \underline{0.2762399917698}536]$,	$[\underline{0.9999999999999}868, \underline{1.0000000000000}000]$
11.	$[\underline{0.4019097512568}943, \underline{0.4019097512569}189]$,	$[\underline{0.0000000000000}000, \underline{0.0000000000000}215]$
12.	$[\underline{0.4019097512568}967, \underline{0.4019097512569}177]$,	$[\underline{0.2268829007442}089, \underline{0.2268829007442}285]$
13.	$[\underline{0.4019097512568}968, \underline{0.4019097512569}169]$,	$[\underline{0.4537658014884}179, \underline{0.4537658014884}354]$
14.	$[\underline{0.4019097512568}973, \underline{0.4019097512569}210]$,	$[\underline{0.6806487022326}268, \underline{0.6806487022326}512]$
15.	$[\underline{0.5983961069856}828, \underline{0.5983961069857}054]$,	$[\underline{0.1134414503720}853, \underline{0.1134414503721}164]$
16.	$[\underline{0.5983961069856}840, \underline{0.5983961069857}049]$,	$[\underline{0.3403243511162}964, \underline{0.3403243511163}233]$
17.	$[\underline{0.5983961069856}844, \underline{0.5983961069857}049]$,	$[\underline{0.5672072518605}076, \underline{0.5672072518605}315]$
18.	$[\underline{0.5983961069856}861, \underline{0.5983961069857}088]$,	$[\underline{0.7940901526047}168, \underline{0.7940901526047}401]$
19.	$[\underline{0.5031228925140}242, \underline{0.5031228925140}623]$,	$[\underline{0.9999999999999}830, \underline{1.0000000000000}000]$
20.	$[\underline{0.7948824627144}688, \underline{0.7948824627144}926]$,	$[\underline{0.0000000000000}000, \underline{0.0000000000000}197]$
21.	$[\underline{0.7948824627144}706, \underline{0.7948824627144}921]$,	$[\underline{0.2268829007442}089, \underline{0.2268829007442}256]$
22.	$[\underline{0.7948824627144}710, \underline{0.7948824627144}921]$,	$[\underline{0.4537658014884}178, \underline{0.4537658014884}334]$
23.	$[\underline{0.7948824627144}749, \underline{0.7948824627144}919]$,	$[\underline{0.6806487022326}267, \underline{0.6806487022326}419]$
24.	$[\underline{0.729082658483}5373, \underline{0.729082658483}7260]$,	$[\underline{0.979554100334}8881, \underline{0.979554100335}6186]$
25.	$[\underline{0.9999999999999}812, \underline{1.0000000000000}000]$,	$[\underline{0.0969672447170}968, \underline{0.0969672447171}463]$
26.	$[\underline{0.9999999999999}812, \underline{1.0000000000000}000]$,	$[\underline{0.3238501454613}097, \underline{0.3238501454613}550]$
27.	$[\underline{0.9999999999999}820, \underline{1.0000000000000}000]$,	$[\underline{0.5507330462055}206, \underline{0.5507330462055}635]$
28.	$[\underline{0.9999999999999}859, \underline{1.0000000000000}000]$,	$[\underline{0.7776159469497}363, \underline{0.7776159469497}720]$
29.	$[\underline{0.955042424453}0482, \underline{0.955042424453}2232]$,	$[\underline{0.9999999999999}683, \underline{1.0000000000000}000]$

in the tables (such as the determination of m_l in Phase 2 and the execution time of Algorithms 4 and 5) took about only one additional minute for each problem instance. The total time complexities are remarkably less than the forecasted execution times (i.e., one or even more decades) of the earlier methods. The memory complexity of our procedure was small in most parts of the method, since the results of the intermediate phases can be saved to storage devices. Moreover, due to the sequential and quick processing of the tile combinations, the main optimization method required small amount of memory. In several cases, the memory requirement of Algorithms 4 and 5

Table 8.6

The enclosure of the global maximizer point for packing 30 *points.*

i	X_i	Y_i
1.	$[\underline{0.10198814187564}26, \underline{0.10198814187564}76]$,	$[\underline{0.00000000000000}00, \underline{0.00000000000000}26]$
2.	$[\underline{0.00000000000000}00, \underline{0.00000000000000}26]$,	$[\underline{0.19999999999999}95, \underline{0.20000000000000}22]$
3.	$[\underline{0.10198814187564}51, \underline{0.10198814187564}77]$,	$[\underline{0.39999999999999}89, \underline{0.40000000000000}17]$
4.	$[\underline{0.00000000000000}00, \underline{0.00000000000000}25]$,	$[\underline{0.59999999999999}84, \underline{0.60000000000000}11]$
5.	$[\underline{0.10198814187564}51, \underline{0.10198814187564}78]$,	$[\underline{0.79999999999999}78, \underline{0.80000000000000}06]$
6.	$[\underline{0.00000000000000}00, \underline{0.00000000000000}45]$,	$[\underline{0.99999999999999}72, \underline{1.00000000000000}00]$
7.	$[\underline{0.32649110640673}07, \underline{0.32649110640673}57]$,	$[\underline{0.00000000000000}00, \underline{0.00000000000000}28]$
8.	$[\underline{0.22450296453108}81, \underline{0.22450296453109}07]$,	$[\underline{0.19999999999999}95, \underline{0.20000000000000}22]$
9.	$[\underline{0.32649110640673}32, \underline{0.32649110640673}58]$,	$[\underline{0.39999999999999}89, \underline{0.40000000000000}17]$
10.	$[\underline{0.22450296453108}81, \underline{0.22450296453109}05]$,	$[\underline{0.59999999999999}84, \underline{0.60000000000000}11]$
11.	$[\underline{0.32649110640673}32, \underline{0.32649110640673}59]$,	$[\underline{0.79999999999999}78, \underline{0.80000000000000}06]$
12.	$[\underline{0.22450296453108}81, \underline{0.22450296453109}26]$,	$[\underline{0.99999999999999}72, \underline{1.00000000000000}00]$
13.	$[\underline{0.44900592906217}62, \underline{0.44900592906217}88]$,	$[\underline{0.19999999999999}95, \underline{0.20000000000000}21]$
14.	$[\underline{0.44900592906217}62, \underline{0.44900592906217}86]$,	$[\underline{0.59999999999999}84, \underline{0.60000000000000}11]$
15.	$[\underline{0.44900592906217}62, \underline{0.44900592906218}07]$,	$[\underline{0.99999999999999}72, \underline{1.00000000000000}00]$
16.	$[\underline{0.55099407093781}89, \underline{0.55099407093782}39]$,	$[\underline{0.00000000000000}00, \underline{0.00000000000000}25]$
17.	$[\underline{0.55099407093782}13, \underline{0.55099407093782}40]$,	$[\underline{0.39999999999999}89, \underline{0.40000000000000}16]$
18.	$[\underline{0.55099407093782}13, \underline{0.55099407093782}40]$,	$[\underline{0.79999999999999}78, \underline{0.80000000000000}06]$
19.	$[\underline{0.77549703546890}73, \underline{0.77549703546891}20]$,	$[\underline{0.00000000000000}00, \underline{0.00000000000000}24]$
20.	$[\underline{0.67350889359326}43, \underline{0.67350889359326}68]$,	$[\underline{0.19999999999999}95, \underline{0.20000000000000}20]$
21.	$[\underline{0.77549703546890}95, \underline{0.77549703546891}20]$,	$[\underline{0.39999999999999}90, \underline{0.40000000000000}15]$
22.	$[\underline{0.67350889359326}43, \underline{0.67350889359326}66]$,	$[\underline{0.59999999999999}86, \underline{0.60000000000000}11]$
23.	$[\underline{0.77549703546890}95, \underline{0.77549703546891}20]$,	$[\underline{0.79999999999999}81, \underline{0.80000000000000}06]$
24.	$[\underline{0.67350889359326}43, \underline{0.67350889359326}88]$,	$[\underline{0.99999999999999}76, \underline{1.00000000000000}00]$
25.	$[\underline{0.99999999999999}55, \underline{1.00000000000000}00]$,	$[\underline{0.00000000000000}00, \underline{0.00000000000000}23]$
26.	$[\underline{0.89801185812435}25, \underline{0.89801185812435}49]$,	$[\underline{0.19999999999999}95, \underline{0.20000000000000}19]$
27.	$[\underline{0.99999999999999}76, \underline{1.00000000000000}00]$,	$[\underline{0.39999999999999}91, \underline{0.40000000000000}15]$
28.	$[\underline{0.89801185812435}25, \underline{0.89801185812435}48]$,	$[\underline{0.59999999999999}87, \underline{0.60000000000000}11]$
29.	$[\underline{0.99999999999999}77, \underline{1.00000000000000}00]$,	$[\underline{0.79999999999999}81, \underline{0.80000000000000}06]$
30.	$[\underline{0.89801185812435}25, \underline{0.89801185812435}70]$,	$[\underline{0.99999999999999}76, \underline{1.00000000000000}00]$

was large for large sets of input combinations; however, our main memory was still enough to store all necessary combinations at one time to avoid swapping.

9. Summary and future work. We introduced a verified, interval arithmetic-based optimization method for solving circle (point) packing problems. Each part of our method was discussed through detailed algorithmic descriptions and theorems of correctness. We solved the previously unsolved problem instances of packing 28, 29, and 30 points. It was shown that the earlier found best configurations are globally optimal within very tight tolerance values. We reported the boxes enclosing the exact optimizers and stated that the current best function value differs from the maximum by at most $7 \cdot 10^{-15}$, $2 \cdot 10^{-14}$, and $2 \cdot 10^{-15}$ for $n = 28, 29$, and 30, respectively.

Our method contained two key procedures: On the one hand, we developed a very efficient area reduction algorithm. This method was based on the interval adaptation of the earlier known polygon representation procedure. On the other hand, a very effective technique based on subsets of tile combinations was introduced for handling and eliminating groups of subproblems together. As a result, the total computational time was extremely smaller than the predicted time complexity of the earlier methods.

The basic elements of the proposed method will very probably play a role in further studies of both circle packing problems for $n \geq 31$ and the generalizations of the problem class. In the next unsolved case, $n = 31$, already a 7×7 tile splitting can be applied, which will probably require an additional intermediate phase for "growing" the dimensionality of the considered subproblems. Thus, we are planning to develop our accelerated method for eliminating tile combinations in a more general way. The first part of this work would consist of programming features, such as extending Algorithm 4 to include the ability of adding new rows of tiles and modifying Algorithm 5 to handle general intersecting tile positions of the combinations to be joined. By using this more general approach, more sophisticated multiphase methods could be designed, reducing the total cost of processing intermediate tile combinations, and opening the way for solving much harder circle packing problems and their generalizations.

REFERENCES

[1] L. G. Casado, I. García, P. G. Szabó, and T. Csendes, *Equal circles packing in a square* II. *New results for up to* 100 *circles using the TAMSASS-PECS algorithm*, in Optimization Theory: Recent Developments from Mátraháza, F. Giannessi, P. Pardalos, and T. Rapcsák, eds., Kluwer Academic, Dordrecht, 2001, pp. 207–224.

[2] T. Csendes, *Nonlinear parameter estimation by global optimization: Efficiency and reliability*, Acta Cybernet., 8 (1988), pp. 361–370.

[3] T. Csendes and D. Ratz, *Subdivision direction selection in interval methods for global optimization*, SIAM J. Numer. Anal., 34 (1997), pp. 922–938.

[4] R. L. Graham and B. D. Lubachevsky, *Repeated patterns of dense packings of equal disks in a square*, Electron. J. Combin., 3 (1996), article 16.

[5] C. de Groot, M. Monagan, R. Peikert, and D. Würtz, *Packing circles in a square: Review and new results*, in System Modeling and Optimization, P. Kall, ed., Lecture Notes in Control and Inform. Sci. 180, Springer-Verlag, Berlin, 1992, pp. 45–54.

[6] C. de Groot, R. Peikert, and D. Würtz, *The Optimal Packing of Ten Equal Circles in a Square*, IPS Research Report 90-12, ETH, Zürich, 1990.

[7] R. Hammer, M. Hocks, U. Kulisch, and D. Ratz, *Numerical Toolbox for Verified Computing* I, Springer-Verlag, Berlin, 1993.

[8] E. Hansen, *Global Optimization Using Interval Analysis*, Marcel Dekker, New York, 1992.

[9] O. Knüppel, *PROFIL: Programmer's Runtime Optimized Fast Interval Library*, Bericht 93.4, Technische Universität Hamburg–Harburg, 1993.

[10] O. Knüppel, *A Multiple Precision Arithmetic for PROFIL*, Bericht 93.6, Technische Universität Hamburg–Harburg, 1993.

[11] M. Locatelli and U. Raber, *Packing equal circles in a square: A deterministic global optimization approach*, Discrete Appl. Math., 122 (2002), pp. 139–166.

[12] M. Cs. Markót, *An interval method to validate optimal solutions of the "packing circles in a unit square" problems*, CEJOR Cent. Eur. J. Oper. Res., 8 (2000), pp. 63–78.

[13] M. Cs. Markót, *Optimal packing of* 28 *equal circles in a unit square: The first reliable solution*, Numer. Algorithms, 37 (2004), pp. 253–261.

[14] M. Cs. Markót and T. Csendes, *A reliable area reduction technique for solving circle packing problems*, submitted for publication; available online from http://www.inf.u-szeged.hu/˜markot/intpoly.ps.gz.

[15] M. Cs. Markót, T. Csendes, and A. E. Csallner, *Multisection in interval branch-and-bound methods for global optimization* II. *Numerical tests*, J. Global Optim., 16 (2000), pp. 219–228.

[16] R. E. Moore, *Interval Analysis*, Prentice–Hall, Englewood Cliffs, NJ, 1966.

[17] K. J. Nurmela and P. R. J. Östergård, *Packing up to* 50 *circles in a square*, Discrete Comput. Geom., 18 (1997), pp. 111–120.

[18] K. J. Nurmela and P. R. J. Östergård, *Optimal packings of equal circles in a square*, in Combinatorics, Graph Theory, and Algorithms, Y. Alavi, D. R. Lick, and A. Schwenk, eds., New Issues Press, Kalamazoo, MI, 1999, pp. 671–680.

[19] K. J. Nurmela and P. R. J. Östergård, *More optimal packings of equal circles in a square*, Discrete Comput. Geom., 22 (1999), pp. 439–457.

[20] R. Peikert, *Dichteste Packungen von gleichen Kreisen in einem Quadrat*, Elem. Math., 49 (1994), pp. 17–25.

[21] H. Ratschek and J. Rokne, *New Computer Methods for Global Optimization*, Ellis Horwood, Chichester, 1988.

[22] http://www.packomania.com, updated by Eckard Specht.

[23] P. G. Szabó, *Some new structures for the "equal circles packing in a square" problem*, CEJOR Cent. Eur. J. Oper. Res., 8 (2000), pp. 79–91.

SIAM J. Optim.
Vol. 16, No. 1, pp. 220–234

TRANSFORMATION OF FACETS OF THE GENERAL ROUTING PROBLEM POLYTOPE*

GERHARD REINELT† AND DIRK OLIVER THEIS†

Abstract. We study the class of polytopes associated with the 0/1-variable formulation by [G. Ghiani and G. Laporte, *Math. Program.*, 87 (2000), pp. 467–481] of the rural postman problem and general routing problem. We introduce a technique called "switching," which allows one to obtain new valid or facet-defining inequalities from known ones. We study the switched forms which can be obtained from two important types of known valid inequalities, namely, so called path-bridge [A. N. Letchford, *Eur. J. Oper. Res.*, 96 (1997), pp. 317–322] and honeycomb [A. Corberán and J. M. Sanchis, *Eur. J. Oper. Res.*, 108 (1998), pp. 538–550] inequalities. For the switched path-bridge inequalities, a generic blueprint for separation routines is developed. We give a polynomial time separation algorithm for a special subclass of these inequalities.

Key words. polytope, facets, separation, general routing problem, rural postman problem

AMS subject classifications. 90C27, 90C57, 52B15, 90C60

DOI. 10.1137/040607307

1. Introduction.

Given a loopless connected graph G, a vector $c \in \mathrm{R}_{+}^{E(G)}$ of nonnegative edge costs, a set of *required nodes* $V_R \subseteq V(G)$, and a set of *required edges* $E_R \subseteq E(G)$, the *general routing problem (GRP)* consists of finding a closed walk in G containing the required nodes and edges, such that the sum of the edge costs is minimized [15, 11]. The GRP includes as a special case the *rural postman problem (RPP),* obtained when $V_R = \emptyset$. If $E_R = \emptyset$ and $V_R = V(G)$, we obtain the *graphical traveling salesman problem (GTSP).* The RPP and GTSP (hence also the GRP) are NP-hard combinatorial optimization problems [11, 7].

There exists a preprocessing procedure for the GRP which allows us to assume without loss of generality that $V_R = V(G)$, e.g., [3, 2]. In this paper we consider only this case and assume that all nodes are required.

A *semitour* [5, 6] is a vector $x \in \mathrm{Z}_{+}^{E(G)}$, with the property that $x + \chi^{E_R}$ is a spanning closed walk (which is equivalent to being a feasible solution to the GRP because $V_R = V(G)$). Here χ^F denotes the characteristic vector of a set F, i.e., $\chi^F_e = 1$ if $e \in F$, and $\chi^F_e = 0$ otherwise. Clearly, finding a minimum cost feasible solution to the GRP is equivalent to finding a semitour x with minimum cost $cx := \sum_{e \in E(G)} c_e x_e$. The unbounded polyhedron which is the convex hull of all semitours has been extensively studied [5, 6, 12, 13].

Before we characterize the set of semitours, we introduce some more terminology. First, we define the *parity* of a node $u \in V(G)$ as

$$\mathbf{t}(u) := \begin{cases} 0 & \text{if } |\delta(u) \cap E_R| \text{ is even,} \\ 1 & \text{otherwise.} \end{cases} \tag{1.1}$$

*Received by the editors April 26, 2004; accepted for publication (in revised form) December 22, 2004; published electronically September 8, 2005.

http://www.siam.org/journals/siopt/16-1/60730.html

†Discrete Optimization Group, Institute of Computer Science, University of Heidelberg, Heidelberg, Germany (Gerhard.Reinelt@informatik.uni-heidelberg.de, Dirk.Theis@informatik.uni-heidelberg.de).

Here, for a node set U, we denote by $\delta(U)$ the set of edges with precisely one end node in U and abbreviate $\delta(\{u\})$ by $\delta(u)$. Hence, $\mathbf{t}(u) = 1$, if there is an odd number of required edges incident to u, and $\mathbf{t}(u) = 0$, if the number is even. Further, consider the subgraph of G with node set $V(G)$ and edge set E_R. Its connected components are called *R-components*, and the node set of an R-component is called an *R-set.* The R-sets form a partition of the node set $V(G)$. We denote the set of all R-sets by $\mathcal{C}$. We also define the graph $G_{\mathcal{C}}$, which results from shrinking each R-set to a single node while deleting loops but keeping parallel edges. The node of $G_{\mathcal{C}}$ which results from the R-set C is again denoted by C, i.e., we assume that the node set of $G_{\mathcal{C}}$ is $\mathcal{C}$. The edge set of $G_{\mathcal{C}}$ is a subset of $E(G)$, consisting of all edges which have their end nodes in different R-sets. We call these edges *R-external,* and we call edges of G with both end nodes in the same R-set *R-internal.* The set of all R-internal edges is denoted by $E_{\text{int}}(G)$. If a set $C \in \mathcal{C}$ is a singleton set $C = \{v\}$, then we say that v is *R-isolated.*

The following system characterizes the set of semitours [5, 6]:

$$(1.2a) \qquad x(\delta(u)) = \mathbf{t}(u) \bmod 2 \qquad \text{for all } u \in V(G),$$

$$(1.2b) \qquad x(\delta(S)) \geq 2 \qquad \text{for all } S = \bigcup_{C \in \mathcal{S}} C,\ \emptyset \neq \mathcal{S} \subsetneq \mathcal{C},$$

$$(1.2c) \qquad x \in \mathbb{Z}_+^{E(G)},$$

where, for any set of edges $F \subseteq E$, we let $x(F) := \sum_{f \in F} x_f$. We denote the set of all semitours, i.e., the set of all solutions to system (1.2a–c), by $\mathscr{S}^\infty$. The (modular) equations (1.2a) are called *parity constraints,* and the inequalities (1.2b) are called *connectivity inequalities.*

For this characterization of semitours we need only the R-set partition $\mathcal{C}$ and the parities, but not the required edges. In fact, for symmetry reasons, it will prove useful to restrict the attention to some axioms on the triple $(G, \mathcal{C}, \mathbf{t})$, and forget about required edges altogether. Let the addition in the 2-element group $\mathrm{GF}(2) = \{0, 1\}$ be denoted by "$\oplus$." We call a triple $\Gamma := (G, \mathcal{C}, \mathbf{t})$ a *GRP-structure,* if

1. G is a connected graph, $\mathcal{C}$ is a partition of $V(G)$, and $\mathbf{t}$ is a mapping
$$\mathbf{t} \colon V(G) \to \{0, 1\};$$
2. for each $C \in \mathcal{C}$, the induced subgraph $G[C]$ is connected;
3. for each $C \in \mathcal{C}$, the *parity* of the set C, which is defined as
$$\mathbf{t}(C) := \bigoplus_{u \in C} \mathbf{t}(u),$$
is zero. A mapping $t \colon V(G) \to \{0, 1\}$ with this property is called a *parity function.*

Note that, with this definition, we could assume that G is a simple graph. If the graph on which the GRP instance is defined is not simple, we can delete from each set of parallel edges all but the edge with least cost, even if that involves the deletion of required edges, as long as the original set E_R is used in (1.1). This is possible because we define the GRP-structure without using required edges. Thus every GRP-instance defines a GRP-structure with a simple graph, and every GRP-structure with a simple graph can be derived from at least one GRP-instance (although there may be many different sets of required edges which define the same GRP-structure). However, we will replace G by a graph which has parallel edges by construction, so we do not require G to be simple here.

For the remainder of the paper, we will forget about required edges. This in fact simplifies the arguments in many places.

The Ghiani–Laporte polytope. Ghiani and Laporte [8] showed that there exists a spanning tree of $G_{\mathcal{C}}$, whose edge set we denote by T, with the property that $\min\{cx \mid x \in \mathscr{S}^\infty\} = \min\{cx \mid x \in \mathscr{S}^\infty, x \leq \mathbf{1} + \chi^T\}$, where we denote the all-ones vector of appropriate length by $\mathbf{1}$. This means that we can bound the number of times an edge is contained in a semitour by 1 if the edge is not in the tree T, and by 2 if it is. In fact, Ghiani and Laporte [8] proved that any minimum spanning tree of $G_{\mathcal{C}}$, with respect to the cost vector c restricted to the R-external edges, has this property.

Let $G^T := G + T$ denote the graph which results, if we duplicate in G, each of the edges of the tree T. By considering semitours on G^T instead of on G, we can restrict our attention to semitours x with $x \leq \mathbf{1}$. For the rest of the paper, we replace G by G^T, or, from a practical viewpoint, we assume that the duplication of the edges has been performed by some preprocessing procedure. Having agreed on that, we write

$$\begin{aligned} \mathscr{S}(\mathit{\Gamma}) \quad &:= \Big\{x \in \{0,1\}^{E(G)} \mid x \text{ satisfies (1.2a–c)}\Big\}, \\ \mathrm{GRP}(\mathit{\Gamma}) &:= \mathrm{conv}\big(\mathscr{S}(\mathit{\Gamma})\big). \end{aligned}$$

The 0/1-polytope $\mathrm{GRP}(\mathit{\Gamma})$ was introduced in [8]. Therefore we call it the *Ghiani–Laporte polytope.* Ghiani and Laporte [8] gave an IP-formulation for optimizing over $\mathrm{GRP}(\mathit{\Gamma})$. Besides the *trivial inequalities* $x \geq \mathbf{0}$ (where $\mathbf{0}$ is the all-zero vector of appropriate length) and $x \leq \mathbf{1}$, it consists of the connectivity inequalities (1.2b) and the so-called *cocircuit inequalities,*

$$x(\delta(U) \setminus F) - x(F) \geq 1 - |F| \qquad (1.3)$$
$$\text{for } \emptyset \neq U \subsetneq V(G) \text{ and } F \subseteq \delta(U) \text{ with } |F| + \mathbf{t}(U) \text{ odd.}$$

These inequalities are valid for $\mathrm{GRP}(\mathit{\Gamma})$ [8]. For the IP-formulation, only the cocircuit inequalities where U is a singleton node are needed. In their simplest form, namely, if $F = \emptyset$, cocircuit inequalities are also called *R-odd cut inequalities* [5].

Computational experiments have been performed for both the unbounded polyhedron $\mathrm{conv}(\mathscr{S}^\infty)$ [4] and the formulation by Ghiani and Laporte [8, 19]. The results indicate that the 0/1-variable formulation is promising. One of the results of [19] is that of the 40 GRP instances investigated in [4], only 18 remain for which the optimal lower bound cannot be achieved by a pure cutting plane algorithm using only the two basic kinds of inequalities which are described in [8]. With the exception of only six instances, the same or, in many cases, a better lower bound than in [4] can be achieved using the IP-formulation by Ghiani and Laporte and only two of the four nonbasic classes of inequalities used in [4].

Overview of the paper. In this paper, we investigate the Ghiani–Laporte polytope. The remainder of this paper is organized as follows. Section 2 introduces a switching technique, by which we can extend any valid inequality to a whole class of inequalities, and each member of the class defines a face of the same dimension as that of the face induced by the original inequality. In section 3, we expand the switching idea of the second section and obtain even larger classes of valid inequalities. Here we cannot be certain that the inequalities in the larger class define faces of high dimensions, so this question has to be settled for each class individually.

Sections 4 and 5 show examples of switched inequalities, and address the question of facet defining property. In section 4 we describe switched variants of the so-called honeycomb [6] inequalities, while section 5 is dedicated to the switched forms of the so-called path-bridge inequalities [12]. We prove facet-defining property of switched path-bridge inequalities by giving an intriguing geometric relation between the face induced by a switched path-bridge inequality on $\mathit{\Gamma}$, and the face induced by a certain classical path-bridge inequality on a modified GRP-structure $\mathit{\Gamma}'$.

Finally, in the last section, we show how separation routines for the original path-bridge inequalities can be used to find violated switched path-bridge inequalities. We also give a polynomial time separation algorithm for so-called switched simple 2-regular path-bridges.

2. Symmetry and isomorphism of GRP polytopes. We now introduce a transformation method which allows us to create new facets from known ones. We refer to [20] for terminology related to polytopes. What follows is based on ideas for the cycle polytope in [1]. Let m be a positive integer. For $y \in \{0,1\}^m$ we define the following mapping:

$$f_y\colon \mathrm{R}^m \to \mathrm{R}^m\colon x \mapsto y + x^{[y]}, \quad \text{where } (x^{[y]})_e := \begin{cases} x_e & \text{if } y_e = 0 \\ -x_e & \text{if } y_e = 1 \end{cases}.$$

Remark 2.1. The mapping f_y is an affine isomorphism. For two vectors $y_1, y_2 \in \{0,1\}^m$, $f_{y_1 \oplus y_2}$ is equal to the composition of the mappings f_{y_1} and f_{y_2}, while $f_{\mathbf{0}}$ is the identity mapping.

Let $\mathit{\Gamma} = (G, \mathcal{C}, \mathbf{t})$ be a GRP-structure and $y \in \{0,1\}^{E(G)}$ such that

$$y_e = 0 \quad \text{for all } e \notin E_{\text{int}}(G). \tag{2.1}$$

Define $\mathbf{t}_y\colon v \mapsto \mathbf{t}(v) \oplus \bigoplus_{e\in\delta(v)} y_e$ and $\mathit{\Gamma}_y := (G, \mathcal{C}, \mathbf{t}_y)$. Then $\mathit{\Gamma}_y$ is a GRP-structure and we have

$$x \in \mathscr{S}(\mathit{\Gamma}) \text{ if and only if } f_y(x) \in \mathscr{S}(\mathit{\Gamma}_y). \tag{2.2}$$

Note that $\mathit{\Gamma}_{\mathbf{0}} = \mathit{\Gamma}$ and $\mathit{\Gamma}_{y_1 \oplus y_2} = (\mathit{\Gamma}_{y_1})_{y_2}$. A consequence is the following proposition.

PROPOSITION 2.2. *Let y and $\mathit{\Gamma}_y$ be as just described. The mapping f_y is an isomorphism between the polytopes* $\mathrm{GRP}(\mathit{\Gamma})$ *and* $\mathrm{GRP}(\mathit{\Gamma}_y)$.

Proof. From (2.2) it follows that f_y is a bijection between the vertices of the two polytopes. Since f_y is an affine isomorphism, this implies the correctness of the proposition. □

We can apply the proposition in two ways. If

$$\bigoplus_{e\in\delta(v)} y_e = 0 \qquad \text{for every } v \in V(G), \tag{2.3}$$

then the parities remain unchanged and we can study symmetries of the polytope $\mathrm{GRP}(\mathit{\Gamma})$.

COROLLARY 2.3. *The symmetry group of* $\mathrm{GRP}(\mathit{\Gamma})$ *includes a subgroup isomorphic to $\{0,1\}^r$, where $r := |E_{\text{int}}(G)| - |\mathcal{C}|$.*

Proof. The subgroup H of $\{0,1\}^{E(G)}$ consisting of all $y \in \{0,1\}^{E(G)}$ which satisfy (2.1) and (2.3) is just the cycle space of the subgraph of G induced by $E_{\text{int}}(G)$, and r is the dimension of that space. Hence, the group $\{0,1\}^r$ is isomorphic to H. □

If we drop condition (2.3), then we have isomorphisms between polytopes arising from different parity functions.

COROLLARY 2.4. *Let G be a graph and let $\mathcal{C}$ be a partition of its node set. All polytopes* $\mathrm{GRP}(\Gamma)$*, with* $\Gamma = (G, \mathcal{C}, \mathbf{t})$ *for an arbitrary parity function* $\mathbf{t}$*, are isomorphic.*

In terms of required edges, this last corollary states that if two sets of required edges define the same R-sets, then their Ghiani–Laporte polytopes are isomorphic.

The practical value of Proposition 2.2 is the following. Given an inequality $ax \geq \alpha$ and a vector y as above, consider the inequality $a^{[y]}\, x \geq \alpha - ya$. We say that the inequality $ax \geq \alpha$ is *switched* to obtain the inequality $a^{[y]}\, x \geq \alpha - ya$. The slacks of the two inequalities are related by the isomorphism f_y,

$$\alpha - ya \; - \; a^{[y]}\, f_y(x) = \alpha - ax. \tag{2.4}$$

With this relation, we obtain the following proposition as an immediate consequence of Proposition 2.2.

PROPOSITION 2.5.

a. *The inequality $ax \geq \alpha$ is valid for* $\mathrm{GRP}(\Gamma)$ *if and only if $a^{[y]}\, x \geq \alpha - ya$ is valid for* $\mathrm{GRP}(\Gamma_y)$*.*

b. *The mapping f_y is an isomorphism between the face of* $\mathrm{GRP}(\Gamma)$ *induced by $ax \geq \alpha$ and the face of* $\mathrm{GRP}(\Gamma_y)$ *induced by $a^{[y]}\, x \geq \alpha - ya$.*

In particular, the inequality $ax \geq \alpha$ is facet-defining for $\mathrm{GRP}(\Gamma)$ *if and only if $a^{[y]}\, x \geq \alpha - ya$ is facet-defining for* $\mathrm{GRP}(\Gamma_y)$*.* □

If y ranges over all elements of $\{0,1\}^{E(G)}$ which satisfy (2.1) and (2.3), then the switched inequalities $a^{[y]}\, x \geq \alpha - ya$ form a symmetry class of the original inequality $ax \geq \alpha$.

If, more generally, $bx \geq \beta$ is any inequality, and there exists a y with (2.1) and an inequality $ax \geq \alpha$ which is valid (facet-defining) for $\mathrm{GRP}(\Gamma_y)$ such that $b = a^{[y]}$ and $\beta = \alpha - ya$, then we know that $bx \geq \beta$ is valid (facet-defining) for $\mathrm{GRP}(\Gamma)$. In particular, when examining the polytope and facets, we can assume without loss of generality that all nodes have even parity, i.e., $\mathbf{t}(u) = 0$ for all $u \in V(G)$.

3. Relaxation of valid inequalities. We describe the operation of *merging* R-sets. Let $\Gamma = (G, \mathcal{C}, \mathbf{t})$ be a GRP-structure and $f \in E(G)$. Let C_1^f and C_2^f be the R-sets which contain the end nodes of f. Then the R-set partition which results from *merging along* f is defined as

$$\mathcal{C} \circ f := \mathcal{C} \setminus \{C_1^f, C_2^f\} \; \cup \{C_1^f \cup C_2^f\}.$$

This means that the sets C_1^f and C_2^f of the partition are replaced by their union $C_1^f \cup C_2^f$. Let F be a set of edges of G. The R-set partition $\mathcal{C} \circ F$ is defined successively for all $f \in F$ in any order merging along f.

We have $\mathscr{S}(\Gamma) \subseteq \mathscr{S}(G, \mathcal{C} \circ F, \mathbf{t})$, so going from $\mathcal{C}$ to $\mathcal{C} \circ F$ is a relaxation.

Now we abbreviate $y := \chi^F \in \{0,1\}^{E(G)}$ and define $\Gamma_F := (G, \mathcal{C} \circ F, \mathbf{t}_y)$. Suppose that $ax \geq \alpha$ is a valid inequality for $\mathrm{GRP}(\Gamma_F)$. Then it follows from Proposition 2.5 that the inequality $a^{[y]}\, x \geq \alpha - ya$ is valid for $\mathrm{GRP}(G, \mathcal{C} \circ F, \mathbf{t})$. It follows trivially that the inequality is also valid for $\mathrm{GRP}(\Gamma)$.

We still say that the derived inequality $a^{[y]}\, x \geq \alpha - ya$ is obtained by *switching* the valid inequality $ax \geq \alpha$. The switching method of section 2 clearly is the special case when $F \subseteq E_{\mathrm{int}}$. Only in this case do we get the facet-defining property of $a^{[y]}\, x \geq \alpha - ya$ for free, if $ax \geq \alpha$ is facet-defining. If F contains an R-external edge, it is not guaranteed that the switched inequality defines a face of high dimension.

However, we note that there are cases when the classical form of the so-called path-bridge inequalities, which we will define in section 5, is dominated by a switched variant.[1] Moreover, in [18], we show that switched path-bridge inequalities even dominate connectivity inequalities (1.2b) in certain situations. We will address the issue of facet-defining property of switched forms of some classes of valid inequalities in the next two sections.

Remark 3.1. Using the notion of required edges and assuming that the edges in F are not required, the switching process can be described as follows:

1. Make the edges in F required.
2. Find an inequality $ax \geq \alpha$ which is valid in this new situation.
3. Switch the inequality $ax \geq \alpha$ according to section 2. The resulting inequality is valid for the polytope defined with the original set of required edges.

The analogy fails if $F \cap E_R \neq \emptyset$.

Remark 3.2. All the classical inequalities discussed in this paper have in common that they require a GRP-structure with at least some nontrivial R-sets, i.e., R-sets C with $|C| \geq 2$. However, if the set F is chosen appropriately, the resulting inequalities are valid for the 0/1-polytope of the graphical TSP, i.e., the polytope which is defined as the convex hull of all incidence vectors of spanning (connected) Eulerian subgraphs of the graph G. Thus our results have consequences for the polytope (and therefore the solution) of the TSP.

We close this section with two trivial examples. First, the trivial inequalities are related by switching: $x_e \leq 1$ can be obtained by switching the inequality $x_e \geq 0$. Second, all cocircuit inequalities are switched R-odd-cut inequalities. The switching technique thus allows a new understanding of the cocircuit inequalities.

4. Switched honeycomb inequalities. *Honeycomb (HC-) inequalities* were first discussed in [6]. We briefly review the definition, but we avoid the notion of required edges and instead will speak only of the R-set partition and parity function. This is a prerequisite for using the switching idea. The general definition involves sequential lifting to compute some coefficients. For simplicity, we deal with the case when sequential lifting is not necessary (condition 5 below). We then show what switched honeycombs look like.

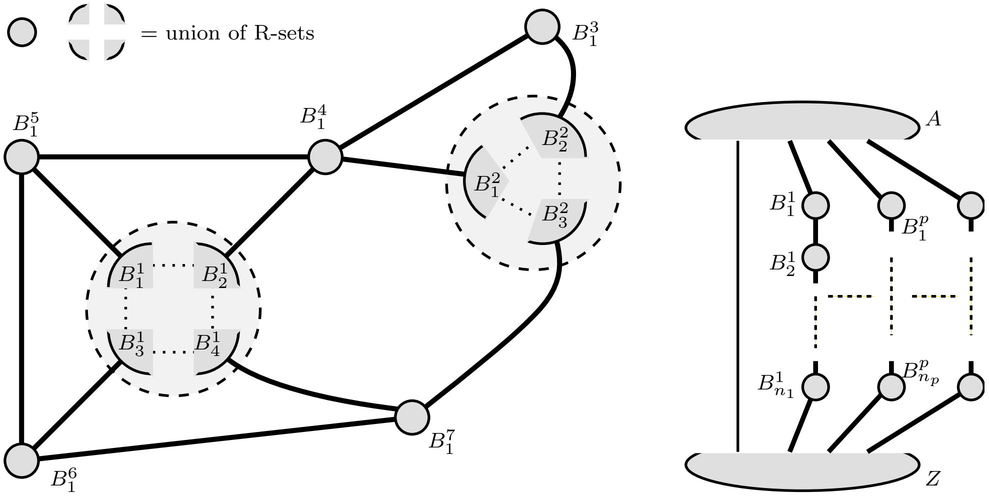
Let there be numbers $0 < L < K$ and $n_a \geq 2$, $a = 1, \ldots, L$, and a partition of $V(G)$ into sets

$$\begin{aligned} &B_b^a \ \text{ for } b = 1, \ldots, n_a \text{ and } a = 1, \ldots, L, \\ &B_1^a \ \text{ for } a = L+1, \ldots, K. \end{aligned}$$

We construct a graph G_B by shrinking in G each of the sets B_b^a into one node which we will again denote by B_b^a. Thus G_B has $K - L + n_1 + \cdots + n_L$ nodes. Further, let T be a spanning tree in G_B. See Figure 4.1 for an illustration. The following conditions are expected to hold:

1. For any $a = 1, \ldots, L$, the distance $\operatorname{dist}_T(B_{b_1}^a, B_{b_2}^a)$ between any two distinct nodes $B_{b_1}^a$ and $B_{b_2}^a$ in the tree T is greater than or equal to three.
2. Each of the sets $\bigcup_{b=1}^{n_a} B_b^a$, for $a = 1, \ldots, L$, and B_1^a, $a = L+1, \ldots, K$, is a union of R-sets.
3. The sets B_b^a are even, i.e., $\mathbf{t}(B_b^a) = 0$.

[1] Personal communication with A. Letchford, 2001.



(a) *Honeycomb configuration with $L = 2$ and $K = 7$.* (b) *Path-bridge configuration.*

FIG. 4.1. *Definition of honeycomb and path-bridge configurations.*

4. For $a \in \{1, \ldots, L\}$, consider the graph G^a with node set $\{1, \ldots, n_a\}$, and where two nodes i, j are neighbors if the sets B_i^a and B_j^a are linked by an R-internal edge in G. The graph G^a must be connected.
5. The leaves of the tree T are precisely the nodes B_b^a, $b = 1, \ldots, n_a$, $a = 1, \ldots, L$.

The condition in [6] which uses required edges is replaced by conditions 3 and 4. A node partition and tree of this kind is called a *honeycomb configuration.*

The coefficients of the honeycomb inequality $ax \geq 2(K-1)$ are defined as follows:

$$a_e := \begin{cases} \operatorname{dist}_T(B_{b_1}^a, B_{b_2}^a) - 2 & \text{if } e \in (B_{b_1}^a : B_{b_2}^a), \text{ with } b_1 \neq b_2, \\ \operatorname{dist}_T(B_{b_1}^{a_1}, B_{b_2}^{a_2}) & \text{if } e \in (B_{b_1}^{a_1} : B_{b_2}^{a_2}), \text{ with } a_1 \neq a_2, \\ 0 & \text{otherwise (i.e., } e \in E(B_b^a)). \end{cases}$$

It can be shown that the HC-inequalities define facets of the 0/1-polytope under the following conditions (see [17]; we note that neither condition is necessary):

- for all B_b^a, $B_{b'}^{a'}$ which are adjacent in T we have $\left|(B_b^a : B_{b'}^{a'})\right| \geq 2$, and
- every induced subgraph $G[B_j^p]$ is a single node or 8-edge connected.

For the definition of the *switched HC-inequalities,* in addition to the partition of the node set into sets B_b^a, we need sets of edges F^a, $a = 1, \ldots, L$, such that for each $f \in F^a$ there exist two distinct $b_1(f), b_2(f) \in \{1, \ldots, n_a\}$ with $f \in (B_{b_1(f)}^a : B_{b_2(f)}^a)$; see Figure 4.2. We replace items 3 and 4 by the following:

3a. The relation $\mathbf{t}(B_b^a) + |F^a \cap \delta(B_b^a)| = 0 \bmod 2$ holds for each $b = 1, \ldots, n_a$ and $a = 1, \ldots, L$.

4a. For $a \in \{1, \ldots, L\}$ consider the graph G^a with node set $\{1, \ldots, n_a\}$, and where two nodes i, j are neighbors if the sets B_i^a and B_j^a are linked by an R-internal edge or an edge in F^a. The graph G^a must be connected.

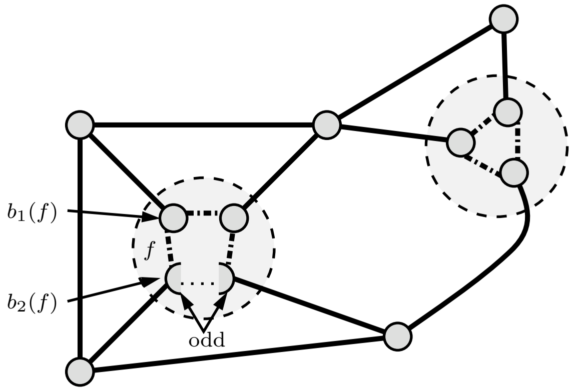


FIG. 4.2. *Switched honeycomb.*

We give the coefficients of the switched honeycomb inequality:

$$a_e := \begin{cases} \mathrm{dist}_T(B^a_{b_1}, B^a_{b_2}) - 2 & \text{if } e \in (B^a_{b_1} : B^a_{b_2}) \setminus F^a, \text{ for } b_1 \neq b_2, \\ 2 - \mathrm{dist}_T(B^a_{b_1(f)}, B^a_{b_2(f)}) & \text{if } e \in F^a, \\ \mathrm{dist}_T(B^{a_1}_{b_1}, B^{a_2}_{b_2}) & \text{if } e \in (B^{a_1}_{b_1} : B^{a_2}_{b_2}), \text{ for } a_1 \neq a_2, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

The right-hand side is

$$\alpha := 2(K-1) + 2\sum_a |F^a| - \sum_a \sum_{f \in F^a} \mathrm{dist}_T(B^a_{b_1(f)}, B^a_{b_2(f)}). \tag{4.2}$$

The switched honeycomb inequality $ax \geq \alpha$ is valid for the Ghiani–Laporte polytope. By Proposition 2.5, the inequality defines a face of GRP(Γ) which has the same dimension as the face of GRP($\Gamma_{\chi^{F \cap E_{\mathrm{int}}}}$) induced by the inequality $a'x \geq \alpha'$, where a' and α' are defined as in (4.1) and (4.2), but with F^a replaced by $F^a \setminus E_{\mathrm{int}}$ for all a. Hence, if $F \subseteq E_{\mathrm{int}}$, then Proposition 2.5 shows that the inequality defines a facet of the polytope under the conditions mentioned above for the classical honeycomb inequalities. For the case that $F \setminus E_{\mathrm{int}} \neq \emptyset$, it can be shown that the switched honeycomb inequalities define facets of GRP(Γ) under the same conditions (see [19]).

5. Switched path-bridge inequalities. The TSP's *path inequalities* [7] have scions among the valid inequalities of the GRP, as explored in [12], where it is shown that the *path-bridge (PB-) inequalities* define facets of the unbounded GRP polyhedron under weak conditions.

Let $P \geq 1$ and $n_p \geq 2$, $p = 1, \ldots, P$, be integers and let there be a partition of the node set of G into sets A, Z, B^j_p, $j = 1, \ldots, n_p$, $p = 1, \ldots, P$. The following conditions must hold:

1. Each of the B^p_j, $j = 1, \ldots, n_p$, $p = 1, \ldots, P$, is a union of R-sets.
2. For the parities we need $P + \mathbf{t}(A) = 1 \bmod 2$. If A is a union of R-sets (which implies that Z is also), the relation $P \geq 3$ must hold.

For ease of notation we let $B^p_0 := A$ and $B^p_{n_p+1} := Z$ for all $p = 1, \ldots, P$. We say that $B^p_0, \ldots, B^p_{n_p+1}$ is the pth *path* of the configuration. See Figure 4.1 for an illustration.

Define coefficients on the edges as follows:

(5.1)

$$a_e := \begin{cases} 1 & \text{if } e \in (A:Z), \\ \frac{|l-j|}{n_p-1} & \text{if } e \in (B_j^p : B_l^p) \text{ for } (j,l) \neq (0,n_p),(n_p,0), \\ \frac{1}{n_p-1} + \frac{1}{n_q-1} + \left|\frac{j-1}{n_p-1} - \frac{l-1}{n_q-1}\right| & \text{if } e \in (B_j^p : B_l^q) \text{ for } p \neq q \\ & \qquad \text{and } (j,l) \neq (0,n_q),(n_p,0), \\ 0 & \text{if } e \in E(B_j^p) \text{ for all } j,p. \end{cases}$$

The right-hand side of the inequality is $\alpha := 1 + \sum_{p=1}^{P} \frac{n_p+1}{n_p-1}$. Under the conditions stated above, it is valid for the GRP polyhedron and polytope. If we allow $P = 0$, the definition includes the R-odd cut inequalities.

It can be shown that the PB-inequalities define facets of the 0/1-polytope under the following conditions (see [17]):

- $\left|(B_j^p : B_{j+1}^p)\right| \geq 2$ for all $j = 0, \ldots, n_p$ and all p, and
- every induced subgraph $G[B_j^p]$ is a single node or 8-edge connected.

A path-bridge inequality is called *n-regular* (or simply *regular*) if $n_p = n$ for $p = 1, \ldots, P$. It is called *simple* if $\left|B_j^p\right| = 1$ for all $j = 1, \ldots, n_p$, $p = 1, \ldots, P$.

We come to the description of *switched path-bridge inequalities.* Let P, n_p, A, Z, B_j^p be as above and $F \subseteq (A:Z)$, but with condition 2 replaced by

2a. $P + \mathbf{t}(A) + |F| = 1 \bmod 2$, and if A is a union of R-sets, then $P + |F| \geq 3$ must hold.

Let a be the vector of coefficients as defined in (5.1), but with the coefficients on edges $f \in F$ changed from 1 to -1. Then $ax \geq 1 - |F| + \sum_{p=1}^{P} \frac{n_p+1}{n_p-1}$ is a switched PB-inequality and hence valid for GRP(Γ). If we allow $P = 0$, then the definition includes the cocircuit inequalities (1.3) as a subclass. We note that the simplest form of switched PB-inequalities, namely, the switched 2-regular PBs, were found independently by Letchford.[2]

It can be shown that the switched PB-inequalities define facets of the polytope under the same conditions which are sufficient for the classical PB-inequalities to define facets (see above). We will give a stronger result for the facet-defining property of switched PB-inequalities based on a more elegant argument. We can assume that the set F does not include R-internal edges, because we can use Proposition 2.5 for these edges. The idea of the following theorem is to turn one edge $f \in F$ into a path. It can be used inductively.

Let P, n_p, A, Z, B_j^p, and F as just defined and let $ax \geq \alpha$ be the corresponding switched PB-inequality. Let $f \in F$ be an R-external edge.

Construct a graph G^f out of G in the following manner. Denote the end nodes of f by w_0 and w_3. Replace f by two nodes w_1, w_2 and six edges e_i, $i = 0, \ldots, 5$, where the end nodes of e_i are $w_{i/2}$ and $w_{i/2+1}$ (see Figure 5.1 for an illustration). Define a partition $\mathcal{C}^f$ of the node set of G^f by

$$\mathcal{C}^f := \mathcal{C} \cup \{\{w_1\},\{w_2\}\}$$

and a parity function by letting $\mathbf{t}^f(v) := \mathbf{t}(v)$ for all $v \in V(G)$ and $\mathbf{t}^f(w_i) := 0$ for $i = 1, 2$. Let $\Gamma^f := (G^f, \mathcal{C}^f, \mathbf{t}^f)$.

[2] Personal communication, 2001.

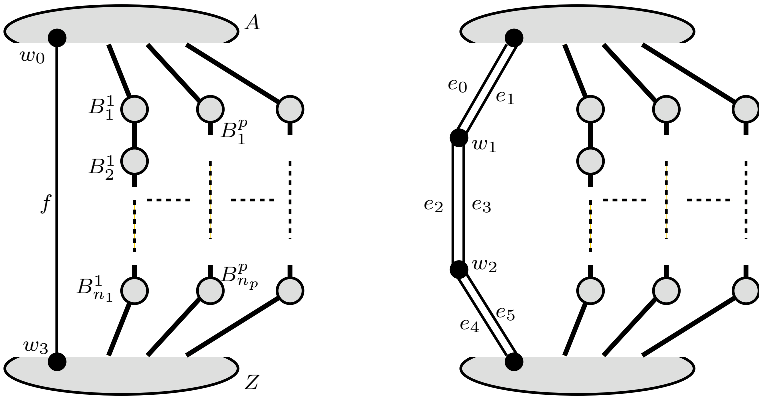


FIG. 5.1. *Illustration for Theorem* 5.1.

We modify the path-bridge configuration on $\varGamma$ to obtain a path-bridge configuration on $\varGamma^f$ by adding an extra path $P+1$:

$$B_0^{P+1} := A,\ \ B_1^{P+1} := \{w_1\},\ \ B_2^{P+1} := \{w_2\},\ \ Z =: B_3^{P+1}.$$

Let $a^f y \geq \alpha^f$ denote the modified switched PB-inequality.

THEOREM 5.1. *If* $\mathrm{GRP}(\varGamma)$ *is full-dimensional, then the original switched PB-inequality $ax \geq \alpha$ is facet-defining for* $\mathrm{GRP}(\varGamma)$ *if the modified switched PB-inequality $a^f y \geq \alpha^f$ defines a facet of* $\mathrm{GRP}(\varGamma^f)$.

It may be said that the conditions imposed in this theorem are fortunate, because for classical PB-inequalities to define facets, the condition that $\left|(B_j^p : B_{j+1}^p)\right| \geq 2$, for $j = 0, \dots, n_p$, is sufficient (see above).

The proof of Theorem 5.1 relates the faces induced by the two inequalities $ax \geq \alpha$ and $a^f y \geq \alpha^f$ geometrically. This relation uses the following mapping and is established in the lemma below. Define the affine mapping

$$h \colon \mathrm{R}^{E(G)\setminus\{f\}\cup\{e_0,\dots,e_5\}} \to \mathrm{R}^{E(G)}$$

by letting, for all $e \in E(G)$,

$$\big(h(y)\big)_e := \begin{cases} 4 - y(\{e_0,\dots,e_5\}) & \text{if } e = f, \\ y_e & \text{if } e \in E(G)\setminus\{f\}. \end{cases}$$

LEMMA 5.2. *The affine mapping h maps the face of* $\mathrm{GRP}(\varGamma^f)$ *induced by the modified switched path-bridge inequality $a^f y \geq \alpha^f$ onto the face of* $\mathrm{GRP}(\varGamma)$ *induced by the original switched path-bridge inequality $ax \geq \alpha$.*

Proof. First of all we show that for all $y \in \mathrm{R}^{E(G^f)}$, if $x := h(y)$, we have $\alpha - ax = \alpha^f - a^f y$. Note that for all edges in $e \in E(G)\setminus\{f\} = E(G^f)\setminus\{e_0,\dots,e_5\}$ the coefficients of the two inequalities are equal, $a_e = a_e^f$, and $x_e = y_e$ holds, too. For $i = 0,\dots,5$, we have $a_{e_i}^f = 1$. For the right-hand sides we have $\alpha^f = \alpha + 4$. From $x_f = 4 - y(\{e_0,\dots,e_5\})$ it follows that

$$\begin{aligned} \alpha - ax &= \alpha - \sum_{e\in E(G)\setminus\{f\}} a_e x_e + x_f \\ &= \alpha - \sum_{e\in E(G^f)\setminus\{e_0,\dots,e_5\}} a_e^f y_e + 4 - y(\{e_0,\dots,e_5\}) = \alpha^f - a^f y. \end{aligned}$$

Now we prove that, if $y \in \mathscr{S}(\Gamma^f)$ satisfies the modified inequality with equality, i.e., $a^f y = \alpha^f$, then $x := h(y) \in \mathscr{S}(\Gamma)$. We have the following implications:

$$\begin{aligned} y\left(\{e_0, \ldots, e_5\}\right) = 3 &\Rightarrow x_f = 1, \\ y\left(\{e_0, \ldots, e_5\}\right) = 4 &\Rightarrow x_f = 0. \end{aligned}$$

This would imply $x = h(y) \in \{0,1\}^{E(G)}$, if no other values of $y(\{e_0, \ldots, e_5\})$ occurred. But first $y(\{e_0, \ldots, e_5\}) \geq 3$ is a consequence of

$$y(\delta(w_1)) \geq 2, \qquad y(\delta(w_2)) \geq 2, \quad \text{and} \qquad y(\delta(\{w_1, w_2\})) \geq 2.$$

And second, if $y(\{e_0, \ldots, e_5\}) = 6$ (the value 5 is impossible for parity reasons), then $a^f y = \alpha$ cannot hold, since the "cheapest" way to connect the remaining sets B_j^p, $p \neq P+1$ would be to select exactly the two edges with the smallest coefficient out of $\delta(B_j^p)$. This is possible by taking exactly one edge in each of the sets $(B_j^p : B_{j+1}^p)$, $j = 0, \ldots, n_p$, which adds up to $\sum_{p \neq P+1} \frac{n_p+1}{n_p-1}$. If all edges $g \in F \setminus \{f\}$ lie in $x_g = 1$, then it follows that

$$a^f y \geq \sum_{p=1}^{P} \frac{n_p+1}{n_p-1} + 6 - |F \setminus \{f\}| \ > \ \sum_{p=1}^{P} \frac{n_p+1}{n_p-1} + 3 + 1 - |F \setminus \{f\}| = \alpha^f,$$

which proves that $y(\{e_0, \ldots, e_5\}) \in \{3,4\}$ and hence $x \in \{0,1\}^{E(G)}$.

We come to show that x satisfies the parity constraints. From $y(\{e_0, \ldots, e_5\}) = 3$ it follows that $y(\{e_{2k}, e_{2k+1}\}) = 1$ for all k, and $y(\{e_0, \ldots, e_5\}) = 4$ implies $y(\{e_{2k}, e_{2k+1}\}) \in \{0,2\}$ for all k. Consequently, the relation

$$y(\delta(v)) = x(\delta(v)) \mod 2$$

holds for all $v \in V(G)$.

To see that x is a semitour, the connectivity condition remains to be shown. Let S be a union of R-sets in $\mathcal{C}$. We have to show $x(\delta(S)) \geq 2$. If $f \notin \delta(S)$, then $x(\delta(S)) = y(\delta(S)) \geq 2$. Otherwise we can find $S' \subseteq V(G^f)$ with $\delta(S') \setminus \{e_0, \ldots, e_5\} = \delta(S) \setminus \{f\}$ and $\delta(S') \cap \{e_0, \ldots, e_5\} \neq \emptyset$. From $y(\delta(S')) \geq 2$ it then follows that $x(\delta(S)) \geq 2$.

Finally, we note that for every $x \in \mathscr{S}(\Gamma)$ there exists $y \in \mathscr{S}(\Gamma^f)$ with $h(y) = x$. Hence the restriction of h to the faces is surjective. This completes the proof of the lemma. □

Now we can tackle the proof of Theorem 5.1. Let P be the face of $\mathrm{GRP}(\Gamma)$ defined by $ax \geq \alpha$, and let P^f be the facet of $\mathrm{GRP}(\Gamma^f)$ defined by $a^f y \geq \alpha^f$. Since by Lemma 5.2 $h(P^f) = P$, and since the kernel of the matrix M which defines the affine mapping h equals 5, we have

$$\dim P = \dim h(P^f) \geq \dim P^f - 5 = |E(G) \setminus \{f\} \cup \{e_0, \ldots, e_5\}| - 1 - 5 = |E(G)| - 1.$$

Hence, $ax \geq \alpha$ defines a facet of $\mathrm{GRP}(\Gamma)$, if the polytope is full-dimensional.

To complete the proof, we give the argument which shows that $\mathrm{GRP}(\Gamma^f)$ has full dimension. If we contract the edges e_2, e_3, e_4, e_5 we get a GRP-structure Γ', which differs from Γ only in that the edge f is duplicated in G'. It is easy to verify that $\mathrm{GRP}(\Gamma')$ has full dimension, because this is the case for $\mathrm{GRP}(\Gamma)$. In [17], we show that a pair of parallel edges can be contracted without losing full-dimensionality of the polytope. Hence the full-dimensionality of $\mathrm{GRP}(\Gamma^f)$ follows from the full-dimensionality of $\mathrm{GRP}(\Gamma')$.

It may be worth mentioning the similarity of the construction of inserting three edges to the one used by Letchford [13] in a different context for the unbounded GRP polyhedron $\mathrm{conv}(\mathscr{S}^\infty)$.

6. Separation algorithms for switched path-bridge inequalities. A *separation algorithm* for a class of inequalities is a procedure which, given a point x, produces a violated inequality of this class or decides that none exists (see, e.g., [10]). In practice, heuristic separation routines are often used, which may succeed in finding a violated inequality, but they may fail even if there exist violated inequalities of the class.

The general case. In order to use the new facets in a cutting-plane algorithm to solve the GRP, a separation algorithm is needed. In this section we show how any separation algorithm for classical path-bridge inequalities can be used to separate the class of switched path-bridge inequalities with $F \subseteq E_{\text{int}}(G)$ (section 2).

We start with a lemma whose proof we omit because it is technical and not very insightful. Then we will describe the algorithmic consequence of this lemma.

LEMMA 6.1. *Suppose* $G, \mathcal{C}, \mathbf{t}$ *is given. Let* $x \in \mathrm{R}^E$*,* $x \geq 0$ *be a vector which satisfies all connectivity inequalities* (1.2b) *and violates a classical PB-inequality. Then the relation*

$$x(A : Z) < 1$$

holds, where A *and* Z *refer to the sets of the PB-configuration as defined in section* 5. □

Let there be given a GRP-structure Γ and a vector x which satisfies all connectivity inequalities. We derive a GRP-structure $\hat{\Gamma} = (\hat{G}, \hat{\mathcal{C}}, \hat{\mathbf{t}})$ and a vector $\hat{x}$, which satisfies all connectivity inequalities on $\hat{\Gamma}$, with the property that x violates a switched PB-inequality with $F \subseteq E_{\text{int}}(G)$ if and only if $\hat{x}$ violates a classical PB-inequality defined on $\hat{\Gamma}$. Thus we can use any separation algorithm or heuristic on $\hat{\Gamma}$ and $\hat{x}$ to produce a violated switched PB-inequality. The construction of $\hat{\Gamma}$ is inspired by [16].

For every $e \in E_{\text{int}}(G)$, we denote by $\psi^0(e), \psi^1(e) \in V(G)$ the two end nodes of e. We construct the graph $\hat{G}$ from G by *splitting* every edge $e \in E_{\text{int}}(G)$ and replacing it with a new node k_e and two edges $\{\psi^0(e), k_e\}$, $\{k_e, \psi^1(e)\}$. We let $\hat{x}_{\{\psi^0(e),k_e\}} = x_e$ and $\hat{x}_{\{k_e,\psi^1(e)\}} = 1 - x_e$. For $e \notin E_{\text{int}}(G)$, we define $\hat{x}_e = x_e$. The R-sets $\hat{\mathcal{C}}$ are constructed from $\mathcal{C}$ by inserting every split node k_e in the same set which contains the end nodes of e.

Next we assign parities to the edges of $\hat{G}$. We call the edge $\{k_e, \psi^1(e)\}$ *odd* for all $e \in E_{\text{int}}(G)$. All other edges are called *even.* For $v \in V(G)$ we let r_v denote the number of odd edges incident to v in $\hat{G}$, and we define the new parities by

$$\hat{\mathbf{t}}(v) := \begin{cases} \mathbf{t}(v) + r_v \mod 2 & \text{if } v \in V(G), \\ 1 & \text{if } v = k_e \text{ for an } e. \end{cases}$$

We have defined a GRP-structure $\hat{\Gamma}$, and $\hat{x}$ satisfies all connectivity constraints.

PROPOSITION 6.2. *A switched path-bridge inequality with* $F \subseteq E_{\text{int}}$ *on* Γ *is violated by* x *if and only if* $\hat{x}$ *violates a classical path-bridge inequality on* $\hat{\Gamma}$*.*

Proof. Let $\hat{A}$, $\hat{Z}$, $\hat{B}_j^p$ be a PB-configuration on $\hat{G}$ and $\hat{a}z \geq \hat{\alpha}$ be the corresponding PB-inequality which is violated by $\hat{x}$. We let $A := \hat{A} \cap V(G)$, $Z := \hat{Z} \cap V(G)$, and $B_j^p := \hat{B}_j^p \cap V(G)$. Further, we denote by F the set of edges of G such that the edge $\{k_e, \psi^1(e)\}$ of $\hat{G}$ is in $(\hat{A} : \hat{Z})$. In this way, a switched PB-configuration is defined, and the slack of the inequality equals $\hat{\alpha} - \hat{a}\hat{x}$.

On the other hand, if x violates a switched PB-inequality, a PB-inequality on $\hat{G}$ violated by $\hat{x}$ can be constructed in the same straightforward way. □

Two different separation heuristics for PB-inequalities are described in [2, 4]. A polynomial time exact separation routine for simple 2-regular PB-inequalities was introduced by Letchford in [12]. Hence, the class of the switched simple 2-regular PB-inequalities with $F \subseteq E_{\text{int}}$ can be separated in polynomial time. Letchford's separation routine can be modified to run in time $O(\hat{n}^2\hat{m}\log(\hat{n}^2/\hat{m}))$, where $\hat{n}$ is the number of nodes and $\hat{m}$ is the number of edges of $\hat{G}$. In the next section, we will show how the class of *all* switched simple 2-regular PB-inequalities, i.e., without restriction on the set F, can be separated into an even better worst case time.

Switched simple 2-regular path-bridge inequalities. We now propose a separation algorithm for switched simple 2-regular PB-inequalities which runs in time $O(n^2m\log(n^2/m))$. Let a *handle* $H \subseteq V(G)$, and *teeth* $T_1, \dots, T_P \subseteq V(G)$, $p \geq 0$, and a set of edges $F \subseteq \delta(H)$ be given. Assume that the following conditions hold:

1. $\mathbf{t}(H) + P + |F| = 1 \mod 2$.
2. $T_p = \{u, v\}$ for R-isolated nodes $u \in H$ and $v \in \overline{H}$, and $E(T_p) \neq \emptyset$.
3. The $P+1$ sets F, $E(T_p)$, $p = 1, \dots, P$ are pairwise disjoint.

The inequality

$$x(\delta(H) \setminus F) - x(F) + \sum_{p=1}^{P} x(\delta(T_p)) \geq 3P - |F| + 1 \tag{6.1}$$

is valid for $\mathrm{GRP}(\mathit{\Gamma})$. The inequalities of this type include the switched 2-regular PB-inequalities. By a technical but standard uncrossing argument, it is possible to obtain a violated switched 2-regular PB-inequality from a violated inequality (6.1).

For the separation of (6.1), we define a simple graph G^s by removing from G all but one edge in each set of parallel edges of G. This means that G^s has a node set $V(G^s) = V(G)$ and $\{u, v\} \in E(G^s)$ if and only if u and v are neighbors in G. Suppose that $u, v \in V(G)$ are adjacent. We define

$$\theta_{\{u,v\}} := \begin{cases} x(\delta(\{u,v\})) + x(u:v) - 3 & \text{if } u \text{ and } v \text{ are both R-isolated, and} \\ \infty & \text{otherwise.} \end{cases}$$

As pointed out in [12], $\theta \geq 0$ holds if all connectivity inequalities are satisfied. Now we let

$$x^0_{\{u,v\}} := \min_{\substack{F \subseteq (u:v) \\ |F| \text{ even}}} \Big(x\big((u:v) \setminus F\big) + |F| - x(F) \Big),$$

$$x^1_{\{u,v\}} := \min \Big[\theta_{\{u,v\}}, \quad \min_{\substack{F \subseteq (u:v) \\ |F| \text{ odd}}} \Big(x\big((u:v) \setminus F\big) + |F| - x(F) \Big) \Big].$$

As an immediate consequence of these definitions, we have the following proposition.

PROPOSITION 6.3. *There exists a violated switched simple* 2*-regular PB inequality if and only if there exists a cut* $\delta(W)$ *in* G^s *and a set of edges* $F \subseteq \delta(W)$ *such that* $x^0(\delta(W) \setminus F) + x^1(F) < 1$. □

We note that $x^0_{\{u,v\}} + x^1_{\{u,v\}} \geq 1$ for all $\{u, v\} \in E(G)$. We could use the blossom separation algorithm for capacitated matching problems of Padberg and Rao [16], which requires that we compute $O(|E(G^s)|)$ maximum flows on a graph with $O(|E(G)|)$ edges in the worst case. Using the algorithm described in [14], this problem can be solved in the time which is required to perform $|V(G^s)|$ max-flow computations on G^s.

Corollary 6.4. *Switched simple* 2*-regular PB-inequalities can be separated in time* $O(n^2 m \log(n^2/m))$, *where* $n := |V(G)|$ *and* $m := |E(G)|$.

Proof. This is the worst-case running time of performing n max-flow computations on a graph with n nodes and m edges if the well-known *pre-flow push* algorithm [9] is used to solve the max-flow problems. □

Of course, we would not expect to find many pairs u, v with more than one edge in $(u : v)$. This may be different if, prior to invoking the separation routine, shrinking operations have been performed on the graph, for example those described in [4] for the classical simple 2-regular PB-inequalities. Another possibility is to separate *switched simple* 2*-regular path-inequalities,* which are the special case of switched simple 2-regular PB-inequalities that occurs when both A and Z are unions of R-sets (compare [7, 6]).

Corollary 6.5. *Switched simple* 2*-regular path-inequalities can be separated in time* $O(n^2 m \log(n^2/m))$, *where* $n := |\mathcal{C}|$ *and* $m := |E(G_{\mathcal{C}})|$.

Proof. Use the above algorithm after shrinking each R-set into a single node. □

Acknowledgment. We thank the referees for their valuable comments which greatly improved the presentation of this paper.

REFERENCES

[1] F. Barahona and M. Grötschel, *On the cycle polytope of a binary matroid*, J. Combin. Theory Ser. B, 40 (1986), pp. 40–62.

[2] E. Benavent, A. Corberán, and J. M. Sanchis, *Linear programming based methods for solving arc routing problems*, in Arc Routing: Theory, Solutions, and Applications, M. Dror, ed., Kluwer Academic Publishers, Boston, 2000, pp. 231–275.

[3] N. Christofides, V. Campos, A. Corberán, and E. Mota, *An Algorithm for the Rural Postman Problem on a Graph*, Technical report, Imperial College, London, 1981.

[4] A. Corberán, A. N. Letchford, and J. M. Sanchis, *A cutting plane algorithm for the general routing problem*, Math. Program. Ser A, 90 (2001), pp. 291–316.

[5] A. Corberán and J. M. Sanchis, *A polyhedral approach to the rural postman problem*, Eur. J. Oper. Res., 79 (1994), pp. 95–114.

[6] A. Corberán and J. M. Sanchis, *The general routing problem polyhedron: Facets from the RPP and GTSP polyhedra.*, Eur. J. Oper. Res., 108 (1998), pp. 538–550.

[7] G. Cornuéjols, J. Fonlupt, and D. Naddef, *The traveling salesman problem on a graph and some related integer polyhedra*, Math. Program., 33 (1985), pp. 1–27.

[8] G. Ghiani and G. Laporte, *A branch-and-cut algorithm for the undirected rural postman problem*, Math. Program. Ser. A, 87 (2000), pp. 467–481.

[9] A. V. Goldberg and R. E. Tarjan, *A new approach to the maximum flow problem*, J. Association for Computing Machinery, 35 (1988), pp. 921–940.

[10] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Wiley, New York, 1988.

[11] J. K. Lenstra and A. H. G. Rinnooy Kan, *On general routing problems*, Networks, 6 (1976), pp. 273–280.

[12] A. N. Letchford, *New inequalities for the general routing problem*, Eur. J. Oper. Res., 96 (1997), pp. 317–322.

[13] A. N. Letchford, *The general routing polyhedron: A unifying framework*, Eur. J. Oper. Res., 112 (1999), pp. 122–133.

[14] A. N. Letchford, G. Reinelt, and D. O. Theis, *A faster exact separation algorithm for blossom inequalities*, in Integer Programming and Combinatorial Optimization IPCO, Lecture Notes Comput. Sci. 3064, D. Bienstock and G. Nemhauser, eds., Springer-Verlag, Berlin Heidelberg, 2004, pp. 196–205.

[15] C. Orloff, *A fundamental problem in vehicle routing*, Networks, 4 (1974), pp. 35–64.

[16] M. W. Padberg and M. R. Rao, *Odd minimum cut-sets and b-matchings*, Math. Oper. Res., 7 (1982), pp. 67–80.

[17] G. Reinelt and D. O. Theis, *On the general routing problem polytope*, A. Clark, R. Eglese, A. N. Letchford, and M. B. Wright, eds., special issue of Discrete Appl. Math. on Combinatorial Optimization (CO2004), to appear.

[18] G. Reinelt and D. O. Theis, *A note on the undirected rural postman problem polytope*, Math. Program., to appear.

[19] D. O. Theis, *Das General Routing Problem mit binären Variablen*, diploma thesis, University of Heidelberg, Heidelberg, Germany, 2001.

[20] G. M. Ziegler, *Lectures on Polytopes*, Graduate Texts in Mathematics, Vol. 152, Springer-Verlag, Berlin, 1998.

SIAM J. OPTIM.
Vol. 16, No. 1, pp. 235–249

EXCESSIVE GAP TECHNIQUE IN NONSMOOTH CONVEX MINIMIZATION[*]

YU. NESTEROV[†]

Abstract. In this paper we introduce a new primal-dual technique for convergence analysis of gradient schemes for nonsmooth convex optimization. As an example of its application, we derive a primal-dual gradient method for a special class of structured nonsmooth optimization problems, which ensures a rate of convergence of order $O(\frac{1}{k})$, where k is the iteration count. Another example is a gradient scheme, which minimizes a nonsmooth strongly convex function with known structure with rate of convergence $O(\frac{1}{k^2})$. In both cases the efficiency of the methods is higher than the corresponding black-box lower complexity bounds by an order of magnitude.

Key words. convex optimization, nonsmooth optimization, complexity theory, black-box oracle, optimal methods, structural optimization

AMS subject classifications. 90C25, 90C47, 68Q25

DOI. 10.1137/S1052623403422285

1. Introduction.

This paper continues the research started in [3], where it was shown that some structured nonsmooth optimization problems can be solved with efficiency estimates $O(\frac{1}{\epsilon})$, where ϵ is the desired accuracy of the solution. This complexity is much better than the theoretical lower complexity bound $O(\frac{1}{\epsilon^2})$ (see [2]). This improvement, of course, is possible because of a certain relaxation of the standard black-box assumption: it was assumed that our problem had an explicit and quite simple minimax structure. The numerical scheme proposed in [3] had a drawback, which decreases its practical efficiency: the number of steps must be fixed in advance, chosen in accordance with a worst-case complexity analysis.

In this paper we propose several new primal-dual gradient schemes for the same class of problems as those in [3]. However, our schemes now are free from the above deficiency. They are derived from a new primal-dual symmetric technique for convergence analysis, which we call the *excessive gap condition*.

The paper is organized as follows. In section 2 we introduce our *model* of optimization problem and recall several useful facts from [3]. In section 3 we describe the excessive gap condition. In sections 4 and 5 we present two different strategies for maintaining the condition during the optimization process. In section 6 we give the convergence result of order $O(\frac{1}{k})$, where k is the iteration counter. This convergence result is valid for all nonsmooth functions, described by our model. However, if we assume more (namely, the strong convexity of the primal objective), then the convergence can be improved up to $O(\frac{1}{k^2})$. This improvement is presented in section 7. Note that both complexity results improve the corresponding general lower complexity bound by an order of magnitude.

In what follows we work with different primal and dual spaces equipped by corresponding norms. For the sake of notation, we apply the following convention. The

[*]Received by the editors January 31, 2003; accepted for publication (in revised form) January 5, 2005; published electronically September 8, 2005. This paper presents research results of the Belgian Program on Interuniversity Poles of Attraction initiated by the Belgian State, Prime Minister's Office, Science Policy Programming. The scientific responsibility is assumed by the author.

http://www.siam.org/journals/siopt/16-1/42228.html

[†]Center for Operations Research and Econometrics (CORE), Catholic University of Louvain (UCL), 34 voie du Roman Pays, 1348 Louvain-la-Neuve, Belgium (nesterov@core.ucl.ac.be).

(primal) finite-dimensional real vector space is always denoted by E, possibly with an index. This space is endowed with a norm $\|\cdot\|$, which has the same index as the corresponding space. The space of linear functions on E is denoted by E^*. For $s \in E^*$ and $x \in E$ we denote by $\langle s, x\rangle$ the value of s at x. The *scalar product* $\langle\cdot,\cdot\rangle$ is marked by the same index as E. The norm for the dual space is defined in the standard way:

$$\|s\|^* = \max_{\|x\|=1} \langle s, x\rangle.$$

For operator $A : E_1 \to E_2^*$ we define the *adjoint* operator $A^* : E_2 \to E_1^*$ by identity

$$\langle Ax, u\rangle_2 \equiv \langle A^* u, x\rangle_1 \quad \forall x \in E_1,\ u \in E_2.$$

The *norm* of such an operator is defined as follows:

$$\|A\|_{1,2} = \max_{\|x\|_1=1} \max_{\|u\|_2=1} \langle Ax, u\rangle_2.$$

Clearly,

$$\|A\|_{1,2} = \|A^*\|_{2,1} = \max_{\|x\|_1=1} \|Ax\|_2^* = \max_{\|u\|_2=1} \|A^* u\|_1^*.$$

Hence, for any $h \in E_1$ we have

$$\|Ah\|_2^* \le \|A\|_{1,2} \cdot \|h\|_1. \tag{1.1}$$

Further, recall that function $d(x)$ is called *strongly convex* on a closed convex set Q if for any $\alpha \in [0,1]$ we have

$$d(\alpha x + (1-\alpha)y) \le \alpha d(x) + (1-\alpha)d(y) - \frac{1}{2}\alpha(1-\alpha)\sigma\|x-y\|^2, \quad x, y \in Q.$$

In this inequality, the constant σ is called the *(strong) convexity parameter* of d. We often use the following property of strongly convex functions:

$$d(x) \ge d(x_*) + \frac{1}{2}\sigma\|x - x_*\|^2, \quad x \in Q, \tag{1.2}$$

where $x_* = \arg\min_{x\in Q} d(x)$. If d is differentiable, the equivalent definitions of strong convexity are as follows (see, for example, [4, section 2.1.3]):

$$d(y) \ge d(x) + \langle \nabla d(x), y - x\rangle + \frac{1}{2}\sigma\|y - x\|_1^2, \quad x, y \in Q, \tag{1.3}$$

$$\langle \nabla d(x) - \nabla d(y), x - y\rangle \ge \sigma\|x - y\|^2, \quad x, y \in Q. \tag{1.4}$$

Finally, we say that function $f(x)$ has a Lipschitz-continuous gradient on a convex set Q if

$$\|\nabla f(x) - \nabla f(y)\|_* \le L\|x - y\|, \quad x, y \in Q,$$

for some constant $L \ge 0$. Then

$$f(y) \le f(x) + \langle \nabla f(x), y - x\rangle + \frac{1}{2}L\|y - x\|^2, \quad x, y \in Q \tag{1.5}$$

(see, for example, [4, section 2.1.1]).

2. A class of structured problems. In this paper we are interested in the minimization problem

$$\text{Find } f^* = \min_{x \in Q_1} f(x), \tag{2.1}$$

where Q_1 is a bounded closed convex set in a finite-dimensional real vector space E_1 and $f(x)$ is a continuous convex function on Q_1. We do not assume f to be differentiable.

Very often, the *structure* of the objective function in (2.1) is known. Let us assume that this structure can be described by the following *model* (see [3] for different examples):

$$f(x) = \hat{f}(x) + \max_{u \in Q_2} \{ \langle Ax, u \rangle_2 - \hat{\phi}(u) \}, \tag{2.2}$$

where function $\hat{f}(x)$ is continuous and convex on Q_1, Q_2 is a closed convex bounded set in a finite-dimensional real vector space E_2, $\hat{\phi}(u)$ is a continuous convex function on Q_2, and the linear operator A maps E_1 to E_2^*. In this case, problem (2.1) can be written in an *adjoint* form:

$$\begin{gathered} \max_{u \in Q_2} \phi(u), \\ \phi(u) = -\hat{\phi}(u) + \min_{x \in Q_1} \{ \langle Ax, u \rangle_2 + \hat{f}(x) \}. \end{gathered} \tag{2.3}$$

We assume that this representation is completely similar to (2.1) in the following sense. The methods described in this paper are implementable only if the optimization problems involved in the definitions of functions $f(x)$ and $\phi(u)$ can be solved in a closed form. So, we assume that the structures of the objects $\hat{f}$, $\hat{\phi}$, Q_1, and Q_2 are simple enough. We also assume that the functions $\hat{f}$ and $\hat{\phi}$ have Lipschitz-continuous gradients with Lipschitz constants $L_1(\hat{f})$ and $L_2(\hat{\phi})$, respectively.

Let us show that the knowledge of structure (2.2) can help in solving problems (2.1) and (2.3). As in [3], we are going to use this structure for constructing a smooth approximation of the objective functions.

Consider a *prox-function* $d_2(u)$ of the set Q_2. This means that $d_2(u)$ is continuous and strongly convex on Q_2 with a strong convexity parameter $\sigma_2 > 0$. Denote by

$$u_0 = \arg \min_{u \in Q_2} d_2(u)$$

the *prox-center* of the function $d_2(\cdot)$. Without loss of generality we assume that $d_2(u_0) = 0$. Thus, in view of (1.2), for any $u \in Q_2$ we have

$$d_2(u) \geq \frac{1}{2} \sigma_2 \| u - u_0 \|_2^2. \tag{2.4}$$

Let μ_2 be a positive *smoothness* parameter. Consider the following function:

$$f_{\mu_2}(x) = \hat{f}(x) + \max_{u \in Q_2} \{ \langle Ax, u \rangle_2 - \hat{\phi}(u) - \mu_2 d_2(u) \}. \tag{2.5}$$

Denote by $u_{\mu_2}(x)$ the optimal solution of above problem. Since function $d_2(u)$ is strongly convex, this solution is unique. In accordance with Danskin's theorem, the gradient of f_{μ_2} is well defined by

$$\nabla f_{\mu_2}(x) = \nabla \hat{f}(x) + A^* u_{\mu_2}(x). \tag{2.6}$$

Moreover, this gradient is Lipschitz-continuous with the constant

$$L_1(f_{\mu_2}) = L_1(\hat{f}) + \frac{1}{\sigma_2 \mu_2}\|A\|^2_{1,2} \tag{2.7}$$

(see, for example, Theorem 1 in [3]).

Similarly, let us consider a prox-function $d_1(x)$ of the set Q_1, which has convexity parameter σ_1, and the prox-center x_0 with $d_1(x_0) = 0$. By (1.2), for any $x \in Q_1$ we have

$$d_1(x) \geq \frac{1}{2}\sigma_1 \|x - x_0\|^2_1. \tag{2.8}$$

Let μ_1 be a positive smoothness parameter. Consider

$$\phi_{\mu_1}(u) = -\hat{\phi}(u) + \min_{x \in Q_1}\{\langle Ax, u\rangle_2 + \hat{f}(x) + \mu_1 d_1(x)\}. \tag{2.9}$$

Since the second term in the above definition is a minimum of linear functions, $\phi_{\mu_1}(u)$ is concave. Denote by $x_{\mu_1}(u)$ the unique optimal solution of the above problem. In accordance with Danskins's theorem and Theorem 1 in [3], the gradient

$$\nabla\phi_{\mu_1}(u) = -\nabla\hat{\phi}(u) + Ax_{\mu_1}(u) \tag{2.10}$$

is Lipschitz-continuous with the constant

$$L_2(\phi_{\mu_1}) = L_2(\hat{\phi}) + \frac{1}{\sigma_1 \mu_1}\|A\|^2_{1,2}. \tag{2.11}$$

3. Excessive gap condition. Note that for any $x \in Q_1$ and $u \in Q_2$ we have

$$\phi(u) \leq f(x), \tag{3.1}$$

and our assumptions guarantee no duality gap for (2.1), (2.3). However, $f_{\mu_2}(x) \leq f(x)$ and $\phi(u) \leq \phi_{\mu_1}(u)$. That opens up a possibility to satisfy the following *excessive gap condition*:

$$f_{\mu_2}(\bar{x}) \leq \phi_{\mu_1}(\bar{u}) \tag{3.2}$$

by certain $\bar{x} \in Q_1$ and $\bar{u} \in Q_2$. It is clear that (3.2) provides us with an upper bound on the quality of the primal-dual pair $(\bar{x}, \bar{u})$.

LEMMA 3.1. *Let $\bar{x} \in Q_1$ and $\bar{u} \in Q_2$ satisfy* (3.2). *Then*

$$0 \leq \max\{f(\bar{x}) - f^*, f^* - \phi(\bar{u})\} \leq f(\bar{x}) - \phi(\bar{u}) \leq \mu_1 D_1 + \mu_2 D_2, \tag{3.3}$$

where $D_1 = \max_{x \in Q_1} \ d_1(x)$ and $D_2 = \max_{u \in Q_2} \ d_2(u)$.

Proof. Indeed, for any $\bar{x} \in Q_1$, $\bar{u} \in Q_2$ we have $f(\bar{x}) - \mu_2 D_2 \leq f_{\mu_2}(\bar{x}) \leq \phi_{\mu_1}(\bar{u}) \leq \phi(\bar{u}) + \mu_1 D_1$. It remains to apply (3.1). □

Our goal is to justify a process for updating recursively the pair $(\bar{x}, \bar{u})$, which keeps satisfying inequality (3.2) as μ_1 and μ_2 go to zero. This can be done in two different ways, which correspond to two different auxiliary problems we must be ready to solve at each iteration. Before we start our analysis, let us prove one useful inequality.

LEMMA 3.2. *For any x and $\bar{y}$ from Q_1 we have*

$$f_{\mu_2}(\bar{y}) + \langle\nabla f_{\mu_2}(\bar{y}), x - \bar{y}\rangle_1 \leq \hat{f}(x) + \langle Ax, u_{\mu_2}(\bar{y})\rangle_2 - \hat{\phi}(u_{\mu_2}(\bar{y})). \tag{3.4}$$

Proof. Let us take arbitrary x and $\bar{y}$ from Q_1. Denote $\bar{u} = u_{\mu_2}(\bar{y})$. Then

$$\begin{aligned}
& f_{\mu_2}(\bar{y}) + \langle\nabla f_{\mu_2}(\bar{y}), x - \bar{y}\rangle_1 \\
\text{(by (2.5), (2.6))} \ =\ & \hat{f}(\bar{y}) + \langle A\bar{y}, \bar{u}\rangle_2 - \hat{\phi}(\bar{u}) - \mu_2 d_2(\bar{u}) + \langle\nabla\hat{f}(\bar{y}) + A^*\bar{u}, x - \bar{y}\rangle_1 \\
\text{(by convexity of } \hat{f}) \ \leq\ & \hat{f}(x) + \langle Ax, \bar{u}\rangle_2 - \hat{\phi}(\bar{u}). \qquad \square
\end{aligned}$$

4. Gradient mapping. Let us justify a possibility to satisfy the excessive gap condition (3.2) at some starting primal-dual pair. For $x \in Q_1$ define the *primal gradient mapping*:

$$T_{\mu_2}(x) = \arg\min_{y \in Q_1} \left\{ \langle \nabla f_{\mu_2}(x), y - x \rangle_1 + \frac{1}{2} L_1(f_{\mu_2}) \|y - x\|_1^2 \right\}. \tag{4.1}$$

LEMMA 4.1. *Let us choose an arbitrary $\mu_2 > 0$. For prox-center x_0 define*

$$\bar{x} = T_{\mu_2}(x_0), \quad \bar{u} = u_{\mu_2}(x_0). \tag{4.2}$$

Then the excessive gap condition (3.2) *is satisfied for any*

$$\mu_1 \geq \frac{1}{\sigma_1} L_1(f_{\mu_2}). \tag{4.3}$$

Proof. Denote $\bar{x} = T_{\mu_2}(x_0)$, $L_1 = L_1(f_{\mu_2})$, and $\bar{u} = u_{\mu_2}(x_0)$. Since the gradient ∇f_{μ_2} is Lipschitz-continuous, by (1.5) we have

$$\begin{aligned} f_{\mu_2}(\bar{x}) &\leq f_{\mu_2}(x_0) + \langle \nabla f_{\mu_2}(x_0), \bar{x} - x_0 \rangle_1 + \frac{1}{2} L_1 \|\bar{x} - x_0\|_1^2 \\ \text{(by (4.1))} &= \min_{x \in Q_1} \left\{ f_{\mu_2}(x_0) + \langle \nabla f_{\mu_2}(x_0), x - x_0 \rangle_1 + \frac{1}{2} L_1 \|x - x_0\|_1^2 \right\} \\ \text{(by (3.4), (4.3))} &\leq \min_{x \in Q_1} \left\{ \hat{f}(x) + \langle Ax, \bar{u} \rangle_2 - \hat{\phi}(\bar{u}) + \frac{1}{2} \mu_1 \sigma_1 \|x - x_0\|_1^2 \right\} \\ \text{(by (2.8))} &\leq -\hat{\phi}(\bar{u}) + \min_{x \in Q_1} \{ \hat{f}(x) + \langle Ax, \bar{u} \rangle_2 + \mu_1 d_1(x) \} \; = \; \phi_{\mu_1}(\bar{u}). \qquad \square \end{aligned}$$

Thus, condition (3.2) can be satisfied for some primal-dual pair. Let us show how we can update points $\bar{x}$ and $\bar{u}$ in order to keep (3.2) valid for smaller values of μ_1 and μ_2. Note that in view of the symmetry of the situation, at the first step of the process we can try to decrease only μ_1, keeping μ_2 unchanged. After that, at the second step, we update μ_2 and keep μ_1, and so on. The main advantage of such a switching strategy is that we need to find a justification only for the first step. The proof for the second one will be symmetric.

THEOREM 4.2. *Let points $\bar{x} \in Q_1$ and $\bar{u} \in Q_2$ satisfy the excessive gap condition* (3.2) *for some positive μ_1 and μ_2. Let us fix $\tau \in (0,1)$ and choose $\mu_1^+ = (1 - \tau)\mu_1$,*

$$\begin{aligned} \hat{x} &= (1 - \tau)\bar{x} + \tau x_{\mu_1}(\bar{u}), \\ \bar{u}_+ &= (1 - \tau)\bar{u} + \tau u_{\mu_2}(\hat{x}), \\ \bar{x}_+ &= T_{\mu_2}(\hat{x}). \end{aligned} \tag{4.4}$$

Then the pair $(\bar{x}_+, \bar{u}_+)$ satisfies condition (3.2) *with smoothness parameters μ_1^+ and μ_2, provided that τ is chosen in accordance with the following relation:*

$$\frac{\tau^2}{1 - \tau} \leq \frac{\mu_1 \sigma_1}{L_1(f_{\mu_2})}. \tag{4.5}$$

Proof. Denote $\hat{u} = u_{\mu_2}(\hat{x})$ and $x_1 = x_{\mu_1}(\bar{u})$. Since $\hat{\phi}$ is convex, in view of the second line in (4.4) we have $\hat{\phi}(\bar{u}_+) \le (1-\tau)\hat{\phi}(\bar{u}) + \tau\hat{\phi}(\hat{u})$. Therefore

$$\begin{aligned}
\phi_{\mu_1^+}(\bar{u}_+) &= \min_{x\in Q_1} \Big\{(1-\tau)\mu_1 d_1(x) + \langle Ax, (1-\tau)\bar{u} + \tau\hat{u}\rangle_2 + \hat{f}(x)\Big\} - \hat{\phi}(\bar{u}_+)\\
&\ge \min_{x\in Q_1} \Big\{(1-\tau)\Big[\mu_1 d_1(x) + \langle Ax, \bar{u}\rangle_2 + \hat{f}(x) - \hat{\phi}(\bar{u})\Big]_1\\
&\quad + \tau\Big[\hat{f}(x) + \langle Ax, \hat{u}\rangle_2 - \hat{\phi}(\hat{u})\Big]_2\Big\}.
\end{aligned}$$

Note that in view of condition (3.2) and the first line in (4.4) we have

$$\phi_{\mu_1}(\bar{u}) \ge f_{\mu_2}(\bar{x}) \ge f_{\mu_2}(\hat{x}) + \langle \nabla f_{\mu_2}(\hat{x}), \bar{x} - \hat{x}\rangle_1 = f_{\mu_2}(\hat{x}) + \tau\langle \nabla f_{\mu_2}(\hat{x}), \bar{x} - x_1\rangle_1.$$

Therefore, in view of property (1.2) and definition (2.9) we can estimate the expression in the first brackets as follows:

$$\begin{aligned}
[\,\cdot\,]_1 &\ge \phi_{\mu_1}(\bar{u}) + \frac{1}{2}\mu_1\sigma_1\|x - x_1\|_1^2\\
(\text{by } (3.2)) &\ge f_{\mu_2}(\bar{x}) + \frac{1}{2}\mu_1\sigma_1\|x - x_1\|_1^2\\
(f \text{ is convex}) &\ge f_{\mu_2}(\hat{x}) + \langle \nabla f_{\mu_2}(\hat{x}), \bar{x} - \hat{x}\rangle_1 + \frac{1}{2}\mu_1\sigma_1\|x - x_1\|_1^2\\
(\text{line 1, } (4.4)) &= f_{\mu_2}(\hat{x}) + \tau\langle \nabla f_{\mu_2}(\hat{x}), \bar{x} - x_1\rangle_1 + \frac{1}{2}\mu_1\sigma_1\|x - x_1\|_1^2.
\end{aligned}$$

In view of (3.4), for the second pair of brackets we have

$$\begin{aligned}
[\,\cdot\,]_2 &\ge f_{\mu_2}(\hat{x}) + \langle \nabla f_{\mu_2}(\hat{x}), x - \hat{x}\rangle_1\\
(\text{line 1, } (4.4)) &= f_{\mu_2}(\hat{x}) + \langle \nabla f_{\mu_2}(\hat{x}), x - x_1 + (1-\tau)(x_1 - \bar{x})\rangle_1.
\end{aligned}$$

Thus, putting everything together, we complete the proof as follows:

$$\begin{aligned}
\phi_{\mu_1^+}(\bar{u}_+) &\ge \min_{x\in Q_1} \Big\{f_{\mu_2}(\hat{x}) + \tau\langle \nabla f_{\mu_2}(\hat{x}), x - x_1\rangle_1 + \frac{1}{2}(1-\tau)\mu_1\sigma_1\|x - x_1\|_1^2\Big\}\\
(\text{by } (4.5)) &\ge \min_{x\in Q_1} \Big\{f_{\mu_2}(\hat{x}) + \tau\langle \nabla f_{\mu_2}(\hat{x}), x - x_1\rangle_1 + \frac{1}{2}\tau^2 L_1(f_{\mu_2})\|x - x_1\|_1^2\Big\}\\
(y = \bar{x} + \tau(x - \bar{x})) &= \min_{y\in \bar{x}+\tau(Q_1-\bar{x})} \Big\{f_{\mu_2}(\hat{x}) + \langle \nabla f_{\mu_2}(\hat{x}), y - \hat{x}\rangle_1 + \frac{1}{2}L_1(f_{\mu_2})\|y - \hat{x}\|_1^2\Big\}\\
(Q_1 \text{ is convex}) &\ge \min_{y\in Q_1} \Big\{f_{\mu_2}(\hat{x}) + \langle \nabla f_{\mu_2}(\hat{x}), y - \hat{x}\rangle_1 + \frac{1}{2}L_1(f_{\mu_2})\|y - \hat{x}\|_1^2\Big\}\\
(\text{line 3, } (4.4)) &= f_{\mu_2}(\hat{x}) + \langle \nabla f_{\mu_2}(\hat{x}), \bar{x}_+ - \hat{x}\rangle_1 + \frac{1}{2}L_1(f_{\mu_2})\|\bar{x}_+ - \hat{x}\|_1^2\\
(\text{by } (1.5)) &\ge f_{\mu_2}(\bar{x}_+). \qquad \square
\end{aligned}$$

5. Bregman projection. Let us assume for simplicity that $d_1(x)$ is differentiable. Then for any $x \in Q_1$ we have

$$\langle \nabla d_1(x_0), x - x_0\rangle_1 \ge 0. \tag{5.1}$$

For x and z from Q_1 denote the *Bregman distance* between z and x as

$$\xi_1(z,x) = d_1(x) - d_1(z) - \langle \nabla d_1(z), x - z\rangle_1.$$

If z is fixed, then $\xi(z,x)$ is strongly convex in x. Moreover, in view of (1.3)

$$\xi_1(z,x) \geq \frac{1}{2}\sigma_1 \|x - z\|_1^2. \tag{5.2}$$

Define the *Bregman projection* of some $g \in E_1^*$ onto the set Q_1 as follows:

$$V_1(z,g) = \arg\min_{x\in Q_1} \{\langle g, x - z\rangle_1 + \xi_1(z,x)\}. \tag{5.3}$$

As compared with the gradient mapping, the Bregman projection has several advantages. First, it is uniquely defined. Second, the optimization problem in (5.3) involves the same objects as (2.9). So, there are more chances for it to be easily solvable (see section 5.3 in [3] for an example).

Let us show that the Bregman projection can also be used for finding a primal-dual pair, which satisfies the excessive gap condition (3.2).

LEMMA 5.1. *Let us choose an arbitrary $\mu_2 > 0$. Denote $\gamma = \frac{\sigma_1}{L_1(f_{\mu_2})}$ and set*

$$\bar{x} = V_1(x_0, \gamma \nabla f_{\mu_2}(x_0)), \quad \bar{u} = u_{\mu_2}(x_0). \tag{5.4}$$

Then the excessive gap condition is satisfied for any $\mu_1 \geq \gamma^{-1}$.

Proof. Indeed, in view of (1.5) we have

$$\begin{aligned}
f_{\mu_2}(\bar{x}) &\leq f_{\mu_2}(x_0) + \langle \nabla f_{\mu_2}(x_0), \bar{x} - x_0\rangle_1 + \frac{1}{2}L_1(f_{\mu_2})\|\bar{x} - x_0\|_1^2\\
&= f_{\mu_2}(x_0) + \frac{1}{\gamma}\left[\gamma\langle \nabla f_{\mu_2}(x_0), \bar{x} - x_0\rangle_1 + \frac{1}{2}\sigma_1\|\bar{x} - x_0\|_1^2\right]\\
\text{(by (5.2))} &\leq f_{\mu_2}(x_0) + \frac{1}{\gamma}\left[\langle \gamma\nabla f_{\mu_2}(x_0), \bar{x} - x_0\rangle_1 + \xi_1(x_0,\bar{x})\right]\\
\text{(by (5.3), (5.4))} &= f_{\mu_2}(x_0) + \frac{1}{\gamma}\min_{x\in Q_1}\left\{\langle \gamma\nabla f_{\mu_2}(x_0), x - x_0\rangle_1 + \xi_1(x_0,x)\right\}\\
&= \min_{x\in Q_1}\left\{f_{\mu_2}(x_0) + \langle \nabla f_{\mu_2}(x_0), x - x_0\rangle_1 + \frac{1}{\gamma}\xi_1(x_0,x)\right\}\\
\text{(by (5.1))} &\leq \min_{x\in Q_1}\left\{f_{\mu_2}(x_0) + \langle \nabla f_{\mu_2}(x_0), x - x_0\rangle_1 + \frac{1}{\gamma}d_1(x)\right\}\\
\text{(using (3.4))} &\leq \min_{x\in Q_1}\left\{\hat{f}(x) + \langle Ax, u_{\mu_2}(x_0)\rangle - \hat{\phi}(u_{\mu_2}(x_0)) + \frac{1}{\gamma}d_1(x)\right\}\\
\text{(by (2.9))} &= \phi_{\gamma^{-1}}(u_{\mu_2}(x_0)) \leq \phi_{\mu_1}(u_{\mu_2}(x_0)). \qquad \square
\end{aligned}$$

As in section 4, we present a justification only for the first (primal) step of the switching primal-dual strategy for maintaining the excessive gap condition (3.2) while the parameters μ_1 and μ_2 go to zero.

THEOREM 5.2. *Let points $\bar{x} \in Q_1$ and $\bar{u} \in Q_2$ satisfy the excessive gap condition* (3.2) *for some positive μ_1 and μ_2. Let us choose $\tau \in (0,1)$ in accordance with* (4.5) *and set*

$$\begin{aligned} \hat{x} &= (1-\tau)\bar{x} + \tau x_{\mu_1}(\bar{u}), \\ \bar{u}_+ &= (1-\tau)\bar{u} + \tau u_{\mu_2}(\hat{x}), \\ \tilde{x} &= V_1\left(x_{\mu_1}(\bar{u}), \frac{\tau}{(1-\tau)\mu_1} \nabla f_{\mu_2}(\hat{x}) \right), \\ \bar{x}_+ &= (1-\tau)\bar{x} + \tau\tilde{x}, \\ \mu_1^+ &= (1-\tau)\mu_1. \end{aligned} \tag{5.5}$$

Then the pair $(\bar{x}_+, \bar{u}_+)$ *satisfies* (3.2) *with the smoothness parameters* μ_1^+ *and* μ_2.

Proof. Denote $\hat{u} = u_{\mu_2}(\hat{x})$ and $x_1 = x_{\mu_1}(\bar{u})$. In view of the rules (5.5), convexity of $\hat{\phi}$, and inequality (3.4), we have

$$\begin{aligned} & (1-\tau)\mu_1 d_1(x) + \langle Ax, (1-\tau)\bar{u} + \tau\hat{u}\rangle_2 + \hat{f}(x) - \hat{\phi}(\bar{u}_+) \\ & \geq (1-\tau)\Big[\mu_1 d_1(x) + \langle Ax, \bar{u}\rangle_2 + \hat{f}(x) - \hat{\phi}(\bar{u})\Big] + \tau[\hat{f}(x) + \langle Ax, \hat{u}\rangle_2 - \hat{\phi}(\hat{u})] \\ & \geq (1-\tau)\Big[\mu_1 d_1(x) + \langle Ax, \bar{u}\rangle_2 + \hat{f}(x) - \hat{\phi}(\bar{u})\Big]_1 + \tau[f_{\mu_2}(\hat{x}) + \langle \nabla f_{\mu_2}(\hat{x}), x - \hat{x}\rangle_1]_2. \end{aligned}$$

The first order optimality conditions for point x_1 are as follows:

$$\langle \mu_1 \nabla d_1(x_1) + A^*\bar{u} + \nabla\hat{f}(x_1), x - x_1\rangle_1 \geq 0, \quad x \in Q_1. \tag{5.6}$$

Therefore, using convexity of $\hat{f}$ and f_{μ_2}, we can estimate the term $[\,\cdot\,]_1$ as follows:

$$\begin{aligned} [\,\cdot\,]_1 &= \mu_1\left(\xi(x_1,x) + d_1(x_1) + \langle \nabla d_1(x_1), x - x_1\rangle_1\right) + \langle Ax, \bar{u}\rangle_2 + \hat{f}(x) - \hat{\phi}(\bar{u}) \\ \text{(by (5.6))} &\geq \mu_1\xi(x_1,x) + \mu_1 d_1(x_1) + \langle Ax_1, \bar{u}\rangle_2 + \hat{f}(x) - \langle \nabla\hat{f}(x_1), x - x_1\rangle_1 - \hat{\phi}(\bar{u}) \\ &\geq \mu_1\xi(x_1,x) + \mu_1 d_1(x_1) + \langle Ax_1, \bar{u}\rangle_2 + \hat{f}(x_1) - \hat{\phi}(\bar{u}) \\ \text{(by (2.9))} &= \mu_1\xi(x_1,x) + \phi_{\mu_1}(\bar{u}) \\ \text{(by (3.2))} &\geq \mu_1\xi(x_1,x) + f_{\mu_2}(\bar{x}) \\ &\geq \mu_1\xi(x_1,x) + f_{\mu_2}(\hat{x}) + \langle \nabla f_{\mu_2}(\hat{x}), \bar{x} - \hat{x}\rangle_1. \end{aligned}$$

Thus, we can continue:

$$\begin{aligned} \phi_{\mu_1^+}(\bar{u}_+) &= \min_{x\in Q_1}\Big\{(1-\tau)\mu_1 d_1(x) + \langle Ax, (1-\tau)\bar{u} + \tau\hat{u}\rangle_2 + \hat{f}(x)\Big\} - \hat{\phi}(\bar{u}_+) \\ &\geq \min_{x\in Q_1}\{(1-\tau)\mu_1\xi(x_1,x) + f_{\mu_2}(\hat{x}) + \langle \nabla f_{\mu_2}(\hat{x}), (1-\tau)\bar{x} + \tau x - \hat{x}\rangle_1\} \\ \text{(line 1, (5.5))} &= \min_{x\in Q_1}\{(1-\tau)\mu_1\xi(x_1,x) + f_{\mu_2}(\hat{x}) + \tau\langle \nabla f_{\mu_2}(\hat{x}), x - x_1\rangle_1\} \\ \text{(line 3, (5.5))} &= (1-\tau)\mu_1\xi(x_1,\tilde{x}) + f_{\mu_2}(\hat{x}) + \tau\langle \nabla f_{\mu_2}(\hat{x}), \tilde{x} - x_1\rangle_1 \\ \text{(by (5.2))} &\geq \frac{1}{2}(1-\tau)\mu_1\sigma_1\|\tilde{x} - x_1\|_1^2 + f_{\mu_2}(\hat{x}) + \tau\langle \nabla f_{\mu_2}(\hat{x}), \tilde{x} - x_1\rangle_1 \\ \text{(by (4.5))} &\geq \frac{1}{2}\tau^2 L_1(f_{\mu_2})\|\tilde{x} - x_1\|_1^2 + f_{\mu_2}(\hat{x}) + \tau\langle \nabla f_{\mu_2}(\hat{x}), \tilde{x} - x_1\rangle_1 \\ \text{(line 4, (5.5))} &= \frac{1}{2}L_1(f_{\mu_2})\|\bar{x}_+ - \hat{x}\|_1^2 + f_{\mu_2}(\hat{x}) + \langle \nabla f_{\mu_2}(\hat{x}), \bar{x}_+ - \hat{x}\rangle_1 \\ \text{(by (1.5))} &\geq f_{\mu_2}(\bar{x}_+). \qquad \square \end{aligned}$$

6. Convergence analysis. In sections 4 and 5 we have seen that the smoothness parameters μ_1 and μ_2 can be decreased by a switching strategy. Thus, in order to convert the results of Theorems 4.2 and 5.2 in an algorithmic scheme we only need to point out a strategy for updating these parameters, which is compatible with the condition (4.5). In this section we do that for an important case, $L_1(\hat{f}) = L_2(\hat{\phi}) = 0$.

It is convenient to represent the smoothness parameters as follows:

$$\mu_1 = \lambda_1 \cdot \|A\|_{1,2} \cdot \sqrt{\frac{D_2}{\sigma_1\sigma_2 D_1}}, \quad \mu_2 = \lambda_2 \cdot \|A\|_{1,2} \cdot \sqrt{\frac{D_1}{\sigma_1\sigma_2 D_2}}. \tag{6.1}$$

Then the estimate (3.3) for the duality gap becomes symmetric:

$$f(\bar{x}) - \phi(\bar{u}) \le (\lambda_1 + \lambda_2) \cdot \|A\|_{1,2} \cdot \sqrt{\frac{D_1 D_2}{\sigma_1\sigma_2}}. \tag{6.2}$$

Since by (2.7), $L_1(f_{\mu_2}) = \frac{1}{\sigma_2\mu_2}\|A\|^2_{1,2}$, the condition (4.5) becomes problem independent:

$$\frac{\tau^2}{1-\tau} \le \mu_1\mu_2 \cdot \frac{\sigma_1\sigma_2}{\|A\|^2_{1,2}} = \lambda_1\lambda_2. \tag{6.3}$$

Let us write down the switching algorithmic scheme in an explicit form. It is convenient to have a permanent iteration counter. In this case at even iterations we apply the primal update (4.4) (or (5.5)), and at odd iterations we apply the corresponding dual update. Since at even iterations λ_2 is not changing, and at odd iterations λ_1 is not changing, it is convenient to put their new values in the same sequence $\{\alpha_k\}_{k=-1}^{\infty}$. Let us fix the following relations between the sequences:

$$\begin{array}{rcl} k = & 2l & : \quad \lambda_{1,k} = \alpha_{k-1}, \quad \lambda_{2,k} = \alpha_k, \\ k = & 2l+1 & : \quad \lambda_{1,k} = \alpha_k, \quad\quad \lambda_{2,k} = \alpha_{k-1}. \end{array} \tag{6.4}$$

Then the parameters τ_k define the reduction rate of the sequence $\{\alpha_k\}_{k=-1}^{\infty}$.

LEMMA 6.1. *For all $k \ge 0$ we have $\alpha_{k+1} = (1-\tau_k)\alpha_{k-1}$.*

Proof. Indeed, in accordance with (6.4), if $k = 2l$, then

$$\alpha_{k+1} = \lambda_{1,k+1} = (1-\tau_k)\lambda_{1,k} = (1-\tau_k)\alpha_{k-1}.$$

Also, if $k = 2l+1$, then $\alpha_{k+1} = \lambda_{2,k+1} = (1-\tau_k)\lambda_{2,k} = (1-\tau_k)\alpha_{k-1}$. □

COROLLARY 6.2. *In terms of the sequence $\{\alpha_k\}_{k=-1}^{\infty}$ the condition* (6.3) *looks as follows:*

$$(\alpha_{k+1} - \alpha_{k-1})^2 \le \alpha_{k+1}\alpha_k\alpha_{k-1}^2, \quad k \ge 0. \tag{6.5}$$

Proof. In view of (6.4) we always have $\lambda_{1,k}\lambda_{2,k} = \alpha_k\alpha_{k-1}$. Since $\tau_k = 1 - \frac{\alpha_{k+1}}{\alpha_{k-1}}$, we get (6.5). □

Clearly, condition (6.5) is satisfied by

$$\alpha_k = \frac{2}{k+2}, \quad k \ge -1. \tag{6.6}$$

Then

$$\tau_k = 1 - \frac{\alpha_{k+1}}{\alpha_{k-1}} = \frac{2}{k+3}, \quad k \ge 0. \tag{6.7}$$

Now we are ready to write down the algorithmic scheme. Let us do that for the gradient mapping update (4.4). In this scheme we use the sequences $\{\mu_{1,k}\}_{k=-1}^{\infty}$ and $\{\mu_{2,k}\}_{k=-1}^{\infty}$, generated in accordance with rules (6.1), (6.4), and (6.6).

Method 1.

1. **Initialization:**
   Choose $\bar{x}_0$ and $\bar{u}_0$ in accordance with (4.2) with $\mu_1 = \mu_{1,0}$ and $\mu_2 = \mu_{2,0}$.
2. **Iterations ($k \geq 0$):**
   a) Set $\tau_k = \frac{2}{k+3}$.
   b) If k is even, then generate $(\bar{x}_{k+1}, \bar{u}_{k+1})$ from $(\bar{x}_k, \bar{u}_k)$ using (4.4).
   c) If k is odd, then generate $(\bar{x}_{k+1}, \bar{u}_{k+1})$ from $(\bar{x}_k, \bar{u}_k)$ using the symmetric dual variant of (4.4).

Theorem 6.3. *Let the sequences $\{\bar{x}_k\}_{k=0}^{\infty}$ and $\{\bar{u}_k\}_{k=0}^{\infty}$ be generated by Method* 1. *Then each pair of points satisfies the excessive gap condition. Therefore*

$$f(\bar{x}_k) - \phi(\bar{u}_k) \leq \frac{4\|A\|_{1,2}}{k+1}\sqrt{\frac{D_1 D_2}{\sigma_1\sigma_2}}. \tag{6.8}$$

Proof. In accordance with our choice of parameters,

$$\mu_{1,0}\mu_{2,0} = \lambda_{1,0}\lambda_{2,0} \cdot \frac{\|A\|_{1,2}^2}{\sigma_1\sigma_2} = \frac{2\mu_{2,0}}{\sigma_1} L_1(f_{\mu_{2,0}}) > \frac{\mu_{2,0}}{\sigma_1} L_1(f_{\mu_{2,0}}).$$

Hence, in view of Lemma 4.1 the pair $(\bar{x}_0, \bar{u}_0)$ satisfies the excessive gap condition. We have already checked that the sequence $\{\tau_k\}_{k=0}^{\infty}$ defined by (6.7) satisfies the conditions of Theorem 4.2. Therefore the excessive gap conditions will be valid for the sequences generated by Method 1. It remains to use inequality (6.2). ∎

Clearly, the same statement is valid for the method based on the updating scheme (5.5).

7. Minimizing a strongly convex function. Consider now the model (2.2), which satisfies the following assumption.

Assumption 1. In representation (2.2) function $\hat{f}(x)$ is strongly convex with strong convexity parameter $\hat{\sigma} > 0$.

Let us prove the following variant of Danskin's theorem.

Lemma 7.1. *Under Assumption* 1, *function $\phi(u)$ defined by* (2.3) *is concave and differentiable. Moreover, its gradient*

$$\nabla\phi(u) = -\nabla\hat{\phi}(u) + Ax_0(u), \tag{7.1}$$

with $x_0(u)$ defined by (2.9), *is Lipschitz-continuous with the constant*

$$L_2(\phi) = \frac{1}{\hat{\sigma}}\|A\|_{1,2}^2 + L_2(\hat{\phi}). \tag{7.2}$$

Proof. Denote $\tilde{\phi}(u) = \min_{x\in Q_1}\{\langle Ax, u\rangle_2 + \hat{f}(x)\}$. This function is concave as a minimum of linear functions. Since $\hat{f}$ is strongly convex, the solution of the above minimization problem is unique. Therefore $\tilde{\phi}(u)$ is differentiable and $\nabla\tilde{\phi}(u) = Ax_0(u)$.

Consider two points u_1 and u_2. From the first order optimality conditions for (2.3) we have

$$\langle A^* u_1 + \nabla\hat{f}(x_0(u_1)), x_0(u_2) - x_0(u_1)\rangle_1 \geq 0,$$

$$\langle A^* u_2 + \nabla\hat{f}(x_0(u_2)), x_0(u_1) - x_0(u_2)\rangle_1 \geq 0.$$

Adding these inequalities and using strong convexity of $\hat{f}(\cdot)$, we continue as follows:

$$\begin{aligned}\langle Ax_0(u_2) - Ax_0(u_1), u_1 - u_2\rangle_2 &\geq \langle \nabla \hat{f}(x_0(u_1)) - \nabla \hat{f}(x_0(u_2)), x_0(u_1) - x_0(u_2)\rangle_1 \\ \text{(by (1.4))} &\geq \hat{\sigma}\|x_0(u_1) - x_0(u_2)\|_1^2 \\ \text{(by (1.1))} &\geq \frac{\hat{\sigma}}{\|A\|_{1,2}^2}\left(\|\nabla\tilde{\phi}(u_1) - \nabla\tilde{\phi}(u_2)\|_2^*\right)^2.\end{aligned}$$

Thus, $\|\nabla\tilde{\phi}(u_1) - \nabla\tilde{\phi}(u_2)\|_2^* \leq \frac{1}{\hat{\sigma}}\|A\|_{1,2}^2 \cdot \|u_1 - u_2\|_2$, and (7.2) follows. □

LEMMA 7.2. *For any u and $\hat{u}$ from Q_2 we have*

$$\phi(\hat{u}) + \langle \nabla\phi(\hat{u}), u - \hat{u}\rangle_2 \geq -\hat{\phi}(u) + \langle Ax_0(\hat{u}), u\rangle_2 + \hat{f}(x_0(\hat{u})). \tag{7.3}$$

Proof. Let us take arbitrary u and $\hat{u}$ from Q_2. Denote $\hat{x} = x_0(\hat{u})$. Then

$$\begin{aligned}\phi(\hat{u}) + \langle \nabla\phi(\hat{u}), u - \hat{u}\rangle_2 &= -\hat{\phi}(\hat{u}) + \langle A\hat{x}, \hat{u}\rangle_2 + \hat{f}(\hat{x}) + \langle -\nabla\hat{\phi}(\hat{u}) + A\hat{x}, u - \hat{u}\rangle_2 \\ (\hat{\phi} \text{ is convex}) &\geq -\hat{\phi}(u) + \langle A\hat{x}, u\rangle_2 + \hat{f}(\hat{x}). \quad \square\end{aligned}$$

In this section we derive an optimization scheme from the following variant of the excessive gap condition:

$$f_{\mu_2}(\bar{x}) \leq \phi(\bar{u}) \tag{7.4}$$

for some $\bar{x} \in Q_1$ and $\bar{u}$ in Q_2.

This condition can be seen as a variant of condition (3.2) with $\mu_1 = 0$. However, in this section we prefer not to use the results of the previous sections since our assumptions are now different. For example, we no longer need the set Q_1 to be bounded.

LEMMA 7.3. *Let points $\bar{x}$ from Q_1 and $\bar{u}$ from Q_2 satisfy* (7.4). *Then*

$$0 \leq f(\bar{x}) - \phi(\bar{u}) \leq \mu_2 D_2. \tag{7.5}$$

Proof. Indeed, for any $x \in Q_1$ we have $f_{\mu_2}(x) \geq f(x) - \mu_2 D_2$. □

Define the adjoint gradient mapping as follows:

$$V(u) = \arg\max_{v\in Q_2}\left\{\langle \nabla\phi(u), v - u\rangle_2 - \frac{1}{2}L_2(\phi)\|v - u\|_2^2\right\}. \tag{7.6}$$

LEMMA 7.4. *The excessive gap condition* (7.4) *is valid for $\mu_2 = \frac{1}{\sigma_2}L_2(\phi)$ and*

$$\bar{x} = x_0(u_0), \quad \bar{u} = V(u_0). \tag{7.7}$$

Proof. Indeed, in view of Lemma 7.1 and (1.5) we get the following relations:

$$\begin{aligned}\phi(V(u_0)) &\geq \phi(u_0) + \langle \nabla\phi(u_0), V(u_0) - u_0\rangle_2 - \frac{1}{2}L_2(\phi)\|V(u_0) - u_0\|_2^2 \\ \text{(by (7.6))} &= \max_{u\in Q_2}\left\{\phi(u_0) + \langle \nabla\phi(u_0), u - u_0\rangle_2 - \frac{1}{2}L_2(\phi)\|u - u_0\|_2^2\right\} \\ \text{(by (2.3) and (7.1))} &= \max_{u\in Q_2}\left\{ -\hat{\phi}(u_0) + \langle Ax_0(u_0), u_0\rangle_2 + \hat{f}(x_0(u_0)) \right. \\ &\qquad \left. + \langle Ax_0(u_0) - \nabla\hat{\phi}(u_0), u - u_0\rangle_2 - \frac{1}{2}\mu_2\sigma_2\|u - u_0\|_2^2\right\} \\ (\hat{\phi} \text{ is convex and (2.4)}) &\geq \max_{u\in Q_2}\left\{-\hat{\phi}(u) + \hat{f}(x_0(u_0)) + \langle Ax_0(u_0), u\rangle_2 - \mu_2 d_2(u)\right\} \\ \text{(by (2.5))} &= f_{\mu_2}(x_0(u_0)). \quad \square\end{aligned}$$

THEOREM 7.5. *Let points $\bar{x} \in Q_1$ and $\bar{u} \in Q_2$ satisfy the excessive gap condition (7.4) for some positive μ_2. Let us fix $\tau \in (0,1)$ and choose $\mu_2^+ = (1-\tau)\mu_2$,*

$$\begin{aligned} \hat{u} &= (1-\tau)\bar{u} + \tau u_{\mu_2}(\bar{x}), \\ \bar{x}_+ &= (1-\tau)\bar{x} + \tau x_0(\hat{u}), \qquad (7.8) \\ \bar{u}_+ &= V(\hat{u}). \end{aligned}$$

Then the pair $(\bar{x}_+, \bar{u}_+)$ satisfies condition (7.4) with smoothness parameter μ_2, provided that τ is chosen in accordance with the following relation:

$$\frac{\tau^2}{1-\tau} \le \frac{\mu_2 \sigma_2}{L_2(\phi)}. \qquad (7.9)$$

Proof. Denote $\hat{x} = x_0(\hat{u})$ and $u_2 = u_{\mu_2}(\bar{x})$. In view of the second line of (7.8) and (2.5) we have

$$\begin{aligned} f_{\mu_2^+}(\bar{x}_+) &= \hat{f}(\bar{x}_+) + \max_{u \in Q_2} \left\{ \langle A((1-\tau)\bar{x} + \tau\hat{x}), u\rangle_2 - \hat{\phi}(u) - (1-\tau)\mu_2 d_2(u) \right\} \\ (\hat{f} \text{ is convex}) &\le \max_{u \in Q_2} \Big\{ (1-\tau) \left[\hat{f}(\bar{x}) + \langle A\bar{x}, u\rangle_2 - \hat{\phi}(u) - \mu_2 d_2(u) \right] \\ &\qquad + \tau[\hat{f}(\hat{x}) + \langle A\hat{x}, u\rangle_2 - \hat{\phi}(u)] \Big\} \\ (\text{by (1.2)}) &\le \max_{u \in Q_2} \Big\{ (1-\tau) \left[f_{\mu_2}(\bar{x}) - \frac{1}{2}\mu_2\sigma_2 \|u - u_2\|_2^2 \right] \\ (\text{by (7.3)}) &\qquad + \tau[\phi(\hat{u}) + \langle \nabla\phi(\hat{u}), u - \hat{u}\rangle_2] \Big\}. \end{aligned}$$

Since ϕ is concave, by (7.4) we obtain

$$\begin{aligned} f_{\mu_2}(\bar{x}) \le \phi(\bar{u}) &\le \phi(\hat{u}) + \langle \nabla\phi(\hat{u}), \bar{u} - \hat{u}\rangle_2 \\ (\text{line 1, (7.8)}) &= \phi(\hat{u}) + \tau\langle \nabla\phi(\hat{u}), \bar{u} - u_2\rangle_2. \end{aligned}$$

Hence, we can finish the proof as follows:

$$\begin{aligned} f_{\mu_2^+}(\bar{x}_+) &\le \max_{u \in Q_2} \left\{ \phi(\hat{u}) + \tau\langle \nabla\phi(\hat{u}), u - u_2\rangle_2 - \frac{1}{2}(1-\tau)\mu_2\sigma_2 \|u - u_2\|_2^2 \right\} \\ (\text{by (7.9)}) &\le \max_{u \in Q_2} \left\{ \phi(\hat{u}) + \tau\langle \nabla\phi(\hat{u}), u - u_2\rangle_2 - \frac{1}{2}\tau^2 L_2(\phi) \|u - u_2\|_2^2 \right\} \\ (v = \bar{u} + \tau(u - \bar{u})) &= \max_{v \in \bar{u} + \tau(Q_2 - \bar{u})} \left\{ \phi(\hat{u}) + \langle \nabla\phi(\hat{u}), v - \hat{u}\rangle_2 - \frac{1}{2} L_2(\phi) \|v - \hat{u}\|_2^2 \right\} \\ (Q_2 \text{ is convex}) &\le \max_{v \in Q_2} \left\{ \phi(\hat{u}) + \langle \nabla\phi(\hat{u}), v - \hat{u}\rangle_2 - \frac{1}{2} L_2(\phi) \|v - \hat{u}\|_2^2 \right\} \\ (\text{by (7.6)}) &= \phi(\hat{u}) + \langle \nabla\phi(\hat{u}), \bar{u}_+ - \hat{u}\rangle_2 - \frac{1}{2} L_2(\phi) \|\bar{u}_+ - \hat{u}\|_2^2 \\ (\text{by (1.5)}) &\le \phi(\bar{u}_+). \qquad \square \end{aligned}$$

Now we can justify the following minimization scheme.

METHOD 2.

```
1. Initialization:
   Set $\mu_{2,0} = \frac{2}{\sigma_2} L_2(\phi)$, $\bar{x}_0 = x_0(u_0)$ and $\bar{u}_0 = V(u_0)$.
2. For $k \geq 0$ iterate:
   Set $\tau_k = \frac{2}{k+3}$ and $\hat{u}_k = (1-\tau_k)\bar{u}_k + \tau_k u_{\mu_{2,k}}(\bar{x}_k)$.
     Update $\mu_{2,k+1} = (1-\tau_k)\mu_{2,k}$,
            $\bar{x}_{k+1} = (1-\tau_k)\bar{x}_k + \tau_k x_0(\hat{u}_k)$,
            $\bar{u}_{k+1} = V(\hat{u}_k)$.
```

THEOREM 7.6. *Let problem* (2.1) *satisfy Assumption* 1. *Then the pairs* $(\bar{x}_k, \bar{u}_k)$ *generated by Method* 2 *satisfy the following inequality:*

$$f(\bar{x}_k) - \phi(\bar{u}_k) \leq \frac{4L_2(\phi)D_2}{(k+1)(k+2)\sigma_2}, \tag{7.10}$$

where $L_2(\phi)$ *is given by* (7.2).

Proof. Indeed, in view of Theorem 7.5 and Lemma 7.4 we need only justify that the sequences $\{\mu_{2,k}\}_{k=0}^{\infty}$ and $\{\tau_k\}_{k=0}^{\infty}$ satisfy relation (7.9). This is straightforward because of relation

$$\mu_{2,k} = \frac{4L_2(\phi)}{(k+1)(k+2)\sigma_2},$$

which is valid for all $k \geq 0$. □

Let us conclude the paper with an example. Consider the problem

$$f(x) = \frac{1}{2}\|x\|_1^2 + \max_{1 \leq j \leq m} [f_j + \langle g_j, x - x_j \rangle_1] \quad \to \quad \min: \ x \in E_1. \tag{7.11}$$

Problems of this type arise, for example, at each iteration of the bundle method [1]. Let $E_1 = R^n$ and we choose

$$\|x\|_1^2 = \sum_{i=1}^{n} (x^{(i)})^2, \quad x \in E_1.$$

Then this problem can be solved by Method 2.

Indeed, we can represent the objective function in (7.11) in the form (2.2) using the following objects:

$$E_2 = R^m, \quad Q_2 = \Delta_m = \left\{ u \in R_+^m : \ \sum_{j=1}^{m} u^{(j)} = 1 \right\},$$

$$\hat{f}(x) = \frac{1}{2}\|x\|_1^2, \quad \hat{\phi}(u) = \langle b, u \rangle_2, \quad b^{(j)} = \langle g_j, x_j \rangle_1 - f_j, \ j = 1, \ldots, m,$$

$$A^T = (a_1, \ldots, a_m).$$

Thus, $\hat{\sigma} = 1$ and $L_2(\hat{\phi}) = 0$. Let us choose for E_2 the following norm:

$$\|u\|_2 = \sum_{j=1}^{m} |u^{(j)}|.$$

Then we can use the entropy distance function (see [3]):

$$d_2(u) = \ln m + \sum_{j=1}^{m} u^{(j)} \ln u^{(j)}, \quad u_0 = \left(\frac{1}{m}, \ldots, \frac{1}{m} \right),$$

for which $\sigma_2 = 1$ and $D_2 = \ln m$. Note that in this case

$$\|A\|_{1,2} = \max_{1 \le j \le m} \|g_j\|_1^*.$$

Thus, Method 2 as applied to problem (7.11) converges with the following rate:

$$f(\bar{x}_k) - \phi(\bar{u}_k) \le \frac{4 \ln m}{(k+1)(k+2)} \cdot \max_{1 \le j \le m} \left(\|g_j\|_1^* \right)^2.$$

Let us study the complexity of this scheme for our example. At each iteration we need to compute the following objects.

1. *Computation of* $u_{\mu_2}(\bar{x})$. This is the solution of the following problem:

$$\max_u \left\{ \sum_{j=1}^{m} u^{(j)} s^{(j)}(\bar{x}) - \mu_2 d_2(u) : \ u \in Q_2 \right\}$$

with $s^{(j)}(\bar{x}) = f_j + \langle g_j, \bar{x} - x_j \rangle$, $j = 1, \ldots, m$. In accordance with (4.14) in [3, Lemma 4], this solution can be found in a closed form:

$$u_{\mu_2}^{(j)}(\bar{x}) = e^{s^{(j)}(\bar{x})/\mu_2} \cdot \left[\sum_{l=1}^{m} e^{s^{(l)}(\bar{x})/\mu_2} \right]^{-1}, \quad j = 1, \ldots, m.$$

2. *Computation of* $x_0(\hat{u})$. In our case this is a solution to the problem

$$\min_x \left\{ \langle Ax, \hat{u} \rangle_2 + \frac{1}{2} \|x\|_1^2 : \ x \in E_1 \right\}.$$

Hence, the answer is very simple: $x_0(\hat{u}) = -A^T \hat{u}$.

3. *Computation of* $V(\hat{u})$. In our case

$$\begin{aligned} \phi(\bar{u}) \ &= \ \min_{x \in E_1} \left\{ \sum_{j=1}^{m} u^{(j)} [f_j + \langle g_j, x - x_j \rangle_1] + \frac{1}{2} \|x\|_1^2 \right\} \\ &= \ -\langle b, u \rangle_2 - \frac{1}{2} \left(\|A^T \hat{u}\|_1^* \right)^2. \end{aligned}$$

Thus, $\nabla \phi(\bar{u}) = -b - AA^T \hat{u}$. Now we can compute $V(\hat{u})$ by (7.6). In [3, section 5.1], it is shown that the complexity of finding $V(\bar{u})$ is of the order $O(m \ln m)$.

We have seen that all computations at each iteration of Method 2 as applied to problem (7.11) are very cheap. The most expensive part of the iteration is the multiplication of the matrix A by a vector. In a straightforward implementation we need three such multiplications per iteration. However, a simple modification of the order of operations can reduce this amount to two.

Acknowledgment. The author thanks the anonymous referees for their very useful suggestions.

REFERENCES

[1] J.-B. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, 1993.

[2] A. Nemirovsky and D. Yudin, *Informational Complexity and Efficient Methods for Solution of Convex Extremal Problems*, J. Wiley & Sons, New York, 1983.

[3] Yu. Nesterov, *Smooth minimization of non-smooth functions*, Math. Program., 103 (2005), pp. 127–152.

[4] Yu. Nesterov, *Introductory Lectures on Convex Optimization: Basic Course*, Kluwer Academic, Boston, MA, 2004.

SIAM J. OPTIM.
Vol. 16, No. 1, pp. 250–262

A SUFFICIENT CONDITION FOR EXACT PENALTY IN CONSTRAINED OPTIMIZATION[*]

ALEXANDER J. ZASLAVSKI[†]

Abstract. In this paper we use the penalty approach to study three constrained minimization problems. A penalty function is said to have the exact penalty property [J.-B. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms*, 2 vols., Springer-Verlag, Berlin, 1993] if there exists a penalty coefficient for which a solution of an unconstrained penalized problem is a solution of the corresponding constrained problem. In this paper we establish a very simple sufficient condition for the exact penalty property.

Key words. Clarke's generalized gradient, Ekeland's variational principle, minimization problem, penalty function

AMS subject classifications. 49M30, 90C26, 90C30

DOI. 10.1137/040612294

1. Introduction.

In this paper we use the penalty approach to study three constrained nonconvex minimization problems with Lipschitzian (on bounded sets) cost functions. The first is an equality constrained problem in a Banach space with a locally Lipschitzian constraint function; the second is an inequality constrained problem in a Banach space with a locally Lipschitzian constraint function; and the third is a problem in a finite-dimensional space with mixed constraints and smooth constraint functions.

A penalty function is said to have the exact penalty property [3] if there exists a penalty coefficient for which a solution of an unconstrained penalized problem is a solution of the corresponding constrained problem. In this paper we establish a very simple sufficient condition for the exact penalty property.

Let $(X, ||\cdot||)$ be a Banach space, $(X^*, ||\cdot||_*)$ its dual space, and let $f : X \to R^1$ be a locally Lipschitzian function. For each $x \in X$, let

$$f^0(x,h) = \limsup_{t\to 0^+, y\to x} [f(y+th) - f(y)]/t, \ h \in X,$$

be the Clarke generalized directional derivative of f at the point x [1]; let

$$\partial f(x) = \{l \in X^* : \ f^0(x,h) \geq l(h) \text{ for all } h \in X\}$$

be the Clarke generalized gradient of f at x [1]; and set

$$\Xi_f(x) = \inf\{f^0(x,h) : \ h \in X \text{ and } ||h|| = 1\}$$

(see [5]).

A point $x \in X$ is called a critical point of f if $0 \in \partial f(x)$. It is not difficult to see that $x \in X$ is a critical point of f if and only if $\Xi_f(x) \geq 0$.

A real number $c \in R^1$ is called a critical value of f if there exists a critical point x of f such that $f(x) = c$.

[*] Received by the editors July 26, 2004; accepted for publication (in revised form) December 27, 2004; published electronically September 8, 2005.
http://www.siam.org/journals/siopt/16-1/61229.html

[†] Department of Mathematics, The Technion-Israel Institute of Technology, 32000 Haifa, Israel (ajzas@tx.technion.ac.il).

It is known [1, Chap. 2, sect. 2.3] that $\partial(-f)(x) = -\partial f(x)$ for any $x \in X$. This equality implies that $x \in X$ is a critical point of f if and only if x is a critical point of $-f$ and implies that $c \in R^1$ is a critical value of f if and only if $-c$ is a critical value of $-f$.

For each function $f : X \to R^1$, set $\inf(f) = \inf\{f(z) : \ z \in X\}$. For each $x \in X$ and each $B \subset X$, put

$$d(x, B) = \inf\{||x - y|| : \ y \in B\}.$$

Consider a minimization problem $h(z) \to \min$, $z \in X$, where $h : X \to R^1$ is a continuous bounded from below function. If the space X is infinite-dimensional, then the existence of solutions of the problem is not guaranteed and in this situation we consider δ-approximate solutions. Namely, $x \in X$ is a δ-approximate solution of the problem $h(z) \to \min$, $z \in X$, where $\delta > 0$, if $h(x) \le \inf(h) + \delta$.

Let $f : X \to R^1$ be a function which is Lipschitzian on all bounded subsets of X and which satisfies the growth condition

$$\lim_{||x|| \to \infty} f(x) = \infty. \tag{1.1}$$

Clearly, f is bounded from below. Let $g : X \to R^1$ be a locally Lipschitzian function which satisfies the following Palais–Smale (PS) condition [4, 5].

If $\{x_i\}_{i=1}^{\infty} \subset X$, the sequence $\{g(x_i)\}_{i=1}^{\infty}$ is bounded, and if

$$\liminf_{i \to \infty} \Xi_g(x_i) \ge 0,$$

then there is a norm convergent subsequence of $\{x_i\}_{i=1}^{\infty}$.

Let $c \in R^1$ be such that $g^{-1}(c)$ is nonempty.

We consider the constrained problems

$$f(x) \to \min \ \text{subject to} \ x \in g^{-1}(c) \tag{P_e}$$

and

$$f(x) \to \min \ \text{subject to} \ x \in g^{-1}((-\infty, c]). \tag{P_i}$$

We associate with these two problems the corresponding families of unconstrained minimization problems

$$f(x) + \lambda|g(x) - c| \to \min, \ x \in X, \tag{$P_{\lambda e}$}$$

and

$$f(x) + \lambda \max\{g(x) - c, \ 0\} \to \min, \ x \in X, \tag{$P_{\lambda i}$}$$

where $\lambda > 0$.

The main result of this paper (Theorem 1.1) stated below implies that if the space X is finite-dimensional, c is not a critical value of g and that, if λ is sufficiently large, then any solution of problem ($P_{\lambda e}$) is a solution of problem (P_e) and any solution of problem ($P_{\lambda i}$) is a solution of problem (P_i). If the space X is infinite-dimensional, then the existence of solutions of problems ($P_{\lambda e}$) and ($P_{\lambda i}$) is not guaranteed. In this case, Theorem 1.1 implies that if c is not a critical value of g, then for each $\epsilon > 0$ there exists $\delta(\epsilon) > 0$, which depends only on ϵ such that the following property holds.

If $\lambda \geq \bar{\lambda}$ and x is a δ-approximate solution of $(P_{\lambda e})$ $((P_{\lambda i})$, resp.), then there exists a $(\bar{\lambda}\epsilon)$-approximate solution of (P_e) $((P_i)$, resp.) such that $||y - x|| \leq \epsilon$.

Here $\bar{\lambda}$ is a positive constant which does not depend on ϵ.

Set

$$\inf(f; c) = \inf\{f(z) : \ z \in g^{-1}(c)\}, \tag{1.2}$$

$$\inf(f; (-\infty, c]) = \inf\{f(z) : \ z \in X \text{ and } g(z) \leq c\}. \tag{1.3}$$

We now state the main result of the paper.

Theorem 1.1. *Assume that the number c is not a critical value of the function g. Then there exist positive numbers λ_0 and λ_1 such that for each $\epsilon > 0$ there exists $\delta \in (0, \epsilon)$ such that the following assertions hold:*

1. *If $\lambda > \lambda_0$, and if $x \in X$ satisfies*

$$f(x) + \lambda|g(x) - c| \leq \inf\{f(z) + \lambda|g(z) - c| : \ z \in X\} + \delta,$$

then there exists $y \in g^{-1}(c)$ such that

$$||y - x|| \leq \epsilon \text{ and } f(y) \leq \inf(f; c) + \lambda_1\epsilon.$$

2. *If $\lambda > \lambda_0$, and if $x \in X$ satisfies*

$$f(x) + \lambda \max\{g(x) - c, \ 0\} \leq \inf\{f(z) + \lambda \max\{g(z) - c, \ 0\} : \ z \in X\} + \delta,$$

then there exists $y \in g^{-1}((-\infty, c])$ such that

$$||y - x|| \leq \epsilon \text{ and } f(y) \leq \inf(f; (-\infty, c]) + \lambda_1\epsilon.$$

Theorem 1.1 will be proved in section 2. In this section we present several important results which follow from Theorem 1.1. We will prove these results in section 2. Theorem 1.1 implies the following result.

Theorem 1.2. *Assume that the number c is not a critical value of the function g. Then there exists $\lambda_0 > 0$ such that the following assertions hold:*

1. *For each $\lambda > \lambda_0$, and for each sequence $\{x_i\}_{i=1}^{\infty} \subset X$ which satisfies*

$$\lim_{i\to\infty} [f(x_i) + \lambda|g(x_i) - c|] = \inf\{f(z) + \lambda|g(z) - c| : \ z \in X\},$$

there exists a sequence $\{y_i\}_{i=1}^{\infty} \subset g^{-1}(c)$ such that

$$\lim_{i\to\infty} f(y_i) = \inf(f; c) \text{ and } \lim_{i\to\infty} ||y_i - x_i|| = 0.$$

2. *For each $\lambda > \lambda_0$, and for each sequence $\{x_i\}_{i=1}^{\infty} \subset X$ which satisfies*

$$\lim_{i\to\infty} [f(x_i) + \lambda \max\{g(x_i) - c, 0\}] = \inf\{f(z) + \lambda \max\{g(z) - c, 0\} : \ z \in X\},$$

there exists a sequence $\{y_i\}_{i=1}^{\infty} \subset g^{-1}((-\infty, c])$ such that

$$\lim_{i\to\infty} f(y_i) = \inf(f; (-\infty, c]) \text{ and } \lim_{i\to\infty} ||y_i - x_i|| = 0.$$

Assertion 1 of Theorem 1.2 implies the following result.

Theorem 1.3. *Assume that the number c is not a critical value of the function g and that there exists $\bar{x} \in g^{-1}(c)$ for which the following conditions hold:*

- *$f(\bar{x}) = \inf(f; c)$;*
- *any sequence $\{x_n\}_{n=1}^{\infty} \subset g^{-1}(c)$ that satisfies $\lim_{n\to\infty} f(x_n) = \inf(f; c)$ converges to $\bar{x}$ in the norm topology.*

Then there exists $\lambda_0 > 0$ such that for each $\lambda > \lambda_0$ the point $\bar{x}$ is a unique solution of the minimization problem

$$f(z) + \lambda|g(z) - c| \to \min,\ z \in X.$$

Assertion 2 of Theorem 1.2 implies the following result.

THEOREM 1.4. *Assume that the number c is not a critical value of the function g and that there exists $\bar{x} \in g^{-1}((-\infty, c])$ for which the following conditions hold:*

- *$f(\bar{x}) = \inf(f; (-\infty, c])$;*
- *any sequence $\{x_n\}_{n=1}^{\infty} \subset g^{-1}((-\infty, c])$ that satisfies $\lim_{n\to\infty} f(x_n) = \inf(f; (-\infty, c])$ converges to $\bar{x}$ in the norm topology.*

Then there exists $\lambda_0 > 0$ such that for each $\lambda > \lambda_0$ the point $\bar{x}$ is a unique solution of the minimization problem

$$f(z) + \lambda \max\{g(z) - c, 0\} \to \min,\ z \in X.$$

The next result follows from Theorem 1.2.

THEOREM 1.5. *Assume that $X = R^n$ and that the number c is not a critical value of the function g. Then there exists $\lambda_0 > 0$ such that the following assertions hold:*

1. *If $\lambda > \lambda_0$, and if x is a solution of the minimization problem*

$$f(z) + \lambda|g(z) - c| \to \min,\ z \in X,$$

then $x \in g^{-1}(c)$ and $f(x) = \inf(f; c)$.

2. *If $\lambda > \lambda_0$, and if x is a solution of the minimization problem*

$$f(z) + \lambda \max\{g(z) - c, 0\} \to \min,\ z \in X,$$

then $g(x) \le c$ and $f(x) = \inf(f; (-\infty, c])$.

Example 1. Assume that $X = R^n$, $g \in C^1(R^n)$, and that the gradient of g is not zero at any point $x \in R^n$. Then Theorems 1.1, 1.2, and 1.5 hold.

Example 2. Assume that $X = R^n$, g is convex and bounded from below, and $c > \inf(g)$. Then Theorems 1.1, 1.2, and 1.5 hold.

Now we give an example which shows that exactness fails when c is a critical value of g.

Example 3. Let $X = R^1$ and consider the minimization problem

$$f(x) \to \min,\ x \in R^1,\ g(x) = 0,$$

where $f(x) = (x - 10)^2$, $x \in R^1$, and

$$g(x) = (x-1)^2,\ x \in [1, \infty),\ g(x) = (x+1)^2,\ x \in (-\infty, -1],\ g(x) = 0,\ x \in (-1, 1).$$

This problem is equivalent to the problem

$$f(x) \to \min,\ x \in R^1,\ g(x) \le 0.$$

Clearly, zero is a critical value of g and $\bar{x} = 1$ is a unique solution of the problem.

We show that, for each $\lambda > 0$, $\inf(f + \lambda g) < f(1) = 81$. Fix $\lambda > 0$. For each $x \in [1, \min\{2, 1 + 4/\lambda\}]$,

$$(f + \lambda g)'(x) = 2(x - 10) + 2\lambda(x - 1) \le -16 + 2\lambda(x - 1) \le -16 + 2\lambda(4/\lambda) = -8.$$

This relation implies that $\inf(f + \lambda g) < f(1)$.

2. Proofs of Theorems 1.1–1.4.

Proof of Theorem 1.1. We prove assertions 1 and 2 simultaneously.

Set

$$A = g^{-1}(c) \text{ in the case of assertion 1}$$

and

$$A = g^{-1}((-\infty, c]) \text{ in the case of assertion 2.}$$

For each $\lambda > 0$ define a function $\psi_\lambda : X \to R^1$ as

$$\psi_\lambda(z) = f(z) + \lambda|g(z) - c|, \ z \in X, \tag{2.1}$$

in the case of assertion 1 and as

$$\psi_\lambda(z) = f(z) + \lambda \max\{g(z) - c, 0\}, \ z \in X, \tag{2.2}$$

in the case of assertion 2.

Clearly, the function ψ_λ is locally Lipschitzian for all $\lambda > 0$.

We show that there exists $\lambda_0 > 0$ such that the following property holds.

(P1) For each $\epsilon \in (0, 1)$ there exists $\delta \in (0, \epsilon)$ such that for each $\lambda > \lambda_0$, and for each $x \in X$ which satisfies

$$\psi_\lambda(x) \le \inf(\psi_\lambda) + \delta,$$

there exists $y \in A$ for which $||y - x|| \le \epsilon$.

Let us assume the converse. Then for each natural number k there exist

$$\epsilon_k \in (0, 1), \ \lambda_k > k, \ \text{and } x_k \in X \tag{2.3}$$

such that

$$\psi_{\lambda_k}(x_k) \le \inf(\psi_{\lambda_k}) + 2^{-1}\epsilon_k k^{-2}, \tag{2.4}$$

$$d(x_k, A) \ge \epsilon_k. \tag{2.5}$$

Let k be a natural number. It follows from (2.4) and Ekeland's variational principle [2] that there exists $y_k \in X$ such that

$$\psi_{\lambda_k}(y_k) \le \psi_{\lambda_k}(x_k), \tag{2.6}$$

$$||y_k - x_k|| \le 2^{-1}k^{-1}\epsilon_k, \tag{2.7}$$

$$\psi_{\lambda_k}(y_k) \le \psi_{\lambda_k}(z) + k^{-1}||z - y_k|| \text{ for all } z \in X. \tag{2.8}$$

By (2.5) and (2.7),

$$y_k \notin A \text{ for all natural numbers } k. \tag{2.9}$$

In the case of assertion 2, we obtain that

$$g(y_k) > c \text{ for all natural numbers } k. \tag{2.10}$$

In the case of assertion 1, we obtain that, for each natural number k,

$$\text{either } g(y_k) > c \text{ or } g(y_k) < c.$$

In the case of assertion 1, by extracting a subsequence and re-indexing we may assume that either $g(y_k) > c$ for all natural numbers k or $g(y_k) < c$ for all natural numbers k. Replacing g with $-g$ and c with $-c$, if necessary, we may assume without loss of generality that (2.10) is valid in the case of assertion 1 too. Now (2.10) is valid in both cases.

Let k be a natural number. It follows from (2.8) that

$$0 \in \partial \psi_{\lambda_k}(y_k) + \{z \in X^* : \ ||z|| \le 1/k\}. \tag{2.11}$$

By (2.1), (2.2), and (2.10),

$$\partial \psi_{\lambda_k}(y_k) = \partial(f + \lambda_k g)(y_k) \subset f(y_k) + \lambda_k \partial g(y_k). \tag{2.12}$$

Relations (2.11) and (2.12) imply that

$$0 \in \partial g(y_k) + \lambda_k^{-1} \partial f(y_k) + \lambda_k^{-1}\{z \in X^* : \ ||z|| \le 1/k\}. \tag{2.13}$$

It follows from (2.1)–(2.4), (2.6), and (2.10) that, for each natural number k,

$$\begin{aligned} f(y_k) &\le f(y_k) + \lambda(g(y_k) - c) = \psi_{\lambda_k}(y_k) \le \inf(\psi_{\lambda_k}) + 1 \\ &\le \inf\{\psi_{\lambda_k}(z) : \ z \in A\} + 1 = \inf\{f(z) : \ z \in A\} + 1. \end{aligned}$$

In view of this inequality and growth condition (1.1), the sequence $\{y_k\}_{k=1}^{\infty}$ is bounded. Since f is Lipschitzian on bounded subsets of X, it follows from the boundedness of the sequence $\{y_k\}_{k=1}^{\infty}$ that there exists $L > 0$ such that

$$\partial f(y_k) \subset \{l \in X^* : \ ||l|| \le L\} \tag{2.14}$$

for each natural number k.

It follows from (2.1)–(2.4), (2.6), and (2.10), that, for each natural number k,

$$\begin{aligned} f(y_k) + \lambda_k(g(y_k) - c) &= \psi_{\lambda_k}(y_k) \le \psi_{\lambda_k}(x_k) \le \inf(\psi_{\lambda_k}) + 1 \\ &\le \inf\{\psi_{\lambda_k}(z) : \ z \in A\} + 1 = \inf\{f(z) : \ z \in A\} + 1. \end{aligned} \tag{2.15}$$

By (2.3), (2.10), and (2.15), for each integer $k \ge 1$,

$$0 < g(y_k) - c \le \lambda_k^{-1}[\inf\{f(z) : \ z \in A\} - \inf(f) + 1] \to 0 \text{ as } k \to \infty.$$

Therefore

$$\lim_{k \to \infty} g(y_k) = c. \tag{2.16}$$

By (2.3), (2.13), and (2.14), for each integer $k \ge 1$,

$$\begin{aligned} 0 &\in \partial g(y_k) + \lambda_k^{-1}\{l \in X^* : \ ||l|| \le L\} + \{l \in X^* : \ ||l|| \le k^{-1}\} \\ &\subset \partial g(y_k) + \{l \in X^* : \ ||l| \le k^{-1}(1 + L)\}. \end{aligned}$$

This inclusion implies that

$$\liminf_{k\to\infty} \Xi_g(y_k) \geq 0. \tag{2.17}$$

It follows from (2.16), (2.17), and the PS condition that there exists a strictly increasing sequence of natural numbers $\{k_j\}_{j=1}^{\infty}$ such that $\{y_{k_j}\}_{j=1}^{\infty}$ converges in the norm topology to $y_* \in X$. In view of (2.16), $g(y_*) = c$. Inequality (2.17) and upper semicontinuity of the Clarke generalized directional derivative $g^0(\xi, \eta)$ with respect to ξ imply that $\Xi_g(y_*) \geq 0$ and $0 \in \partial g(y_*)$. Since $g(y_*) = c$, we obtain that c is a critical value of g, which is a contradiction. The contradiction proves that there exists $\lambda_0 > 0$ such that property (P1) holds.

In view of growth condition (1.1), there exists $K_1 > 0$ such that

$$||x|| \leq K_1 \text{ for each } x \in X \text{ satisfying } f(x) \leq \inf\{f(z) :\ z \in A\} + 1. \tag{2.18}$$

Since f is Lipschitzian on bounded subsets of X, there exists $\lambda_1 > 2$ such that

$$||f(x_1) - f(x_2)|| \leq 2^{-1}\lambda_1 ||x_1 - x_2|| \tag{2.19}$$

for each $x_1, x_2 \in X$ satisfying $||x_i|| \leq K_1 + 1,\ i = 1, 2$.

Assume that $\epsilon \in (0, 1)$. Let $\delta \in (0, \epsilon)$ be guaranteed by property (P1). Now assume that

$$\lambda > \lambda_0,\ x \in X \text{ and } \psi_\lambda(x) \leq \inf(\psi_\lambda) + \delta. \tag{2.20}$$

By property (P1) there exists $y \in X$ such that

$$y \in A \text{ and } ||y - x|| \leq \epsilon. \tag{2.21}$$

It follows from (2.1), (2.2), and (2.20) that

$$f(x) \leq \psi_\lambda(x) \leq \inf(\psi_\lambda) + \delta \leq \inf(\psi_\lambda) + 1$$

$$\leq \inf\{\psi_\lambda(z) :\ z \in A\} + 1 = \inf\{f(z) :\ z \in A\} + 1. \tag{2.22}$$

Relations (2.18) and (2.22) imply that

$$||x|| \leq K_1. \tag{2.23}$$

By (2.21) and (2.23),

$$||y|| \leq ||x|| + ||y - x|| \leq K_1 + 1. \tag{2.24}$$

In view of (2.23), (2.24), the choice of λ_1 (see (2.19)), (2.21), (2.23), and (2.24),

$$|f(y) - f(x)| \leq 2^{-1}\lambda_1 ||y - x|| \leq 2^{-1}\lambda_1 \epsilon. \tag{2.25}$$

It follows from (2.1), (2.2), (2.20), (2.25), and the inequalities $\delta < \epsilon$ and $\lambda_1 > 2$ that

$$f(y) \leq 2^{-1}\lambda_1\epsilon + f(x) \leq 2^{-1}\lambda_1\epsilon + \psi_\lambda(x) \leq 2^{-1}\lambda_1\epsilon + \inf(\psi_\lambda) + \delta$$

$$\leq \lambda_1\epsilon + \inf\{f(z) :\ z \in A\}.$$

Theorem 1.1 is proved. □

Proof of Theorem 1.2. We prove assertions 1 and 2 simultaneously. Let positive numbers λ_0 and λ_1 be guaranteed by Theorem 1.1. For each $\lambda > 0$, define a function $\psi_\lambda : X \to R^1$ by (2.1) in the case of assertion 1 and by (2.2) in the case of assertion 2. Set $A = g^{-1}(c)$ in the case of assertion 1 and $A = g^{-1}((-\infty, c])$ in the case of assertion 2.

Assume that $\lambda > \lambda_0$ and that a sequence $\{x_i\}_{i=1}^\infty \subset X$ satisfies

$$\lim_{i\to\infty} \psi_\lambda(x_i) = \inf(\psi_\lambda). \tag{2.26}$$

It follows from the choice of λ_0 that for each integer $j \geq 1$ there exists $\delta_j > 0$ such that the following property holds.

(P2) If $x \in X$ satisfies $\psi_\lambda(x) \leq \inf(\psi_\lambda) + \delta_j$, then there exists $y \in A$ such that $||y - x|| \leq 1/j$.

By (2.26) there exists a strictly increasing sequence of natural numbers $\{t_j\}_{j=1}^\infty$ such that, for each integer $j \geq 1$,

$$\psi_\lambda(x_i) \leq \inf(\psi_\lambda) + \delta_j \text{ for all integers } i \geq t_j. \tag{2.27}$$

It follows from property (P2) and (2.27) that, for each integer $j \geq 1$,

$$d(x_i, A) \leq 1/j \text{ for all integers } i \geq t_j.$$

This implies that there exists a sequence $\{y_i\}_{i=1}^\infty \subset A$ such that

$$\lim_{i\to\infty} ||y_i - x_i|| = 0. \tag{2.28}$$

In view of growth condition (1.1) and (2.26), the sequence $\{x_i\}_{i=1}^\infty$ is bounded. Since f is Lipschitzian on bounded subsets of X, it follows from (2.1), (2.2), (2.26), and (2.28) that

$$\inf\{f(z) : \ z \in A\} \geq \inf(\psi_\lambda) = \lim_{i\to\infty} \psi_\lambda(x_i) \geq \limsup_{i\to\infty} f(x_i) = \limsup_{i\to\infty} f(y_i).$$

Together with the inclusion $\{y_i\}_{i=1}^\infty \subset A$, this relation implies that $\lim_{i\to\infty} f(y_i) = \inf\{f(z) : \ z \in A\}$. Theorem 1.2 is proved. □

Proofs of Theorems 1.3 *and* 1.4. We prove Theorems 1.3 and 1.4 simultaneously. Let a positive number λ_0 be guaranteed by Theorem 1.2. For each $\lambda > 0$ define a function $\psi_\lambda : X \to R^1$ by (2.1) in the case of Theorem 1.3 and by (2.2) in the case of Theorem 1.4. Set $A = g^{-1}(c)$ in the case of Theorem 1.3 and $A = g^{-1}((-\infty, c])$ in the case of Theorem 1.4.

Let $\lambda > \lambda_0$. Assume that a sequence $\{x_i\}_{i=1}^\infty \subset X$ satisfies

$$\lim_{i\to\infty} \psi_\lambda(x_i) = \inf(\psi_\lambda). \tag{2.29}$$

In view of Theorem 1.2, there exists a sequence

$$\{y_i\}_{i=1}^\infty \subset A \tag{2.30}$$

such that

$$\lim_{i\to\infty} ||y_i - x_i|| = 0 \tag{2.31}$$

and

$$\lim_{i\to\infty} f(y_i) = \inf\{f(z) :\ z \in A\}. \tag{2.32}$$

Relations (2.30) and (2.32) imply that $\lim_{i\to\infty} ||y_i - \bar{x}|| = 0$. Together with (2.31), this equality implies that

$$\lim_{i\to\infty} ||x_i - \bar{x}|| = 0. \tag{2.33}$$

Thus we have shown that if $\{x_i\}_{i=1}^{\infty} \subset X$ satisfies (2.29), then (2.33) is true. This implies that $\bar{x}$ is a unique solution of the minimization problem $\psi_\lambda(z) \to \min$, $z \in X$. Theorems 1.3 and 1.4 are proved. □

3. An optimization problem with mixed constraints in a finite-dimensional space. Let p be a nonnegative integer; let m, n be natural numbers such that $p \le m \le n$; let $X = R^n$ with the Euclidean norm; and let

$$I_1 = \{i :\ i \text{ is an integer such that } 1 \le i \le p\},$$
$$I_2 = \{i :\ i \text{ is an integer such that } p < i \le m\}.$$

(Note that one of the sets I_1, I_2 may be empty.)

For each $h : R^n \to R^1$ set $\inf(h) = \inf\{h(z) :\ z \in R^n\}$. For each $h \in C^1(R^n)$ put

$$\nabla h(z) = (\partial h/\partial z_1(z), \dots, \partial h/\partial z_n(z)),\ z \in R^n.$$

Assume that $f_0 : R^n \to R^1$ is a locally Lipschitzian function which satisfies the growth condition

$$\lim_{||x||\to\infty} f_0(x) = \infty. \tag{3.1}$$

Let $F = (f_1, \dots, f_m) : R^n \to R^m$ and let $f_i \in C^1(R^n)$, $i = 1, \dots, m$. The points of R^n at which the rank of F is less than m are called critical points of F. A point $c \in R^m$ such that $F^{-1}(c)$ contains at least one critical point is called a critical value of F.

Let $c = (c_1, \dots, c_m) \in R^m$. In this section we consider the optimization problem

$$f_0(x) \to \min \tag{P}$$

$$\text{subject to } x \in R^n,\ f_i(x) = c_i,\ i \in I_1,\ f_j(x) \le c_j,\ j \in I_2.$$

Set

$$A = \{z \in R^n :\ f_i(z) = c_i,\ i \in I_1,\ \text{and } f_j(z) \le c_j,\ j \in I_2\}. \tag{3.2}$$

We assume that $A \ne \emptyset$. Since f_0 satisfies growth condition (3.1), problem (P) has a solution. Set

$$\inf(f_0; A) = \inf\{f_0(z) :\ z \in A\}.$$

For each vector $\gamma = (\gamma_1, \dots, \gamma_m) \in (0, \infty)^m$, define

$$\psi_\gamma(z) = f_0(z) + \sum_{i\in I_1} \gamma_i |f_i(z) - c_i| + \sum_{i\in I_2} \gamma_i \max\{f_i(z) - c_i, 0\},\ z \in R^n. \tag{3.3}$$

Clearly, for each $\gamma \in (0, \infty)^m$ the function ψ_γ is locally Lipschitzian, and the problem

$$\psi_\gamma(z) \to \min,\ z \in R^n,$$

has a solution.

In this section we establish the following result.

THEOREM 3.1. *Assume that the following condition holds:*

- *For each finite strictly increasing sequence of natural numbers $\{j_i\}_{i=1}^q$ that satisfies*

$$I_1 \subset \{j_1, \dots, j_q\} \subset \{1, \dots, m\},$$

the point $(c_{j_1}, \dots, c_{j_q})$ is not a critical value of the mapping $(f_{j_1}, \dots, f_{j_q}) : R^n \to R^q$.

- *Let $\gamma = (\gamma_1, \dots, \gamma_m) \in (0, \infty)^m$. Then there exists $\lambda_0 > 0$ such that if $\lambda > \lambda_0$, and if $x \in R^n$ satisfies*

$$\psi_{\lambda\gamma}(x) = \inf(\psi_{\lambda\gamma}),$$

then $f_i(x) = c_i,\ i \in I_1,\ f_j(x) \le c_j,\ j \in I_2$, and $f_0(x) = \inf(f; A)$.

Proof. Let us assume the converse. Then there exist a sequence $\{\lambda_k\}_{k=1}^\infty \subset (0, \infty)$ and a sequence $\{x^{(k)}\}_{k=1}^\infty \subset R^n$ such that

$$\lambda_k \ge k \text{ for all natural numbers } k, \tag{3.4}$$

$$\psi_{\lambda_k\gamma}(x^{(k)}) = \inf(\psi_{\lambda_k\gamma}) \text{ for all natural numbers } k, \tag{3.5}$$

$$x^{(k)} \notin A \text{ for all natural numbers } k. \tag{3.6}$$

For each natural number k, set

$$\begin{aligned} I_{1k+} &= \{i \in I_1 :\ f_i(x^{(k)}) > c_i\},\ I_{1k-} = \{i \in I_1 :\ f_i(x^{(k)}) < c_i\}, \\ I_{2k+} &= \{i \in I_2 :\ f_i(x^{(k)}) > c_i\},\ I_{2k-} = \{i \in I_2 :\ f_i(x^{(k)}) < c_i\}. \end{aligned} \tag{3.7}$$

By (3.2), (3.6), and (3.7),

$$I_{1k+} \cup I_{1k-} \cup I_{2k+} \ne \emptyset \text{ for all integers } k \ge 1. \tag{3.8}$$

Extracting a subsequence and re-indexing, we may assume without loss of generality that, for all natural numbers k,

$$I_{1k+} = I_{11+},\ I_{1k-} = I_{11-},\ I_{2k+} = I_{21+},\ I_{2k-} = I_{21-}.$$

Set

$$I_{11} = I_1 \setminus (I_{11+} \cup I_{11-}),\ I_{21} = I_2 \setminus (I_{21+} \cup I_{21-}). \tag{3.9}$$

It follows from (3.3) and (3.5) that, for each natural number k,

$$\begin{aligned} &\inf\{f_0(z) :\ z \in A\} = \inf\{\psi_{\lambda_k\gamma}(z) :\ z \in A\} \ge \inf(\psi_{\lambda_k\gamma}) \\ &= \psi_{\lambda_k\gamma}(x^{(k)}) = f_0(x^{(k)}) + \sum_{i \in I_1} \lambda_k\gamma_i |f_i(x^{(k)}) - c_i| + \sum_{i \in I_2} \lambda_k\gamma_i \max\{f_i(x^{(k)}) - c_i, 0\}. \end{aligned} \tag{3.10}$$

Relation (3.10) implies that

$$f_0(x^{(k)}) \le \inf\{f_0(z):\ z \in A\} \text{ for all natural numbers } k. \tag{3.11}$$

Since f_0 satisfies growth condition (3.1), it follows from (3.11) that the sequence $\{x^{(k)}\}_{k=1}^{\infty}$ is bounded. Extracting a subsequence and re-indexing, we may assume without loss of generality that there exists

$$\bar{x} = \lim_{k\to\infty} x^{(k)}. \tag{3.12}$$

Since f_0 is locally Lipschitzian, there exists a number $L > 0$ such that

$$\partial f_0(x^{(k)}) \subset \{l \in R^n:\ ||l|| \le L\} \text{ for all natural numbers } k. \tag{3.13}$$

Let $i \in I_1$. It follows from (3.10) and (3.12) that

$$|f_i(\bar{x}) - c_i| = \lim_{k\to\infty} |f_i(x^{(k)}) - c_i| \le \limsup_{k\to\infty} \lambda_k^{-1}\gamma_i^{-1}[\inf(f_0;A) - \inf(f_0)] = 0.$$

Therefore

$$f_i(\bar{x}) = c_i \text{ for all } i \in I_1. \tag{3.14}$$

Let $i \in I_2$. It follows from (3.10) and (3.12) that

$$\begin{aligned} \max\{f_i(\bar{x}) - c_i, 0\} &= \lim_{k\to\infty} \max\{f_i(x^{(k)}) - c_i, 0\} \\ &\le \limsup_{k\to\infty} \lambda_k^{-1}\gamma_i^{-1}[\inf(f_0;A) - \inf(f_0)] = 0. \end{aligned}$$

Therefore

$$f_i(\bar{x}) \le c_i \text{ for all } i \in I_2. \tag{3.15}$$

It follows from (3.3), (3.5), (3.7), and (3.9) that, for each integer $k \ge 1$,

$$\begin{aligned} 0 \in \partial\psi_{\lambda_k\gamma}(x^{(k)}) \subset \partial f_0(x^{(k)}) &+ \sum_{i\in I_{11+}} \lambda_k\gamma_i \nabla f_i(x^{(k)}) - \sum_{i\in I_{11-}} \lambda_k\gamma_i \nabla f_i(x^{(k)}) \\ &+ \sum_{i\in I_{11}} \lambda_k\gamma_i \{\alpha\nabla f_i(x^{(k)}):\ \alpha \in [-1,1]\} \\ &+ \sum_{i\in I_{21+}} \lambda_k\gamma_i \nabla f_i(x^{(k)}) \\ &+ \sum_{i\in I_{21}} \lambda_k\gamma_i \{\alpha\nabla f_i(x^{(k)}):\ \alpha \in [0,1]\}. \end{aligned} \tag{3.16}$$

In view of (3.13) and (3.16), for each integer $k \ge 1$,

$$\begin{aligned} &\Bigg[\sum_{i\in I_{11+}} \gamma_i\nabla f_i(x^{(k)}) - \sum_{i\in I_{11-}} \gamma_i\nabla f_i(x^{(k)}) + \sum_{i\in I_{21+}} \gamma_i\nabla f_i(x^{(k)}) \\ &\quad + \sum_{i\in I_{11}} \{\alpha\nabla f_i(x^{(k)}):\ \alpha\in[-\gamma_i,\gamma_i]\} + \sum_{i\in I_{21}} \{\alpha\nabla f_i(x^{(k)}):\ \alpha\in[0,\gamma_i]\}\Bigg] \\ &\quad \cap \{l \in R^n:\ ||l|| \le L/\lambda_k\} \ne \emptyset. \end{aligned}$$

Therefore, for each natural number k and each $i \in I_{11} \cup I_{21}$, there exists

$$\alpha_{ki} \in [-\gamma_i, \gamma_i] \tag{3.17}$$

such that

$$\Bigg\| \sum_{i \in I_{11+}} \gamma_i \nabla f_i(x^{(k)}) - \sum_{i \in I_{11-}} \gamma_i \nabla f_i(x^{(k)}) + \sum_{i \in I_{21+}} \gamma_i \nabla f_i(x^{(k)}) + \sum_{i \in I_{11} \cup I_{21}} \alpha_{ki} \nabla f_i(x^{(k)}) \Bigg\| \le L/\lambda_k. \tag{3.18}$$

Extracting a subsequence and re-indexing, we may assume without loss of generality that, for each $i \in I_{11} \cup I_{21}$, there exists

$$\alpha_i = \lim_{k \to \infty} \alpha_{ki}. \tag{3.19}$$

It follows from (3.18), (3.19), (3.12), and (3.4) that

$$\sum_{i \in I_{11+}} \gamma_i \nabla f_i(\bar{x}) - \sum_{i \in I_{11-}} \gamma_i \nabla f_i(\bar{x}) + \sum_{i \in I_{21+}} \gamma_i \nabla f_i(\bar{x}) + \sum_{i \in I_{11} \cup I_{21}} \alpha_i \nabla f_i(\bar{x}) = 0. \tag{3.20}$$

By (3.7), (3.9), and (3.15),

$$f_i(\bar{x}) = c_i,\ i \in I_{21+} \cup I_{21}.$$

Together with (3.14), this implies that

$$f_i(\bar{x}) = c_i,\ i \in I_1 \cup I_{21+} \cup I_{21}. \tag{3.21}$$

If $I_{21+} \cup I_{21} = \emptyset$, define

$$\tilde{F} = (f_1, \dots, f_p) : R^n \to R^p,\ \tilde{c} = (c_1, \dots, c_p).$$

If $I_{21+} \cup I_{21} \neq \emptyset$, set

$$\tilde{F} = (f_1, \dots, f_p, f_{j_1}, \dots, f_{j_q}) : R^n \to R^{p+q},\ \tilde{c} = (c_1, \dots, c_p, c_{j_1}, \dots, c_{j_q}),$$

where $\{j_i\}_{i=1}^q$ is a strictly increasing sequence of integers such that $\{j_1, \dots, j_q\} = I_{21+} \cup I_{21}$. In view of (3.20) and (3.8), $\bar{x}$ is a critical point of $\tilde{F}$. Together with (3.21), this implies that $\tilde{c}$ is a critical value of $\tilde{F}$, which is a contradiction. The contradiction proves Theorem 3.1. □

Note that in the proof of Theorem 3.1 we did not use the results of the previous sections. Now we explain why we did not try to apply them.

Let us consider a particular case of the problem studied in this section with $I_2 = \emptyset$, $p = m$, and $c = 0$. Thus we have the optimization problem

$$f_0(x) \to \min$$
$$\text{subject to } x \in R^n,\ f_i(x) = 0,\ i \in I_1.$$

We assume that all the functions f_i, $i \in I_1$, are linear and that they are linear independent. Clearly, Theorem 3.1 holds for this problem. The problem

$$f_0(z) + \sum_{i \in I_1} \lambda \gamma_i |f_i(z)| \to \min,\ z \in X,$$

where $\lambda > 0$ and $\gamma_i > 0$, $i \in I_1$, is the corresponding unconstrained penalized problem. We cannot apply the results of section 1 since zero is a critical value of the function

$$g(z) = \sum_{i \in I_1} \gamma_i |f_i(z)|,\ z \in R^n.$$

Acknowledgments. The author is very grateful to the referees for their helpful comments and suggestions.

REFERENCES

[1] F. H. Clarke, *Optimization and Nonsmooth Analysis*, 2nd ed., Classics in Appl. Math. 5, SIAM, Philadelphia, 1990.

[2] I. Ekeland, *On the variational principle*, J. Math. Anal. Appl., 47 (1974), pp. 324–353.

[3] J.-B. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms*, 2 vols., Springer-Verlag, Berlin, 1993.

[4] R. Palais, *Lusternik-Schnirelman theory on Banach manifolds*, Topology, 5 (1966), pp. 115–132.

[5] A. J. Zaslavski, *On critical points of Lipschitz functions on smooth manifolds*, Siberian Math. J., 22 (1981), pp. 63–68.

SIAM J. OPTIM.
Vol. 16, No. 1, pp. 263–296

COMPUTING THE LOCAL-NONGLOBAL MINIMIZER OF A LARGE SCALE TRUST-REGION SUBPROBLEM[*]

CHARLES FORTIN[†]

Abstract. We propose two algorithms for computing the local-nonglobal minimizer of a quadratic function subject to a constraint set that is a Euclidean sphere. We also discuss the case where the constraint set is a Euclidean ball. At each iteration of the algorithms, we compute the two smallest eigenvalues of a parametric matrix using an ARPACK subroutine. Only matrix-vector multiplications are required. Hence, we are able to exploit the possible sparsity of the Hessian matrix of the quadratic objective, making the algorithms suitable for large problems. This improves previous approaches based on matrix factorizations. We also give a geometric relationship, based on extremal ellipsoids, between the global and the local-nonglobal minimizers of the quadratic function under the given sphere constraint.

Key words. trust-region subproblem, local minimizers, extremal ellipsoids

AMS subject classifications. 90C06, 90C26, 93B60

DOI. 10.1137/030602290

1. Introduction.

Consider the following quadratic minimization problems:

$$\text{(TRS)} \qquad \begin{array}{ll} \min\limits_{x} & x^T A x - 2a^T x \\ \text{s.t.} & \|x\| \leq \Delta, \end{array} \tag{1.1}$$

and

$$\text{(TRS=)} \qquad \begin{array}{ll} \min\limits_{x} & x^T A x - 2a^T x \\ \text{s.t.} & \|x\| = \Delta. \end{array} \tag{1.2}$$

Here, A is an $n \times n$ symmetric (possibly indefinite) matrix, a is an n-vector, x is the n-vector of unknowns, and the ball radius Δ is a positive scalar. All matrix and vector entries are real. Without loss of generality, we assume in this paper, unless mentioned otherwise, that $\Delta = 1$, since we may scale the matrix A, the vector a, and the vector x, respectively, by Δ^2, Δ, and $1/\Delta$ so that we end up minimizing over the unit ball. This is done to simplify the notation.

Problem (1.1) is referred to as the trust-region subproblem (denoted TRS). However, we will mainly consider the neighboring problem (1.2), where the inequality constraint is replaced by an equality constraint. Computing a solution for either problem has been well studied (among many references [6, 9, 15, 18, 21, 22, 25, 27, 28]). Such solutions will be referred to as *global minimizers* of problem (1.1) or (1.2). However, each problem may possess a local minimizer which is not a global minimizer. This feasible solution will be referred to as a *local-nonglobal minimizer*. In a paper by Martínez [16], this local-nonglobal minimizer is well characterized. In particular, it is shown that such a minimizer may not exist, but when it does, it is unique. An

[*] Received by the editors December 8, 2003; accepted for publication (in revised form) January 6, 2004; published electronically September 15, 2005. This work was supported by a doctoral scholarship (B2) from the Fonds de recherche sur la nature et les technologies du Québec and a doctoral scholarship from the Institut des Sciences Mathématiques.

http://www.siam.org/journals/siopt/16-1/60229.html

[†] Department of Mathematics and Statistics, McGill University, 805 Sherbrooke Ouest, Montréal, Québec, Canada, H3A 2K6 (fortin@math.mcgill.ca).

algorithm for computing such a point or declaring it does not exist, and requiring Choleski or LU factorizations, or spectral decompositions of matrices of the same dimension and as dense as the matrix A, is proposed. Therefore, if A is a large but sparse matrix, this algorithm may be unable to exploit the potential sparsity of the matrix A.

In a paper by Rendl and Wolkowicz [21], an algorithm was proposed for computing a global minimizer of problem (1.1) or (1.2) and exploiting the sparsity of the matrix A. The algorithm is based on a reformulation of problem (1.2) in terms of a parametric eigenvalue problems. The parametric matrices which appear in these problems have eigenvalues which interlace the eigenvalues of A.

In this paper, we build on this algorithm to propose two methods for either computing a local-nonglobal minimizers of problem (1.2), if it exists, or proving such a minimizer does not exist. The main efforts of the methods lie in computing the first two eigenvalues of parametric matrices, and only matrix-vector product are required. In this sense, the algorithms are *matrix-free*. We discuss how we may compute a local-nonglobal minimizer of problem (1.1) in the appendix. Since a local-nonglobal minimizer of problem (1.1) is also a local-nonglobal minimizer of problem (1.2), the main idea is to compute the latter local-nonglobal minimizer and monitor the sign of the Lagrange multiplier.

The interest in a local-nonglobal minimizer of problem (1.1) is that it may be a global minimizer of the following problem:

$$\begin{array}{ll} \min\limits_{x} & x^T A x - 2a^T x \\ \text{s.t.} & h(x) \leq 0, \\ & \|x\| \leq \Delta, \end{array} \tag{1.3}$$

where $h : \mathbb{R}^n \to \mathbb{R}^m$. Indeed, if x^* is a global minimizer of the latter problem and $h(x^*) < 0$, then clearly x^* is a local minimizer of problem (1.1). In this case, x^* is a local-nonglobal minimizer of problem (1.1) if all global minimizers of problem (1.1) do not satisfy the first constraint of problem (1.3). An important special case known as the *two trust-region subproblem* is when h is a convex quadratic [1, 11]. This problem appears while computing the Celis–Dennis–Tapia problem [5, 20, 31] in a sequential quadratic programming approach for solving nonlinear programs. It also appears as a subproblem in the numerical solution of parameter identification problems of the form

$$\begin{array}{ll} \min\limits_{x} & \|F(x) - y\|^2 \\ \text{s.t.} & \|x\| \leq \Delta; \end{array}$$

see [10, 11]. For this subproblem, the constraints in (1.3) are two ball constraints. More generally, Martínez and Santos [17] described an algorithm for minimizing a differentiable function over a Euclidean ball, where minimizing a quadratic function over the intersection of two Euclidean balls also appears as a subproblem.

It is important to mention, even in the case where h is a convex quadratic, that there are no known polynomial time algorithms for computing the global minimum of problem (1.3). It is not known either if the problem is NP-hard. Therefore, in general one may only expect approximate solutions. However, in special cases, it is possible to compute a solution as close as we want to the exact solution. For example, the paper of Ye and Zhang [30] combines the matrix decomposition result of Sturm and Zhang [26] and semidefinite relaxation to show that some cases of the two trust-region subproblem may be solved. They even propose an algorithm, for the two trust-region subproblem, which follows the path of solutions of a family of parametrized problems. However, no convergence result is proved for this algorithm, but it is illustrated on some examples.

1.1. Outline.

The paper is organized as follow: In section 2 we state some of the results proved by Martínez [16] which will be of use for our assumptions and analysis. The beginning of section 3 gives some insights of the ideas that are used in the two methods we propose for computing the local-nonglobal minimizer of problem (1.2). We recall the parametric eigenvalue problem that is used in the algorithm of Rendl and Wolkowicz and extend the ideas behind their algorithm to our needs. We also recast problem (1.2) in terms of a new parametric eigenvalue problem, where we make use of extremal ellipsoids. Each of these two reformulations of problem (1.2) induces an algorithm for computing a local-nonglobal minimizer. In section 3.1, we give a geometric view of some of the results obtained by Martínez [16] on global and local-nonglobal minimizers when the trust-region radius is taken to infinity. In sections 3.2 and 3.3, based on our new parametric eigenvalue problem and the one introduced by Rendl and Wolkowicz, we derive two algorithms for computing a local-nonglobal minimizer. In each of these two sections, convergence results are given and the local-nonglobal minimizer is related to a local minimizer of some well-chosen function. We give concluding remarks in section 4 and preliminary numerical results obtained by comparing the two algorithms developed in this paper. The paper mainly focuses on the local-nonglobal minimizer of problem (1.2), but we indicate in an appendix how one can compute a local-nonglobal minimizer of problem (1.1).

1.2. Notation.

We will use the following standard notation throughout the paper. All norms are two-norms. The identity matrix is I. The space of $n \times m$ real matrices is denoted by $\mathbb{M}^{n,m}$. For $M \in \mathbb{M}^{n,m}$, we denote its transpose by M^T. If $n = m$, we denote its inverse by M^{-1} and its determinant by $\det(M)$. Given an $n \times n$ symmetric matrix S, $\lambda_j(S)$ denotes the jth smallest eigenvalue of S, where $1 \leq j \leq n$. Thus $\lambda_1(S) \leq \lambda_2(S) \leq \cdots \leq \lambda_n(S)$. If S is positive semidefinite (definite), we use $S \succeq 0$ ($S \succ 0$). If $S \succeq 0$, we denote its square root by $S^{1/2}$ and the inverse of the latter matrix by $S^{-1/2}$. When we write $x \searrow a$, we mean x converges to a and $x > a$. Similarly, $x \nearrow a$ means x converges to a and $x < a$. For a function $f : \mathbb{R} \mapsto \mathbb{R}$, we define

$$f(a^+) := \lim_{x \searrow a} f(x), \qquad f(a^-) := \lim_{x \nearrow a} f(x).$$

2. Background results.

We start by surveying the work of Martínez [16]. Hence, the reader is referred to this paper for the corresponding proofs of the lemmas and theorems which appear in this section.

The first theorem states the classical necessary optimality conditions for local minimizers of problem (1.2).

THEOREM 2.1. *Assume that x^* is a local minimizer of* (1.2). *Then there exists a unique Lagrange multiplier $\lambda^* \in \mathbb{R}$ such that*

$$(A - \lambda^* I)x^* = a \tag{2.1}$$

and

$$w^T(A - \lambda^* I)w \geq 0 \tag{2.2}$$

for all $w \in \mathbb{R}^n$ such that $w^T x^* = 0$. □

It is well known, if x^* is a global minimizer of problem (1.2), that its Lagrange multiplier λ^* lies in the interval $(-\infty, \lambda_1(A)]$; see, e.g., [7, 23]. There exist bounds on the Lagrange multiplier of a local-nonglobal minimizer which depend as well on eigenvalues of A.

LEMMA 2.2. *If x^* is a local-nonglobal minimizer of* (1.2)*, then* (2.1) *holds with* $\lambda^* \in (\lambda_1(A), \lambda_2(A))$. □

Global minimizers of problem (1.2) always exist, since a continuous function is minimized over a compact set. However, it is not always the case that a local-nonglobal minimizer exists for this problem. In particular, we have the two following cases for which no such point exists. The first case is an obvious consequence of the previous lemma.

COROLLARY 2.3. *If $\lambda_1(A) = \lambda_2(A)$, then there are no local-nonglobal minimizer of problem* (1.2). □

LEMMA 2.4. *If a is orthogonal to an eigenvector of A for the eigenvalue $\lambda_1(A)$, then there are no local-nonglobal minimizer of problem* (1.2). □

The situation where a is orthogonal to all eigenvectors of A for the eigenvalue $\lambda_1(A)$ is commonly referred as the *hard case* in the literature concerned with computing a global minimizer of problem (1.2). However, the previous lemma states this cannot happen if a local-nonglobal minimizer exists.

Define, for $\lambda \in (\lambda_1(A), \lambda_2(A))$,

$$\varphi(\lambda) := \|(A - \lambda I)^{-1} a\|^2. \tag{2.3}$$

Let

$$A = QDQ^T \tag{2.4}$$

be an orthonormal diagonalization of A; i.e., the columns of Q are orthonormal eigenvectors of A and D is a diagonal matrix with the eigenvalues of A on its diagonal ordered increasingly such that $D_{11} = \lambda_1(A)$. Also let

$$\bar{a} := Q^T a. \tag{2.5}$$

The function φ and its derivatives are given by the following formulas:

$$\varphi(\lambda) = \sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i(A) - \lambda)^2}, \tag{2.6a}$$

$$\varphi'(\lambda) = 2\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i(A) - \lambda)^3}, \tag{2.6b}$$

$$\varphi''(\lambda) = 6\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i(A) - \lambda)^4}. \tag{2.6c}$$

Suppose $\lambda_1(A) < \lambda_2(A)$ and $\bar{a}_1 \neq 0$. Note from (2.6c) that φ is a strictly convex function over the interval $(\lambda_1(A), \lambda_2(A))$. Therefore the equation $\varphi(\lambda) = 1$ has at most two roots in $(\lambda_1(A), \lambda_2(A))$, and the following theorem shows that in the case where two roots exist, only the smallest root may be the Lagrange multiplier of a local-nonglobal minimizer of problem (1.2).

THEOREM 2.5.

1. *If x^* is a local-nonglobal minimizer of* (1.2)*, then* (2.1) *holds with* $\lambda^* \in (\lambda_1(A), \lambda_2(A))$ *and* $\varphi'(\lambda^*) \leq 0$.

2. *There exists at most one local-nonglobal minimizer of* (1.2).

3. *If* $\|x^*\| = 1$, (2.1) *holds for some* $\lambda^* \in (\lambda_1(A), \lambda_2(A))$ *and* $\varphi'(\lambda^*) < 0$, *then* x^* *is a strict local-nonglobal minimizer of* (1.2). □

3. Characterizing and computing a local-nonglobal minimizer. The way we wish to compute a local-nonglobal minimizer is largely inspired by two approaches for computing a global minimizer. Therefore, we begin by partially describing these two approaches in order to give some insight into the ideas we shall use in the two upcoming algorithms for computing a local-nonglobal minimizer. On the basis of Corollary 2.3 and Lemma 2.4, we assume for the rest of this paper that the following assumptions hold.

Assumption 1. $\lambda_1(A) < \lambda_2(A)$.

Assumption 2. $\bar{a}_1 \neq 0$.

In a paper by Rendl and Wolkowicz [21], it is shown that a global minimizer x^* of problem (1.2) may be obtained if we solve the following linear semidefinite program:

$$\begin{aligned} \max_{t\in\mathbb{R},\lambda\in\mathbb{R}} \quad & 2\lambda - t, \\ & D(t) - \lambda \succeq 0, \end{aligned} \tag{3.1}$$

where

$$D(t) := \begin{bmatrix} t & -a^T \\ -a & A \end{bmatrix}.$$

The latter problem is equivalent to

$$\max_{t\in\mathbb{R}} \quad k(t) := 2\lambda_1(D(t)) - t, \tag{3.2}$$

where k is shown to be a concave function. In the case where Assumption 2 holds, it is further shown that $\lambda_1(D(t))$ has multiplicity one, and thus k is differentiable. Thus, the optimal t^* for problem (3.2) may be obtained by solving the equation $k'(t) = 0$, which can be expended as (see, e.g., [12])

$$2\tilde{y}_0(t)^2 - 1 = 0, \tag{3.3}$$

where $\tilde{y}_0(t)$ is the first component of a unit eigenvector for the eigenvalue $\lambda_1(D(t))$. (From now on we shall omit the dependence of $\tilde{y}_0$ on t for conciseness since it is clear from the context.) In other words, if $\tilde{y}$ is a unit eigenvector of $\lambda_1(D(t))$, then $\tilde{y} = (\tilde{y}_0, \tilde{z})^T$, where $\tilde{y}_0 \in \mathbb{R}$ and $\tilde{z} \in \mathbb{R}^n$.

Using the eigenvalue equation $D(t)\tilde{y} = \lambda_1(D(t))\tilde{y}$, we have

$$t\tilde{y}_0 - a^T\tilde{z} = \lambda_1(D(t))\tilde{y}_0, \tag{3.4a}$$

$$-\tilde{y}_0 a + A\tilde{z} = \lambda_1(D(t))\tilde{z}. \tag{3.4b}$$

From Assumption 2, for all $t \in \mathbb{R}$, [21] shows that $\tilde{y}_0 \neq 0$. If we let $\tilde{x}(t) = 1/\tilde{y}_0\, \tilde{z}$, then, using (3.4b), the stationarity equation

$$(A - \lambda_1(D(t))I)\tilde{x}(t) = a \tag{3.5}$$

is satisfied. Rendl and Wolkowicz [21] show a global minimizer x^* may be obtained by setting $x^* = \tilde{x}(t^*)$. Note that since $\|\tilde{z}\|^2 = 1 - \tilde{y}_0^2$, solving (3.3) or $\|\tilde{x}(t)\|^2 = 1$ is equivalent. The corresponding Lagrange multiplier for the global minimizer is obtained from

$$\lambda^* = \lambda_1(D(t^*)). \tag{3.6}$$

As mentioned in section 2, the Lagrange multiplier λ^* of a global minimizer satisfies $\lambda^* \in (-\infty, \lambda_1(A)]$. It is not surprising that this property is preserved through (3.6). Indeed, since the matrix $D(t)$ is of dimension $n+1$ and is simply the matrix A to which a new row and column has been added, the fact that the eigenvalues of A interlace those of $D(t)$ is known as Cauchy's inequalities [4] or the interlacing eigenvalues theorem for bordered matrices. It is formally stated in the next lemma, and a proof may be found in [29].

Lemma 3.1. *For $t \in \mathbb{R}$,*

$$\lambda_1(D(t)) \leq \lambda_1(A) \leq \lambda_2(D(t)) \leq \lambda_2(A) \leq \cdots \leq \lambda_n(D(t)) \leq \lambda_n(A) \leq \lambda_{n+1}(D(t)). \quad \square$$

Let us summarize what we presented so far from the Rendl–Wolkowicz [21] paper: In order to solve problem (3.2), the optimal t^* is obtained by solving $\|\tilde{x}(t)\|^2 = 1$. The corresponding global minimizer for problem (1.2) is then $x^* = \tilde{x}(t^*)$ and its corresponding Lagrange multiplier, $\lambda^* = \lambda_1(D(t^*))$, is guaranteed by Lemma 3.1 to satisfy $\lambda^* \leq \lambda_1(A)$.

Now the key observation to extend this procedure in order to compute a local-nonglobal minimizer is that, according to Lemmas 2.2 and 3.1, the eigenvalue $\lambda_2(D(t))$ lies precisely in the interval where the Lagrange multiplier for a local-nonglobal minimizer may be found. The algorithm we present in section 3.3 is based on the following idea. For $t \in \mathbb{R}$, compute a unit eigenvector (which depends on t) $y = (y_0, z)^T$ for the eigenvalue $\lambda_2(D(t))$. Let $x(t) = 1/y_0 \; z$ (assuming $y_0 \neq 0$) and notice that, analogously to equation (3.5), $x(t)$ satisfies the stationarity equation

$$(A - \lambda_2(D(t))I)x(t) = a. \tag{3.7}$$

Finally solve $\|x(t)\|^2 = 1$. Since this equation is equivalent to setting the first derivative of the function $m(t) := 2\lambda_2(D(t)) - t$ to zero, we shall also be interested in this function. Since from the optimum of $k(t)$ one may obtain a global minimizer of problem (1.2), we shall see in section 3.3 there is a link between an optimum of the function $m(t)$ and a local-nonglobal minimizer of problem (1.2).

Let us now look at another approach for computing a global minimizer of problem (1.2) which may be extended again to compute a local-nonglobal minimizer. We use the feasibility constraint in (1.2) to obtain another equality-constrained trust-region subproblem which has the same optimal solutions, but a strictly convex objective with ellipsoidal level curves centered in the interior of the unit ball. Precisely, let

$$\bar{\lambda} := \lambda_1(A) - \|a\|, \tag{3.8a}$$

$$B := A - \bar{\lambda}I. \tag{3.8b}$$

Note from (2.4) that $B = Q^T(D - \bar{\lambda}I)Q$. Now observe from Assumptions 1 and 2 and definition (2.5) that

$$B \succ 0, \tag{3.9a}$$

$$\|B^{-1}a\|^2 = \sum_{j=1}^{n} \frac{(\bar{a}_j)^2}{(\lambda_i(A) - \bar{\lambda})^2} < \frac{1}{(\lambda_1(A) - \bar{\lambda})^2} \sum_{j=1}^{n} (\bar{a}_j)^2 = \frac{1}{\|a\|^2}\|a\|^2 = 1. \tag{3.9b}$$

For x feasible for (1.2), $x^TAx - 2a^Tx = x^TBx - 2a^Tx + \bar{\lambda}$, and, by completing the square, $x^TAx - 2a^Tx = (x - B^{-1}a)^TB(x - B^{-1}a) + \bar{\lambda} - a^TB^{-1}a$. Therefore, (1.2) and the following problem share the same optimal solutions:

$$\begin{aligned} r_G^2 := \min \;\; & r^2(x) := (x - B^{-1}a)^TB(x - B^{-1}a) \\ \text{s.t. } & \|x\| = 1. \end{aligned} \tag{3.10}$$

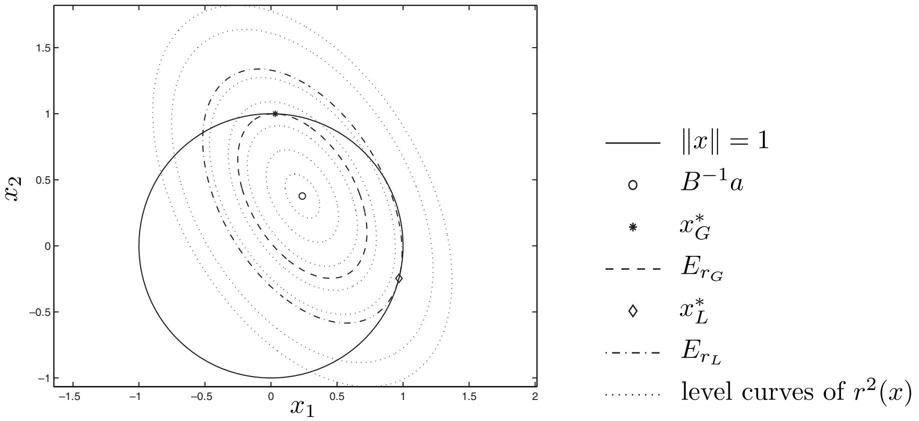


FIG. 3.1. *This figure shows the local minimizers of problem* (3.10). *The global minimizer x_G^* intersects the unit sphere and E_{r_G}, the largest volume ellipsoid E_r contained in the unit ball. The local-nonglobal minimizer x_L^* intersects the unit sphere and the ellipsoid E_{r_L} which is locally contained in the unit ball at x_L^*.*

The level curves $r^2(x) = r^2$, for r a fixed nonnegative constant, are the boundaries of the ellipsoids E_r, where

$$E_r := \{x = rB^{-1/2}u + B^{-1}a \mid \|u\| \leq 1\}. \tag{3.11}$$

The volume of each of these ellipsoids, where the unit is taken to be the volume of the unit ball in $\mathbb{R}^n$, is the determinant of $rB^{-1/2}$ (see, e.g., [2]) and their center, $B^{-1}a$, is by (3.9b) in the interior of the unit ball. Therefore, problem (3.10) is equivalent to finding the largest volume ellipsoid, say E_{r_G}, among the ellipsoids E_r contained in the unit ball. Equivalence is in the sense that E_{r_G} intersects the unit sphere at a global minimizer x_G^* for (1.2). Similarly, a local-nonglobal minimizer x_L^* of problem (1.2) should be within the intersection of an ellipsoid E_{r_L} and the unit sphere for some $r_L > r_G$. This ellipsoid is not contained in the unit ball, but points which are elements of E_{r_L} and close enough to x_L^* are contained in the unit ball. This is illustrated in Figure 3.1.

The constraint imposed on E_r to be contained in the unit ball may be modeled by a semidefinite constraint as indicated by the following lemma [2, 3].

LEMMA 3.2. *An ellipsoid*

$$E = E(Z, z) := \{x = Zu + z \mid \|u\| \leq 1\}, \qquad Z \in \mathbb{M}^{n,m},$$

is contained in the ellipsoid

$$W = W(Y, y) := \{x \mid (x-y)^T YY^T (x-y) \leq 1\}, \qquad Y \in \mathbb{M}^{n,n},\ \det(Y) \neq 0,$$

if and only if there exists γ such that

$$\begin{pmatrix} I & Y(z-y) & YZ \\ (z-y)^T Y^T & 1-\gamma & 0 \\ Z^T Y^T & 0 & \gamma I \end{pmatrix} \succeq 0. \qquad \square \tag{3.12}$$

Applying Lemma 3.2 with $Z = rB^{-1/2}$, $z = B^{-1}a$, $Y = I$, and $y = 0$, and multiplying the matrix in (3.12) from the right and the left by a well-chosen block diagonal matrix yields the linear semidefinite program

$$\begin{aligned} r_G = \quad & \max_{r,\gamma} \; r \\ & \text{s.t.} \begin{pmatrix} I & B^{-1}a & rI \\ a^TB^{-1} & 1-\gamma & 0 \\ rI & 0 & \gamma B \end{pmatrix} \succeq 0, \end{aligned} \tag{3.13}$$

whose optimal value is r_G. Notice that this semidefinite program is different from the usual semidefinite relaxation associated with problem (1.2). Furthermore, as in [21], the semidefinite program (3.13) may be solved implicitly by maximizing a function of one variable. To show this, we will need the following lemma on the Schur complement which is a consequence of Sylvester's law of inertia (see, e.g., [2]).

Lemma 3.3. *Let*

$$M = \begin{pmatrix} N & C^T \\ C & D \end{pmatrix}$$

be a symmetric matrix with $k \times k$ block N and $g \times g$ block D. Assume that N is positive definite. Then $M \succeq 0$ ($\succ 0$) if and only if the matrix $D - CN^{-1}C^T \succeq 0$ ($\succ 0$). □

By feasibility of (3.13), the top left square matrix of size $n+1$ is positive semidefinite, and applying Lemma 3.3 with $N = I$ gives $\gamma \leq 1 - \|B^{-1}a\|^2 < 1$ (note that the leftmost inequality follows from Assumption 2). Furthermore, $\gamma \geq 0$ must hold for the constraint of problem (3.13) to be satisfied. Note that the constraint yields that if $\gamma = 0$, then $r = 0$. Hence we may only consider $\gamma > 0$ in problem (3.13), since we know from the fact that the ellipsoids (3.11) are centered in the interior of the unit ball that $r_G > 0$. Lemma 3.3 may again be used to gain further information by applying it to the full constraint matrix in (3.13), with this time N equal to the bottom right matrix of size $n+1$. This gives

$$I - \frac{r^2}{\gamma}B^{-1} - \frac{1}{1-\gamma}B^{-1}aa^TB^{-1} \succeq 0. \tag{3.14}$$

Multiplying the left-hand side of (3.14) from the left and the right by $(\gamma B)^{1/2}$ yields $r \leq \sqrt{\gamma\lambda_1(B(\gamma))}$, where

$$B(\gamma) := B - \frac{1}{1-\gamma}B^{-1/2}aa^TB^{-1/2}. \tag{3.15}$$

Hence, for a fixed $0 < \gamma < 1$, the largest possible r for which the matrix in (3.13) stays positive semidefinite is $r = \sqrt{\gamma\lambda_1(B(\gamma))}$. Thus to solve (3.13), one needs to find γ^*, which solves

$$\begin{aligned} r_G^2 = \max \; & f(\gamma) = \gamma\lambda_1(B(\gamma)) \\ \text{s.t. } & 0 < \gamma < 1. \end{aligned} \tag{3.16}$$

Note so far that this second approach is in structure similar to the approach of Rendl and Wolkowicz: from a linear semidefinite program, we have obtained an equivalent maximization problem of one variable. There are more links between the two approaches, one of them being that the eigenvalues of $B(\gamma)$ interlace those of B.

This result is a corollary of Cauchy's inequality, Lemma 3.1; a proof may be found in [29].

COROLLARY 3.4. *For $\gamma \neq 1$, the eigenvalues of $B(\gamma)$ and B interlace.*

1. *For $\gamma > 1$,*

$$\lambda_1(B) \leq \lambda_1(B(\gamma)) \leq \lambda_2(B) \leq \lambda_2(B(\gamma)) \leq \cdots \leq \lambda_n(B) \leq \lambda_n(B(\gamma)).$$

2. *For $\gamma < 1$,*

$$\lambda_1(B(\gamma)) \leq \lambda_1(B) \leq \lambda_2(B(\gamma)) \leq \lambda_2(B) \leq \cdots \leq \lambda_n(B(\gamma)) \leq \lambda_n(B). \qquad \square$$

Let

$$\lambda(\gamma) := \begin{cases} \lambda_2(B(\gamma)) & \text{for } \gamma < 1, \\ \lambda_1(B(\gamma)) & \text{for } \gamma > 1. \end{cases}$$

Using definition (3.8b), we observe from Corollary 3.4 that $\lambda(\gamma) + \bar{\lambda} \in [\lambda_1(A), \lambda_2(A)]$. Furthermore, if $B^{1/2}v$ is an eigenvector of the matrix $B(\gamma)$ (note that v depends on γ, but we write v instead of $v(\gamma)$ for conciseness) and $\lambda(\gamma)$ its corresponding eigenvalue, then it is easy to see they satisfy the generalized eigenvalue equation

$$\left(B^2 - \frac{1}{1-\gamma} aa^T\right) v = \lambda(\gamma) Bv. \tag{3.17}$$

Let us assume for now that $\lambda(\gamma)$ has multiplicity one and $a^T v \neq 0$ (these assumptions are justified in section 3.2). If we define

$$x(\gamma) := \frac{1-\gamma}{a^T v} Bv$$

and use (3.8b) and (3.17), then $x(\gamma)$ satisfies the stationarity equation

$$(A - (\lambda(\gamma) + \bar{\lambda})I)x(\gamma) = a. \tag{3.18}$$

Notice in the previous equation that if we replace $x(\gamma)$ by $x(t)$ and $\lambda(\gamma) + \bar{\lambda}$ by $\lambda_2(D(t))$ (both quantities lie in the interval $[\lambda_1(A), \lambda_2(A)]$ where the Lagrange multiplier of a local-nonglobal minimizer lies), we obtain (3.7). Thus it is not without surprise that the method we present in this case for computing a local-nonglobal minimizer relies on solving the equation $\|x(\gamma)\|^2 = 1$. We shall also be interested in the function $g(\gamma) = \gamma\lambda(\gamma)$ obtained by replacing $\lambda_1(B(\gamma))$ by $\lambda(\gamma)$ in $f(\gamma)$. This is analogous to the fact that one obtains $m(t)$ by replacing $\lambda_1(D(t))$ by $\lambda_2(D(t))$ in $k(t)$. As it is the case for the function $m(t)$, we will see there is a link between a local minimizer of the function $g(\gamma)$ and a local-nonglobal minimizer of problem (1.2).

Section 3 is divided into three smaller sections. The last two sections, section 3.2 and section 3.3, describe the two algorithms for computing a local-nonglobal minimizer for which we have just given an insight on the functions to be considered. We begin, however, in section 3.1 by presenting a geometric view of the limiting behavior of local minimizers of (1.2) when $\Delta \to \infty$ and recovering a result proved in Martínez [16].

3.1. Local optimum of TRS for infinitely large trust regions. Consider problem (1.2) where Δ is now any positive number. We wish to investigate the limiting behavior of local (global and nonglobal) minimizers of problem (1.2) as $\Delta \to \infty$.

Obviously, x^* is a local minimizer of problem (1.2) if and only if x^*/Δ is a local minimizer of the following problem:

$$\begin{array}{ll} \min\limits_x & x^T(\Delta^2 A)x - 2(\Delta a)^T x \\ \text{s.t.} & \|x\| = 1. \end{array} \tag{3.19}$$

Thus, equivalently, we investigate the limiting behavior of local minimizers of problem (3.19) as $\Delta \to \infty$.

In this new setting, define

$$\bar{\lambda}_\Delta := \Delta^2 \lambda_1(A) - \Delta\|a\|, \tag{3.20a}$$

$$B_\Delta := \Delta^2 A - \bar{\lambda}_\Delta I. \tag{3.20b}$$

Analogously to the inequalities (3.9), $B_\Delta \succ 0$ and $\|\Delta B_\Delta^{-1} a\| < 1$ hold.

Recall from (2.4) that $A = QDQ^T$. Let $Q = [q_1, \ldots, q_n]$, where q_i is the ith column of Q. If we define $x_G^*(\Delta)$ and $x_L^*(\Delta)$ to be, respectively, the global and the local-nonglobal minimizers of (3.19), then the following result has been shown.

Lemma 3.5 (Martínez [16]). *There exists* $\Delta_0 > 0$ *such that* (3.19) *admits a local-nonglobal minimizer for all* $\Delta > \Delta_0$ *and*

$$x_G^*(\infty) := \lim_{\Delta\to\infty} x_G^*(\Delta) = \frac{\bar{a}_1}{|\bar{a}_1|} q_1, \tag{3.21}$$

$$x_L^*(\infty) := \lim_{\Delta\to\infty} x_L^*(\Delta) = -\frac{\bar{a}_1}{|\bar{a}_1|} q_1. \tag{3.22}$$

Analogously to how we obtained problem (3.10) from problem (1.2), problem (3.19) shares the same optimal solution as

$$\begin{array}{ll} \min\limits_x & r_\Delta^2(x) := (x - \Delta B_\Delta^{-1} a)^T B_\Delta (x - \Delta B_\Delta^{-1} a) \\ \text{s.t.} & \|x\| = 1. \end{array} \tag{3.23}$$

We want to show from a geometric view of problem (3.23) how we recover the results of Lemma 3.5.

Define for $p > 0$ the set

$$\Omega_{p,\Delta} := \{x : (x - B_\Delta^{-1}\Delta a)^T B_\Delta (x - B_\Delta^{-1}\Delta a) \le p^2 \lambda_1(B_\Delta)\}.$$

This set is an ellipsoid centered in the interior of the unit ball and bounded by the level curve $r_\Delta^2 = p^2\lambda_1(B_\Delta)$. The choice of the latter constant is made to simplify the upcoming expressions. It follows that x^* is a local minimizer of problem (3.19) if and only if, for some $p > 0$, $x^* \in \Omega_{p,\Delta} \cap \{x : \|x\| = 1\}$ and, for some $\delta > 0$, $\Omega_{p,\Delta} \cap \{x : \|x - x^*\| \le \delta, \|x\| > 1\} = \emptyset$. This means x^* lies on the boundary of $\Omega_{p,\Delta}$, for some $p > 0$, which is locally contained in the unit ball and tangent to the unit sphere at x^*.

We have $B_\Delta = Q(\Delta^2 D - \bar{\lambda}_\Delta I)Q^T$, so that $B_\Delta = Q\Lambda Q^T$, where

$$\Lambda := \begin{pmatrix} \Delta\|a\| & & & 0 \\ & \Delta^2(\lambda_2(A) - \lambda_1(A)) + \Delta\|a\| & & \\ & & \ddots & \\ 0 & & & \Delta^2(\lambda_n(A) - \lambda_1(A)) + \Delta\|a\| \end{pmatrix}.$$

Note that $\lambda_i(B_\Delta) = (\Lambda)_{ii}$ for $i = 1, \ldots, n$. Now we may write $\Omega_{p,\Delta}$ as

$$\Omega_{p,\Delta} := \{x : (Q^T(x - B_\Delta^{-1}\Delta a))^T \Lambda (Q^T(x - B_\Delta^{-1}\Delta a)) \le p^2 \lambda_1(B_\Delta)\}.$$

Therefore

$$\Omega_{p,\Delta} = \left\{ x = Qz + \Delta B_\Delta^{-1} a : \frac{z_1^2}{p^2} + \frac{z_2^2}{\left(\frac{p^2\lambda_1(B_\Delta)}{\lambda_2(B_\Delta)}\right)} + \cdots + \frac{z_n^2}{\left(\frac{p^2\lambda_1(B_\Delta)}{\lambda_n(B_\Delta)}\right)} \le 1 \right\}. \tag{3.24}$$

Now it follows from Assumption 1 that, for $i = 2, \ldots, n$,

$$\lim_{\Delta\to\infty} \frac{\lambda_1(B_\Delta)}{\lambda_i(B_\Delta)} = \lim_{\Delta\to\infty} \frac{\|a\|}{\|a\| + \Delta(\lambda_i(A) - \lambda_1(A))} = 0$$

and that

$$\begin{aligned}\lim_{\Delta\to\infty} \Delta B_\Delta^{-1} a &= \lim_{\Delta\to\infty} Q(\Delta\Lambda^{-1})\bar{a},\\ &= \lim_{\Delta\to\infty} \frac{\bar{a}_1}{\|a\|} q_1 + \frac{\bar{a}_2}{\Delta(\lambda_2(A) - \lambda_1(A)) + \|a\|} q_2 + \cdots \\ &+ \frac{\bar{a}_n}{\Delta(\lambda_n(A) - \lambda_1(A)) + \|a\|} q_n,\\ &= \frac{\bar{a}_1}{\|a\|} q_1. \end{aligned}\tag{3.25}$$

Hence, as Δ becomes large, the length of the $n-1$ smaller axis of the ellipsoid (3.24) tends to zero, the length of the larger axis tends to $2p$, and the center of the ellipsoid tends to $\frac{\bar{a}_1}{\|a\|} q_1$. In other words, as Δ becomes large, the ellipsoid (3.24) converges to the segment

$$\Omega_{p,\infty} := \left\{ x = Qz + \frac{\bar{a}_1}{\|a\|} q_1 : |z_1| \le p\,; z_i = 0 \ \text{ for } \ i = 2, \ldots, n \right\},$$

which can be rewritten as

$$\Omega_{p,\infty} = \left\{ x = \left(z_1 + \frac{\bar{a}_1}{\|a\|} \right) q_1 : |z_1| \le p \right\}. \tag{3.26}$$

Therefore, $x_G^*(\infty)$ and $x_L^*(\infty)$ are obtained from the intersection of the boundary of the unit sphere with an end point of a segment of the form $\Omega_{p,\infty}$. Hence we have to look for values of $p = |\alpha|$ such that

$$|\alpha + \bar{a}_1/\|a\|| = 1. \tag{3.27}$$

There are two values of α that satisfy (3.27):

$$\alpha_1 := 1 - \bar{a}_1/\|a\| \quad \text{and} \quad \alpha_2 := -1 - \bar{a}_1/\|a\|.$$

Now let

$$m := \min\{|\alpha_1|, |\alpha_2|\} = 1 - |\bar{a}_1|/\|a\|.$$

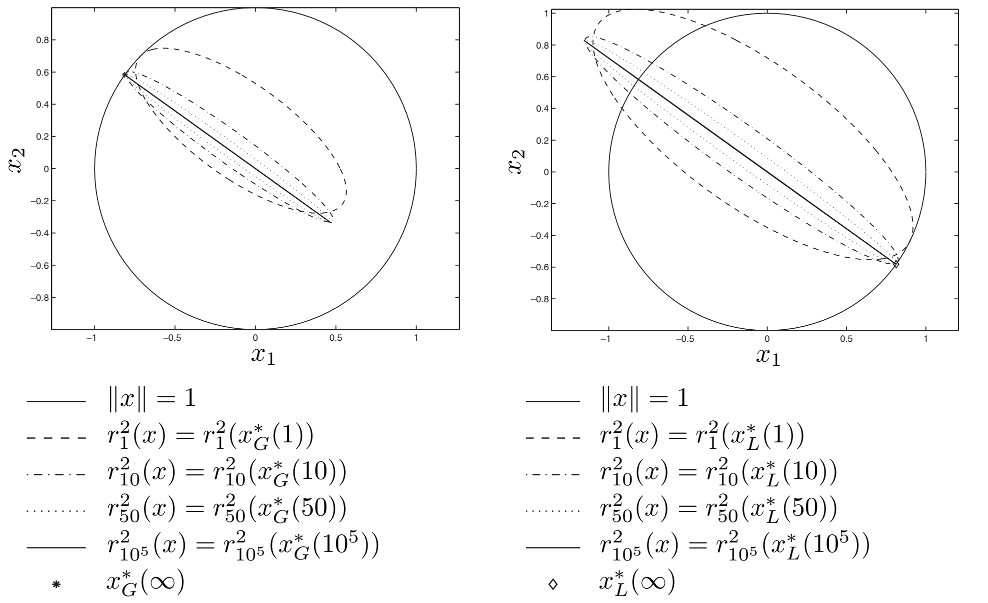


FIG. 3.2. *Limiting behavior as $\Delta \to \infty$ of the ellipsoids to which the points $x_G^*(\Delta)$ (on the left) and $x_L^*(\Delta)$ (on the right) belong.*

Therefore $x_G^*(\infty)$ is the intersection of the limiting level set $\Omega_{m,\infty}$ with the unit sphere. It follows that

$$x_G^*(\infty) = \begin{cases} q_1 & \text{if} \quad \bar{a}_1 \geq 0, \\ -q_1 & \text{if} \quad \bar{a}_1 < 0. \end{cases}$$

Thus (3.21) holds. Similarly, (3.22) holds, since for

$$M := \max\{|\alpha_1|, |\alpha_2|\} = 1 + |\bar{a}_1|/\|a\|,$$

we have that $x_L^*(\infty)$ is the intersection of the end points of the limiting level set $\Omega_{M,\infty}$ with the unit sphere.

For each value of Δ, there exists a value of p for which the global minimizer $x_G^*(\Delta)$ lies in the intersection of the unit sphere and the ellipsoid $\Omega_{p,\Delta}$. In Figure 3.2(left) we illustrate, as Δ varies, this sequence of ellipsoids $\Omega_{p,\Delta}$. One sees as Δ becomes large that the ellipsoids converge to the segment $\Omega_{m,\infty}$. Figure 3.2(right) illustrates the same concept, but this time, for different values of Δ, we plot the ellipses $\Omega_{p,\Delta}$ and their intersection with the unit sphere at the local-nonglobal minimizers $x_L^*(\Delta)$. In this case, the ellipses converge to the segment $\Omega_{M,\infty}$.

3.2. Computing a local-nonglobal minimizer: First method. We now consider our first algorithm for computing a local-nonglobal minimizer of problem (1.2). It is inspired by problem (3.16), where $B(\gamma)$ is defined in (3.15) and $\bar{\lambda}$ and B are defined in (3.8). As mentioned at the beginning of section 3, to derive our algorithm, we shall focus on the functions $\lambda(\gamma)$ and $\|x(\gamma)\|^2 - 1$. Our first lemma investigates the function $\lambda(\gamma)$.

Let

$$\mathcal{K} := \{k : \lambda_k(B) = \lambda_2(B)\}, \tag{3.28a}$$

$$\mathcal{J} := \{j : \bar{a}_j \neq 0\}, \tag{3.28b}$$

$$d(\lambda) := \sum_{j \in \mathcal{J}} \frac{\bar{a}_j^2}{\lambda_j(B)(\lambda_j(B) - \lambda)}.$$

Note that Assumptions 1 and 2 may be stated as $1 \in \mathcal{K}^c \cap \mathcal{J}$, where $\mathcal{K}^c$ is the complement of the set $\mathcal{K}$.

LEMMA 3.6. *Let*

$$\lambda(\gamma) := \begin{cases} \lambda_2(B(\gamma)) & \textit{for } \gamma < 1, \\ \hat{\lambda} & \textit{for } \gamma = 1, \\ \lambda_1(B(\gamma)) & \textit{for } \gamma > 1, \end{cases}$$

where $\hat{\lambda} = \lambda_2(B)$ if $\mathcal{K} \cap \mathcal{J} = \emptyset$ and $d(\lambda_2(B)) \leq 0$; otherwise, $\hat{\lambda}$ is the unique value in the interval $(\lambda_1(B), \lambda_2(B))$ to satisfy $d(\hat{\lambda}) = 0$.

1. *If $\mathcal{K} \cap \mathcal{J} \neq \emptyset$, then $\lambda(\gamma)$ is infinitely differentiable and satisfies $d(\lambda(\gamma)) = 1 - \gamma$. Moreover,*

$$\lambda'(\gamma) = \frac{-1}{d'(\lambda(\gamma))} \quad \textit{and} \quad \lambda''(\gamma) = \frac{-d''(\lambda(\gamma))}{[d'(\lambda(\gamma))]^3} \tag{3.29}$$

for all $\gamma \in \mathbb{R}$.

2. *If $\mathcal{K} \cap \mathcal{J} = \emptyset$, then $\lambda(\gamma)$ is continuous and infinitely differentiable for $\gamma \in \mathbb{R} \setminus \{1 - d(\lambda_2(B))\}$.*

(i) *For $\gamma > 1 - d(\lambda_2(B))$, $\lambda(\gamma)$ satisfies $d(\lambda(\gamma)) = 1 - \gamma$ and*

$$\lambda'(\gamma) = \frac{-1}{d'(\lambda(\gamma))} \quad \textit{and} \quad \lambda''(\gamma) = \frac{-d''(\lambda(\gamma))}{[d'(\lambda(\gamma))]^3}. \tag{3.30}$$

(ii) *For $\gamma < 1 - d(\lambda_2(B))$, $\lambda(\gamma) = \lambda_2(B)$, $\lambda'(\gamma) = 0$, and $\lambda''(\gamma) = 0$.*

(iii) *For $\gamma = 1 - d(\lambda_2(B))$, $\lambda(\gamma) = \lambda_2(B)$ and satisfies $d(\lambda(\gamma)) = 1 - \gamma$; the right- and left-hand side derivatives are given, respectively, by*

$$\lambda'(\gamma^+) = \frac{-1}{d'(\lambda_2(B))}, \quad \lambda''(\gamma^+) = \frac{-d''(\lambda_2(B))}{[d'(\lambda_2(B))]^3}, \tag{3.31a}$$

$$\lambda'(\gamma^-) = 0, \qquad \lambda''(\gamma^-) = 0. \tag{3.31b}$$

Proof. 1. For $\gamma \neq 1$, we have

$$\det(B(\gamma) - \lambda I) = \det\left((B - \lambda I)\left(I - \frac{1}{1-\gamma}(B - \lambda I)^{-1} B^{-1/2} a a^T B^{-1/2}\right)\right),$$

$$= \det(B - \lambda I)\left(1 - \frac{1}{1-\gamma} a^T B^{-1/2} (B - \lambda I)^{-1} B^{-1/2} a\right),$$

$$= \prod_{j=1}^{n} (\lambda_j(B) - \lambda)\left(1 - \frac{1}{1-\gamma} \sum_{j \in \mathcal{J}} \frac{\bar{a}_j^2}{\lambda_j(B)(\lambda_j(B) - \lambda)}\right), \tag{3.32a}$$

$$= \prod_{j=1}^{n} (\lambda_j(B) - \lambda)\left(1 - \frac{1}{1-\gamma} d(\lambda)\right), \tag{3.32b}$$

where the second equality follows from Golub and Van Loan [8]. Since $\mathcal{K}\cap\mathcal{J}\neq\emptyset$ and $\bar{a}_1\neq 0$,

$$\lim_{\lambda\searrow\lambda_1(B)} d(\lambda)=-\infty \quad\text{and}\quad \lim_{\lambda\nearrow\lambda_2(B)} d(\lambda)=\infty.$$

Furthermore,

$$d'(\lambda)=\sum_{j\in\mathcal{J}}\frac{\bar{a}_j^2}{\lambda_j(B)(\lambda_j(B)-\lambda)^2}>0.$$

Therefore, for all $\gamma\in\mathbb{R}$, $d^{-1}(1-\gamma)$ is well defined, where $d^{-1}(1-\gamma)\in(\lambda_1(B),\lambda_2(B))$. Moreover, (3.32b) shows it is an eigenvalue of $B(\gamma)$. From Corollary 3.4, this shows

$$d^{-1}(1-\gamma)=\begin{cases}\lambda_2(B(\gamma)) & \text{for } \gamma<1,\\ \lambda_1(B(\gamma)) & \text{for } \gamma>1.\end{cases}$$

Hence, $\lambda(\gamma)=d^{-1}(1-\gamma)$ and is infinitely differentiable. Equations (3.29) are obtained by implicit differentiation.

2. Since $\mathcal{K}\cap\mathcal{J}=\emptyset$, then $d(\lambda_2(B))$ is well defined. Let $\gamma\neq 1$. By (3.32b), $\lambda_2(B)$ is an eigenvalue of $B(\gamma)$. Also, Assumptions 1 and 2 imply that $\lambda_1(B)$ is not an eigenvalue of $B(\gamma)$, since from (3.32a) we obtain

$$\lim_{\lambda\to\lambda_1(B)}\det(B(\gamma)-\lambda I)=-\prod_{j=2}^{n}(\lambda_j(B)-\lambda_1(B))\left(\frac{\bar{a}_1^2}{(1-\gamma)\lambda_1(B)}\right)\neq 0.$$

Note again that $d(\lambda)$ is strictly increasing for $\lambda\in(\lambda_1(B),\lambda_2(B)]$ and therefore

$$d(\lambda)<d(\lambda_2(B)) \quad\text{for}\quad \lambda\in(\lambda_1(B),\lambda_2(B)). \tag{3.33}$$

(i) If $1-\gamma<d(\lambda_2(B))$, then $d^{-1}(1-\gamma)$ is well defined, where $d^{-1}(1-\gamma)\in(\lambda_1(B),\lambda_2(B))$. The rest of the proof is similar to the proof of item 1.

(ii) and (iii) If $1-\gamma\geq d(\lambda_2(B))$, then by (3.33), $d(\lambda)<1-\gamma$ for $\lambda\in(\lambda_1(B),\lambda_2(B))$. Therefore, there are no eigenvalues of the matrix $B(\gamma)$ in the interval $[\lambda_1(B),\lambda_2(B))$ and, from Corollary 3.4,

$$\lambda_2(B)=\begin{cases}\lambda_2(B(\gamma)) & \text{for } \gamma<1,\\ \hat{\lambda} & \text{for } \gamma=1,\\ \lambda_1(B(\gamma)) & \text{for } \gamma>1.\end{cases}$$

Thus $\lambda(\gamma)=\lambda_2(B)$. In particular, the derivatives of $\lambda(\gamma)$ for $\gamma<1-d(\lambda_2(B))$ are zero, and equations (3.31b) hold. Note finally that $1-\gamma=d(\lambda(\gamma))$ for $\gamma\geq 1-d(\lambda_2(B))$, and thus (3.31a) holds. □

COROLLARY 3.7. *For $\gamma\in\mathbb{R}$, $\lambda(\gamma)>\lambda_1(B)$ and $\lim_{\gamma\to\infty}\lambda(\gamma)=\lambda_1(B)$. Moreover,*

1. *if $\mathcal{K}\cap\mathcal{J}\neq\emptyset$, then $\lambda(\gamma)<\lambda_2(B)$ and $\lim_{\gamma\to-\infty}\lambda(\gamma)=\lambda_2(B)$;*
2. *if $\mathcal{K}\cap\mathcal{J}=\emptyset$, then*
   (i) *$\lambda(\gamma)=\lambda_2(B)$ for $\gamma\leq 1-d(\lambda_2(B))$,*
   (ii) *$\lambda(\gamma)<\lambda_2(B)$ for $\gamma>1-d(\lambda_2(B))$.* □

Let

$$\Gamma:=\begin{cases}\mathbb{R} & \text{if } \mathcal{K}\cap\mathcal{J}\neq\emptyset,\\ (1-d(\lambda_2(B),\infty) & \text{if } \mathcal{K}\cap\mathcal{J}=\emptyset.\end{cases}$$

Note from Lemma 3.6 and Corollary 3.7 that

$$\Gamma = \{\gamma : \lambda(\gamma) \in (\lambda_1(B), \lambda_2(B))\}, \tag{3.34}$$

$$\lambda'(\gamma) < 0 \quad \text{for } \gamma \in \Gamma. \tag{3.35}$$

For $\gamma \in \Gamma$, define

$$x(\gamma) := \begin{cases} \frac{1-\gamma}{a^T v} Bv & \text{if } \gamma \neq 1, \\ (A - (\lambda(1) + \bar{\lambda})I)^{-1} a & \text{if } \gamma = 1, \end{cases} \tag{3.36}$$

where v is any vector that satisfies the generalized eigenvalue equation (3.17). Corollary 3.4 and (3.34) imply, for $\gamma \neq 1$, that $\lambda(\gamma)$ is an eigenvalue of $B(\gamma)$ of multiplicity one. Furthermore, $a^T v \neq 0$; otherwise this would imply that $\lambda(\gamma)$ is an eigenvalue of B. Hence $x(\gamma)$ is well defined and (3.18) holds. For $\gamma \in \Gamma$, it follows from (3.8b) and (3.34) that $\lambda(\gamma) + \bar{\lambda} \in (\lambda_1(A), \lambda_2(A))$. Therefore $A - (\lambda(\gamma) + \bar{\lambda})I$ is invertible, and from (2.3), (3.8b), and (3.18) we may write $\|x(\gamma)\|^2$ as

$$\|x(\gamma)\|^2 = \varphi(\lambda(\gamma) + \bar{\lambda}). \tag{3.37}$$

Hence

$$\frac{d\|x(\gamma)\|^2}{d\gamma} = \frac{d\varphi(\lambda(\gamma) + \bar{\lambda})}{d\lambda} \lambda'(\gamma). \tag{3.38}$$

The algorithm of Martínez [16] which is used to compute a local-nonglobal minimizer of problem (1.2) finds a root of the function $\varphi(\lambda) - 1$ in the interval $(\lambda_1(A), \lambda_2(A))$. As we shall see in section 3.2.2, our algorithm finds a root of the function $\|x(\gamma)\|^2 - 1$ in the interval Γ. We may immediately derive the equivalent of Theorem 2.5.

Theorem 3.8.

1. *If x^* is a local-nonglobal minimizer of problem* (1.2)*, then* (2.1) *holds with $\lambda^* \in (\lambda_1(A), \lambda_2(A))$. Let γ^* be the unique solution to $\lambda(\gamma) + \bar{\lambda} = \lambda^*$; then $x^* = x(\gamma^*)$ and $\frac{d\|x(\gamma^*)\|^2}{d\gamma} \geq 0$.*

2. *If, for $\gamma^* \in \Gamma$, $\|x(\gamma^*)\| = 1$ and $\frac{d\|x(\gamma^*)\|^2}{d\gamma} > 0$, then $x(\gamma^*)$ is a strict local-nonglobal minimizer of* (1.2)*.*

3. *For $\gamma \in \Gamma \cap \{\gamma : \frac{d\|x(\gamma)\|^2}{d\gamma} > 0\}$, $x(\gamma)$ is a strict local-nonglobal minimizer of*

$$\begin{array}{ll} \min\limits_{x} & x^T A x - 2a^T x \\ s.t. & \|x\| = \|x(\gamma)\| \end{array} \tag{3.39}$$

with Lagrange multiplier $\lambda(\gamma) + \bar{\lambda}$.

Proof. Since $\gamma^* \in \Gamma$, the proofs of items 1 and 2 follow from Theorem 2.5 and (3.35), (3.37), and (3.38). To prove item 3, fix $\gamma \in \Gamma$ and let $\delta := \|x(\gamma)\|$. Note that $x(\gamma)$ is a local-nonglobal minimizer of problem (3.39) if and only if $x(\gamma; \delta) := x(\gamma)/\delta$ is a local-nonglobal minimizer of

$$\begin{array}{ll} \min\limits_{x} & x^T (\delta^2 A) x - 2(\delta a)^T x \\ \text{s.t.} & \|x\| = 1. \end{array} \tag{3.40}$$

Now we may write $\varphi(\lambda; \delta) := \|((\delta^2 A) - \lambda I)^{-1} (\delta a)\|^2$ as

$$\varphi(\lambda;\delta) = \sum_{i=1}^{n} \frac{\delta^2 \bar{a}_i^2}{(\delta^2\lambda_i(A) - \lambda)^2} \tag{3.41}$$

and its derivative as

$$\varphi'(\lambda;\delta) = 2\sum_{i=1}^{n} \frac{\delta^2 \bar{a}_i^2}{(\delta^2\lambda_i(A) - \lambda)^3}.$$

For $\gamma \in \Gamma \cap \{\gamma : \frac{d\|x(\gamma)\|^2}{d\gamma} > 0\}$, it follows from (3.35) and (3.38) that

$$\frac{d\varphi(\lambda(\gamma)+\bar{\lambda})}{d\lambda} < 0. \tag{3.42}$$

By item 3 of Theorem 2.5, $x(\gamma;\delta)$ is a local-nonglobal minimizer of problem (3.40), since

$$((\delta^2 A) - (\delta^2(\lambda(\gamma)+\bar{\lambda}))I)x(\gamma;\delta) = \delta a,$$

since $\lambda_1(\delta^2 A) < \delta^2(\lambda(\gamma)+\bar{\lambda}) < \lambda_2(\delta^2 A)$, and since, from (3.42),

$$\varphi'(\delta^2(\lambda(\gamma)+\bar{\lambda});\delta) = \varphi'(\lambda(\gamma)+\bar{\lambda}) < 0. \qquad \square$$

From (2.6c) we easily see that φ is a strictly convex function on the open interval $(\lambda_1(A), \lambda_2(A))$. The algorithm of Martínez [16] takes advantage of this property. The following lemma shows that $\|x(\gamma)\|^2$ is also strictly convex over Γ. Convexity will play a main role in the convergence analysis of our algorithm, since the secant method will be used to find a root of the function $\|x(\gamma)\|^2 - 1$.

Lemma 3.9. *Consider the function $\|x(\gamma)\|^2$ with domain Γ. Then it is an infinitely differentiable strictly convex function and $\lim_{\gamma\to\infty} \|x(\gamma)\|^2 = \infty$.*

Proof. Since $\varphi(\lambda)$ and $\lambda(\gamma)$ are infinitely differentiable, respectively, on the intervals $(\lambda_1(A), \lambda_2(A))$ and Γ, and $\lambda(\gamma) + \bar{\lambda} \in (\lambda_1(A), \lambda_2(A))$, then infinite differentiability follows from (3.37). By Corollary 3.7, $\lim_{\gamma\to\infty} \lambda(\gamma) + \bar{\lambda} = \lambda_1(A)$ and $\lambda(\gamma) + \bar{\lambda} > \lambda_1(A)$, and, by Assumption 2, $\lim_{\lambda\searrow\lambda_1(A)} \varphi(\lambda) = \infty$. Thus, using (3.37), $\lim_{\gamma\to\infty} \|x(\gamma)\|^2 = \infty$.

All that is left to prove is strict convexity. For simplicity, let $\lambda_i = \lambda_i(B)$ for $i = 1, \ldots, n$ and let $\lambda_\gamma = \lambda(\gamma)$. There are two cases to consider.

Case 1. *$\bar{a}_1 \neq 0$ and $\bar{a}_j = 0$ for $j = 2, \ldots, n$.* We have in this case

$$\begin{aligned} \|x(\gamma)\|^2 &= \frac{\bar{a}_1^2}{(\lambda_1 - \lambda_\gamma)^2}, \\ \frac{d\|x(\gamma)\|^2}{d\gamma} &= \frac{2\bar{a}_1^2}{(\lambda_1 - \lambda_\gamma)^3}\lambda'(\gamma) = -\frac{2\lambda_1}{\lambda_1 - \lambda_\gamma}, \end{aligned}$$

where we have used (3.29) and (3.30) to obtain the first derivative. Thus, using (3.35),

$$\frac{d^2\|x(\gamma)\|^2}{d\gamma^2} = -\frac{2\lambda_1}{(\lambda_1 - \lambda_\gamma)^2}\lambda'(\gamma) > 0.$$

Case 2. *$\exists j \geq 2$ such that $\bar{a}_1\bar{a}_j \neq 0$.* We have, using once again (3.29) and (3.30),

$$\|x(\gamma)\|^2 = \sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda_\gamma)^2},$$

$$\frac{d\|x(\gamma)\|^2}{d\gamma} = 2\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda_\gamma)^3}\lambda'(\gamma) = \frac{-2\sum\limits_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i-\lambda_\gamma)^3}}{\sum\limits_{i=1}^{n} \frac{\bar{a}_i^2}{\lambda_i(\lambda_i-\lambda_\gamma)^2}},$$

$$\frac{d^2\|x(\gamma)\|^2}{d\gamma^2} = \frac{\left(-6\sum\limits_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i-\lambda_\gamma)^4} \sum\limits_{i=1}^{n} \frac{\bar{a}_i^2}{\lambda_i(\lambda_i-\lambda_\gamma)^2} + 4\sum\limits_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i-\lambda_\gamma)^3} \sum\limits_{i=1}^{n} \frac{\bar{a}_i^2}{\lambda_i(\lambda_i-\lambda_\gamma)^3}\right)\lambda'(\gamma)}{\left(\sum\limits_{i=1}^{n} \frac{\bar{a}_i^2}{\lambda_i(\lambda_i-\lambda_\gamma)^2}\right)^2}.$$

From (3.34) and (3.35), our result is proved if we can show for all $\lambda \in (\lambda_1, \lambda_2)$ that

$$-3\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda)^4} \sum_{j=1}^{n} \frac{\bar{a}_j^2}{\lambda_j(\lambda_j - \lambda)^2} + 2\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda)^3} \sum_{j=1}^{n} \frac{\bar{a}_j^2}{\lambda_j(\lambda_j - \lambda)^3}$$

is strictly negative. In fact, we prove the stronger statement, for $\lambda \in (\lambda_1, \lambda_2)$, that

$$(3.43) \qquad -\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda)^4} \sum_{j=1}^{n} \frac{\bar{a}_j^2}{\lambda_j(\lambda_j - \lambda)^2} + \sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda)^3} \sum_{j=1}^{n} \frac{\bar{a}_j^2}{\lambda_j(\lambda_j - \lambda)^3}$$

is strictly negative. We may rewrite (3.43) as

$$\begin{aligned}\sum_{i,j=1}^{n} \frac{\bar{a}_i^2\bar{a}_j^2}{\lambda_j(\lambda_i - \lambda)^4(\lambda_j - \lambda)^2}\left(-1 + \frac{\lambda_i - \lambda}{\lambda_j - \lambda}\right) &= \sum_{i,j=1}^{n} \frac{\bar{a}_i^2\bar{a}_j^2}{\lambda_j(\lambda_i - \lambda)^4(\lambda_j - \lambda)^2}\left(\frac{\lambda_i - \lambda_j}{\lambda_j - \lambda}\right) \\ &= \sum_{i,j=1, i\neq j}^{n} \frac{\bar{a}_i^2\bar{a}_j^2}{\lambda_j(\lambda_i - \lambda)^4(\lambda_j - \lambda)^2}\left(\frac{\lambda_i - \lambda_j}{\lambda_j - \lambda}\right).\end{aligned}$$

The previous sum may be rewritten as

$$\begin{aligned}&\sum_{j=2}^{n}\left\{\frac{\bar{a}_1^2\bar{a}_j^2}{\lambda_j(\lambda_1 - \lambda)^4(\lambda_j - \lambda)^2}\left(\frac{\lambda_1 - \lambda_j}{\lambda_j - \lambda}\right) + \frac{\bar{a}_j^2\bar{a}_1^2}{\lambda_1(\lambda_j - \lambda)^4(\lambda_1 - \lambda)^2}\left(\frac{\lambda_j - \lambda_1}{\lambda_1 - \lambda}\right)\right\} \\ &+ \sum_{i=2}^{n}\sum_{j>i}\left\{\frac{\bar{a}_i^2\bar{a}_j^2}{\lambda_j(\lambda_i - \lambda)^4(\lambda_j - \lambda)^2}\left(\frac{\lambda_i - \lambda_j}{\lambda_j - \lambda}\right) + \frac{\bar{a}_j^2\bar{a}_i^2}{\lambda_i(\lambda_j - \lambda)^4(\lambda_i - \lambda)^2}\left(\frac{\lambda_j - \lambda_i}{\lambda_i - \lambda}\right)\right\}.\end{aligned}$$

Recall, from Assumption 2, that $\lambda_1 < \lambda_2$. Thus, the first sum is strictly negative for $\lambda \in (\lambda_1, \lambda_2)$, where we use the fact that there exists $j \geq 2$ such that $\bar{a}_1\bar{a}_j \neq 0$. We next claim, for $2 \leq i \leq n$ and $i < j \leq n$, that

$$(3.44) \qquad \frac{\bar{a}_i^2\bar{a}_j^2}{\lambda_j(\lambda_i - \lambda)^4(\lambda_j - \lambda)^2}\left(\frac{\lambda_i - \lambda_j}{\lambda_j - \lambda}\right) + \frac{\bar{a}_j^2\bar{a}_i^2}{\lambda_i(\lambda_j - \lambda)^4(\lambda_i - \lambda)^2}\left(\frac{\lambda_j - \lambda_i}{\lambda_i - \lambda}\right)$$

is negative. Indeed, if $\bar{a}_i\bar{a}_j = 0$ or $\lambda_i = \lambda_j$, it is trivial. Otherwise, $\bar{a}_i\bar{a}_j \neq 0$ and $\lambda_i < \lambda_j$ and (3.44) is negative if and only if

$$\frac{-1}{\lambda_j(\lambda_i - \lambda)^4(\lambda_j - \lambda)^3} + \frac{1}{\lambda_i(\lambda_j - \lambda)^4(\lambda_i - \lambda)^3}$$

is negative. Rewriting the last expression, we obtain

$$\frac{-\lambda(\lambda_j-\lambda_i)}{\lambda_i\lambda_j(\lambda_i-\lambda)^4(\lambda_j-\lambda)^4},$$

which is negative, since $\lambda \geq \lambda_1 > 0$ and $\lambda_j > \lambda_i$. Thus (3.43) is strictly negative and $\|x(\gamma)\|^2$ is a strictly convex function for $\gamma \in \Gamma$. □

3.2.1. Bounds on λ^* and γ^*. For this section, we assume a local-nonglobal minimizer of problem (1.2) exists. Our algorithm is based on finding a root γ^* to $\|x(\gamma)\|^2-1$, and we need initial bounds on γ^*. If $\mathcal{K}\cap\mathcal{J}=\emptyset$, then Lemma 3.6 implies $1-d(\lambda_2(B)) < \gamma^* < \infty$. However, this lower bound is of no practical utility, since we aim for an algorithm which exploits the sparsity of A, and computing $d(\lambda_2(B))$ requires a full spectral decomposition of A. Otherwise, if $\mathcal{K}\cap\mathcal{J}\neq\emptyset$, the lemma does not gives us any supplementary information on bounds for γ^*; i.e., we only know $\gamma^* \in \mathbb{R}$. Our next lemma shows that better bounds on γ^* exist, and these will improve the bounds on λ^* in Lemma 2.2.

LEMMA 3.10. *Suppose x^* is a local-nonglobal minimizer of problem* (1.2) *with a corresponding Lagrange multiplier λ^* that satisfies* (2.1)*. Let γ^* be the unique solution to $\lambda(\gamma)+\bar{\lambda}=\lambda^*$. Then $\gamma^* \in [0,2]$.*

Proof. From Theorem 3.8, $x(\gamma^*)$ is the local-nonglobal minimizer, and by feasibility

$$\|x(\gamma^*)\|^2 = \left(\frac{1-\gamma^*}{a^Tv}\right)^2 v^TB^2v = 1.$$

Therefore, it follows from the Cauchy–Schwarz inequality and $v^TB^2v \geq \lambda_1(B)^2\|v\|^2$ that

$$(1-\gamma^*)^2 = \frac{(a^Tv)^2}{v^TB^2v} \leq \left(\frac{\|a\|}{\lambda_1(B)}\right)^2.$$

Taking square roots on both sides of the previous equation yields

$$\gamma^* \in \left[1-\frac{\|a\|}{\lambda_1(B)}, 1+\frac{\|a\|}{\lambda_1(B)}\right].$$

Finally, note from (3.8b) that $\lambda_1(B)=\|a\|$. □

COROLLARY 3.11. *Suppose x^* is a local-nonglobal minimizer of problem* (1.2)*; then* (2.1) *holds with $\lambda^* \in \left[\bar{\lambda}+\lambda(2), \bar{\lambda}+\lambda(0)\right]$.*

Proof. Recall, from Lemma 3.6, that $\lambda(\gamma)$ is a decreasing function. □

Note that Corollary 3.7 implies

$$\bar{\lambda}+\lambda\,(2) > \bar{\lambda}+\lambda_1(B) = \lambda_1(A).$$

Since the inequality holds strictly, it follows that $\bar{\lambda}+\lambda\,(2)$ is a better lower bound for λ^* than the lower bound of $\lambda_1(A)$ which is given in Lemma 2.2.

Similarly, unless $\mathcal{K}\cap\mathcal{J}=\emptyset$ and $0 \leq 1-d(\lambda_2(B))$, it also follows from Corollary 3.7 that $\bar{\lambda}+\lambda(0) < \lambda_2(A)$. Thus unless this case occurs, $\bar{\lambda}+\lambda(0)$ is a better upper bound for λ^* than the upper bound of $\lambda_2(A)$ which is given in Lemma 2.2.

It is also possible to deduce bounds on λ^* from the feasibility of x^*. From (2.3), we have $\|x^*\|^2=\varphi(\lambda^*)$. It follows from (2.6a) that

$$\frac{\bar{a}_i^2}{(\lambda_i(A)-\lambda^*)^2} \leq 1 \text{ for all } i=1,\ldots,n.$$

Hence, by taking square roots on both sides

$$\lambda_1(A) + |\bar{a}_1| \leq \lambda^* \leq \min\{\lambda_i(A) - |\bar{a}_i| : i = 2, \dots, n\}.$$

3.2.2. The algorithm. We are now ready to describe our first algorithm for either computing a possible local-nonglobal minimizer of problem (1.2) or declaring that such a candidate does not exist. From Theorem 3.8, in order to compute the local-nonglobal minimizer we need to find the largest root γ^* of $\|x(\gamma)\|^2 - 1$. Our algorithm is mainly the secant method. It exploits the fact that $\|x(\gamma)\|^2$ is strictly convex for $\gamma \in \Gamma$ and that we have an upper bound on γ^* when a local-nonglobal minimizer exists. To simplify our analysis, let $h(\gamma) := \|x(\gamma)\|^2 - 1$.

```
ALGORITHM 3.1.
1. INITIALIZATION.
  1.1. Let $\gamma_L = 0$, $\gamma_U = 2$, $\gamma_0 = 2.1$, $\gamma_1 = \gamma_U$, and $k = 1$.
  1.2. If $\lambda(\gamma_U) = \lambda_2(B)$ or if $\frac{h(\gamma_1)-h(\gamma_0)}{\gamma_1-\gamma_0} \leq 0$, LNGM $= 0$, else LNGM $= 1$.
2. ITERATION. While LNGM $= 1$ and $\|x(\gamma_k)\| \neq 1$, do
  2.1. $\gamma_{k+1} = \gamma_k - \frac{h(\gamma_k)(\gamma_k-\gamma_{k-1})}{h(\gamma_k)-h(\gamma_{k-1})}$.
  2.2. If $\lambda(\gamma_{k+1}) = \lambda_2(B)$, $\frac{h(\gamma_{k+1})-h(\gamma_k)}{\gamma_{k+1}-\gamma_k} \leq 0$, or $\gamma_{k+1} < \gamma_L$, then LNGM $= 0$.
  2.3. $k = k + 1$.
```

The convergence results for Algorithm 3.1, which we are about to present, are based on the fact that we are using the secant method to find the root of a strictly convex function. To facilitate our analysis, we define the following linear function of γ, which depends on the parameters γ_k and γ_{k-1}:

$$s(\gamma; \gamma_k, \gamma_{k-1}) := h(\gamma_k) + \frac{(h(\gamma_k) - h(\gamma_{k-1}))(\gamma - \gamma_k)}{\gamma_k - \gamma_{k-1}}.$$

The following lemma is a well-known consequence of strict convexity for the function $h(\gamma)$.

LEMMA 3.12. *Let $\gamma_k < \gamma_{k-1}$. For $\gamma \in \mathbb{R}$, the following inequalities and equality hold:*

1. *$h(\gamma) < s(\gamma; \gamma_k, \gamma_{k-1})$ if $\gamma \in (\gamma_k, \gamma_{k-1})$.*
2. *$h(\gamma) = s(\gamma; \gamma_k, \gamma_{k-1})$ if $\gamma \in \{\gamma_k, \gamma_{k-1}\}$.*
3. *$h(\gamma) > s(\gamma; \gamma_k, \gamma_{k-1})$ if $\gamma \notin [\gamma_k, \gamma_{k-1}]$.* □

In Algorithm 3.1, the secant iteration is initiated in γ_0 and γ_1 and halted if, for $k \geq 1$, $\lambda(\gamma_k) = \lambda_2(B)$ or the slope of the secant line going through the points $(\gamma_k, h(\gamma_k))$ and $(\gamma_{k-1}, h(\gamma_{k-1}))$ is not strictly positive. The next lemma shows, in the case in which these situations do not occur, that the sequence $\{\gamma_k\}$ produced by the secant iteration is strictly decreasing and converges to a root of h if bounded below. Such a bound could be γ_L.

LEMMA 3.13. *Let γ_0 and γ_1 be defined as in Algorithm* 3.1, *and assume*

1. *$s(\gamma_{k+1}; \gamma_k, \gamma_{k-1}) = 0$ for $k \geq 1$;*
2. *$\gamma_k \in \Gamma$ for $k \geq 0$;*
3. *$\frac{h(\gamma_k)-h(\gamma_{k-1})}{\gamma_k-\gamma_{k-1}} > 0$ for $k \geq 1$;*
4. *$h(\gamma_1) > 0$.*

Then $\{\gamma_k\}$ is a strictly decreasing sequence. Furthermore, if $\{\gamma_k\}$ is bounded below, then the sequence converges to $\bar{\gamma}$, which satisfies $h(\bar{\gamma}) = 0$ and $h'(\bar{\gamma}) \geq 0$.

Proof. Since $s(\gamma; \gamma_1, \gamma_0)$ is a function with positive slope by assumption, since γ_2 is its root, and since $h(\gamma_1) > 0$, then clearly $\gamma_2 < \gamma_1$. By item 3 of Lemma 3.12,

$h(\gamma_2) > s(\gamma_2; \gamma_1, \gamma_0) = 0$. By induction we may similarly prove that $\{\gamma_k\}$ is a strictly decreasing sequence and $h(\gamma_k) > 0$ for $k \geq 0$.

If $\{\gamma_k\}$ is bounded below, then since it is a decreasing sequence it converges, say, to $\bar{\gamma}$. Now by the mean value theorem, for $k \geq 1$,

$$\frac{h(\gamma_k) - h(\gamma_{k-1})}{\gamma_k - \gamma_{k-1}} = h'(c_k) \quad \text{for } c_k \in [\gamma_k, \gamma_{k-1}].$$

By assumption $h'(c_k) > 0$ for $k \geq 1$, and since h is strictly convex, we deduce that for $k \geq 1$

$$\frac{h(\gamma_k) - h(\gamma_{k-1})}{\gamma_k - \gamma_{k-1}} < h'(c_0). \tag{3.45}$$

Convergence of $\{\gamma_k\}$ implies

$$0 = \lim_{k\to\infty} |\gamma_{k+1} - \gamma_k| = \lim_{k\to\infty} \frac{h(\gamma_k)}{\frac{h(\gamma_k)-h(\gamma_{k-1})}{\gamma_k - \gamma_{k-1}}}.$$

By (3.45) the denominator in the last limit is bounded away from infinity; hence the numerator converges to zero, i.e., $h(\bar{\gamma}) = 0$. Finally, since $c_k \in [\gamma_k, \gamma_{k-1}]$, then c_k converges to $\bar{\gamma}$ and thus $h'(\bar{\gamma}) = \lim_{k\to\infty} h'(c_k) \geq 0$. □

For our convergence results, we need to make one further assumption concerning problem (1.2).

Assumption 3. If problem (1.2) does not have a local-nonglobal minimizer, then for $\epsilon > 0$ small enough, the equality-constrained trust-region subproblem

$$\begin{array}{ll} \min\limits_{x} & x^T A x - 2a^T x \\ \text{s.t.} & \|x\| = 1 + \epsilon \end{array} \tag{3.46}$$

does not have a local-nonglobal minimizer.

The scalar 1 on the right-hand side of the equality constraint of problem (3.46) is due to the fact that we assume $\Delta = 1$ in problem (1.2). Now consider for a moment that Δ is allowed to vary in (1.2). Our assumption mainly says that for Δ larger than but close enough to 1, there is no local-nonglobal minimizer. Note that if Assumption 3 holds, then for $\Delta < 1$ there is no local-nonglobal minimizer as well. This is a consequence of Theorem 2.5 and of the strict convexity of the function φ defined in (2.3). Furthermore, in view of Lemma 3.5, there exists Δ_0 such that (1.2) admits a local-nonglobal minimizer for all $\Delta > \Delta_0$. Thus in case a local-nonglobal minimizer does not exist for (1.2) the assumption is equivalent to the inequality $1 < \Delta_0$.

Under the extra Assumption 3, the following theorem holds.

THEOREM 3.14. *The sequence $\{\gamma_k\}$ produced by Algorithm* 3.1 *either converges to γ^* such that $x(\gamma^*)$ is a local-nonglobal minimizer of problem* (1.2) *or there does not exist a local-nonglobal minimizer of problem* (1.2) *and LNGM is set to* 0.

Proof. First, consider the case where a local-nonglobal minimizer for problem (1.2) exists. Let γ^* be defined as in item 1 of Theorem 3.8. Then

$$h(\gamma^*) = 0 \qquad \text{and} \qquad h'(\gamma^*) \geq 0. \tag{3.47}$$

Recall that γ_1 is an upper bound on γ^*. If $\gamma_1 = \gamma^*$, then $\|x(\gamma_1)\| = 1$. Hence, Algorithm 3.1 terminates and there is nothing to prove.

Assume $\gamma_1 > \gamma^*$. Note, from Lemma 2.2, that $\lambda^* \in (\lambda_1(A), \lambda_2(A))$ and, from Theorem 3.8, that $\lambda^* = \lambda(\gamma^*) + \bar{\lambda}$. Hence $\lambda(\gamma^*) \in (\lambda_1(B), \lambda_2(B))$, i.e., $\gamma^* \in \Gamma$. Since $\lambda(\gamma)$ is a decreasing function, $\lambda(\gamma_1) \leq \lambda(\gamma^*) < \lambda_2(B)$. From Corollary 3.7, $\lambda(\gamma_1) > \lambda_1(B)$. Hence, $\gamma_1 \in \Gamma$. Since h is strictly convex, since $\gamma_1 > \gamma^*$, and since (3.47) holds, $h(\gamma_1) > 0$ and $h'(\gamma_1) > 0$.

Suppose for $k \geq 1$ that $\gamma_k > \gamma^*$, $\gamma_k \in \Gamma$, $h(\gamma_k) > 0$, and $h'(\gamma_k) > 0$. Note that we just proved that these conditions hold for $k = 1$. We wish to show the following:

1. $\frac{h(\gamma_k)-h(\gamma_{k-1})}{\gamma_k-\gamma_{k-1}} > 0$.
2. $h(\gamma_{k+1}) > 0$.
3. $\gamma_{k+1} > \gamma^*$.
4. $h'(\gamma_{k+1}) > 0$.
5. $\gamma_{k+1} \in \Gamma$.

Since

$$\frac{h(\gamma_k) - h(\gamma_{k-1})}{\gamma_k - \gamma_{k-1}} = h'(c_k) \quad \text{for } c_k \in [\gamma_k, \gamma_{k-1}],$$

and h is strictly convex, then $h'(c_k) \geq h'(\gamma_k) > 0$, proving item 1. It follows that $s(\gamma; \gamma_k, \gamma_{k-1})$ is strictly increasing. Since $0 = s(\gamma_{k+1}; \gamma_k, \gamma_{k-1})$ and $s(\gamma_k; \gamma_k, \gamma_{k-1}) = h(\gamma_k) > 0$, then $\gamma_{k+1} < \gamma_k$. By Lemma 3.12, $h(\gamma_{k+1}) > s(\gamma_{k+1}; \gamma_k, \gamma_{k-1}) = 0$, and this proves item 2. We have $\gamma^* < \gamma_{k+1}$; otherwise $\gamma^* \in (\gamma_{k+1}, \gamma_k)$, and by Lemma 3.12

$$0 = s(\gamma_{k+1}; \gamma_k, \gamma_{k-1}) < s(\gamma^*; \gamma_k, \gamma_{k-1}) < h(\gamma^*).$$

This contradicts $h(\gamma^*) = 0$, proving item 3. Since $h'(\gamma^*) \geq 0$, by strict convexity we have $h'(\gamma_{k+1}) > 0$, proving item 4. Finally, from an argument similar to that above, $\gamma_{k+1} \in \Gamma$ holds, proving item 5.

By induction it follows that for all $k \geq 1$

1. $\frac{h(\gamma_k)-h(\gamma_{k-1})}{\gamma_k-\gamma_{k-1}} > 0$;
2. $h(\gamma_k) > 0$;
3. $\gamma_k > \gamma^*$;
4. $h'(\gamma_k) > 0$;
5. $\gamma_k \in \Gamma$.

It follows from Lemma 3.13 that $\{\gamma_k\}$ converges, say, to $\bar{\gamma}$, which satisfies $h(\bar{\gamma}) = 0$ and $h'(\bar{\gamma}) \geq 0$. By strict convexity of h and since $h(\gamma^*) = 0$ and $h'(\gamma^*) \geq 0$, then $\bar{\gamma} = \gamma^*$.

Second, consider the case where a local-nonglobal minimizer of problem (1.2) does not exist. Suppose there exists $\hat{\gamma} \in \Gamma$ such that $h(\hat{\gamma}) < 0$; then since h is strictly convex and, by Lemma 3.9, since $\lim_{\gamma\to\infty} h(\gamma) = \infty$, there exists $\gamma^* > \hat{\gamma}$ such that $\gamma^* \in \Gamma$, $h(\gamma^*) = 0$, and $h'(\gamma^*) > 0$. Then by Theorem 3.8, there exists a local-nonglobal minimizer of problem (1.2), yielding a contradiction. Thus, $h(\gamma) \geq 0$ for $\gamma \in \Gamma$. In fact, this inequality holds strictly. Otherwise, from the strict convexity of h, for every $\epsilon > 0$, the equation $h(\gamma) = \epsilon(\epsilon+2)$ has a solution, say $\hat{\gamma} \in \Gamma$, with $h'(\hat{\gamma}) > 0$. From item 3 of Theorem 3.8, $x(\hat{\gamma})$ is a local-nonglobal minimizer of problem (3.39) with $\|x(\hat{\gamma})\| = 1 + \epsilon$. This contradicts Assumption 3. Hence $h(\gamma) > 0$ for $\gamma \in \Gamma$. We obtain in particular that $h(\gamma_1) > 0$.

If a sequence $\{\gamma_k\}$ obtained with Algorithm 3.1 would be bounded below and satisfy, for all $k \geq 1$, $\gamma_k \in \Gamma$ and

$$\frac{h(\gamma_k) - h(\gamma_{k-1})}{\gamma_k - \gamma_{k-1}} > 0,$$

then by Lemma 3.13, $\{\gamma_k\}$ would converge, say to $\bar{\gamma}$, which would satisfy $h(\bar{\gamma}) = 0$ and $h'(\bar{\gamma}) \geq 0$. This would imply that for every $\epsilon > 0$, the equation $h(\gamma) = \epsilon(\epsilon + 2)$ has a solution, say $\hat{\gamma} \in \Gamma$, with $h'(\hat{\gamma}) > 0$, contradicting (as explained above) Assumption 3. Thus, there must exist some $\bar{k} \geq 1$ such that one of the following cases is true:

1. $\gamma_{\bar{k}} < \gamma_L$.
2. $\frac{h(\gamma_{\bar{k}}) - h(\gamma_{\bar{k}-1})}{\gamma_{\bar{k}} - \gamma_{\bar{k}-1}} \leq 0$.
3. $\gamma_{\bar{k}} \notin \Gamma$ (if and only if $\lambda(\gamma_{\bar{k}}) = \lambda_2(B)$).

In each case, $LNGM$ is set to 0 and a local-nonglobal minimizer of problem (1.2) does not exist. □

COROLLARY 3.15. *Suppose x^* is a local-nonglobal minimizer of problem* (1.2) *with a corresponding Lagrange multiplier λ^* that satisfies* (2.1). *Let γ^* be the unique solution to $\lambda(\gamma) + \bar{\lambda} = \lambda^*$. If $h'(\gamma^*) > 0$, the sequence $\{\gamma_k\}$ produced by Algorithm* 3.1 *converges to γ^* superlinearly and $x(\gamma^*)$ is a strict local-nonglobal minimizer of problem* (1.2).

Proof. The proof holds from Theorems 3.8 and 3.14 and because the secant method converges superlinearly when it converges to a simple root; see, e.g., Neumaier [19, Corollary 5.4.2]. □

3.2.3. The relation between a local-nonglobal minimizer and $g(\gamma)$. As mentioned at the beginning of section 3, we expect the function $g(\gamma) = \gamma\lambda(\gamma)$ to be intimately related to a local-nonglobal minimizer of problem (1.2). This will be made clear in Theorems 3.18 and 3.19. We are for now concerned with the first derivative of g.

LEMMA 3.16. *For $\gamma \in \Gamma \setminus \{1\}$, let the vector $v \in \mathbb{R}^n$ satisfy the generalized eigenvalue equation* (3.17). *Then we may write the derivative of g as*

$$g'(\gamma) = \lambda(\gamma) - \frac{\gamma}{v^T B v}\left(\frac{a^T v}{1-\gamma}\right)^2. \tag{3.48}$$

Proof. Using Corollary 3.4 and (3.34), we obtain

$$\begin{aligned} \lambda_1(B) < \lambda_1(B(\gamma)) < \lambda_2(B) \leq \lambda_2(B(\gamma)) \quad &\text{if } \gamma > 1,\\ \lambda_1(B(\gamma)) \leq \lambda_1(B) < \lambda_2(B(\gamma)) < \lambda_2(B) \leq \lambda_3(B(\gamma)) \quad &\text{if } \gamma < 1. \end{aligned}$$

Hence $\lambda(\gamma)$ is an eigenvalue of $B(\gamma)$ of multiplicity one and it is easy to see that $\frac{B^{1/2}v}{\|B^{1/2}v\|}$ is a corresponding unit norm eigenvector. Therefore (see, e.g., [12]),

$$\begin{aligned} g'(\gamma) &= \lambda(\gamma) - \gamma\left(\frac{B^{1/2}v}{\|B^{1/2}v\|}\right)^T\left(\frac{1}{(1-\gamma)^2}B^{-1/2}aa^TB^{-1/2}\right)\left(\frac{B^{1/2}v}{\|B^{1/2}v\|}\right)\\ &= \lambda(\gamma) - \frac{\gamma}{v^TBv}\left(\frac{a^Tv}{1-\gamma}\right)^2. \qquad \square \end{aligned}$$

Our next lemma shows how $\|x(\gamma)\|$ is related to the first derivative of g.

LEMMA 3.17. *Let $\gamma \in \Gamma$. Then*

$$\|x(\gamma)\| > \ (=,<)\ 1 \iff g'(\gamma) > \ (=,<)\ 0. \tag{3.49}$$

Proof. For $\gamma \in \Gamma \setminus \{1\}$, we have

$$\|x(\gamma)\|^2 = \left(\frac{1-\gamma}{a^T v}\right)^2 v^T B^2 v,$$

$$= 1 - \gamma + \lambda(\gamma)\left(\frac{1-\gamma}{a^T v}\right)^2 v^T B v \tag{3.50a}$$

$$= 1 + g'(\gamma)\left(\frac{1-\gamma}{a^T v}\right)^2 v^T B v \tag{3.50b}$$

$$= 1 + g'(\gamma)\|x(\gamma)\|^2 \frac{v^T B v}{v^T B^2 v}, \tag{3.50c}$$

where (3.50a) follows from (3.17), and (3.50b) follows from (3.48). The conclusion follows by writing (3.50c) as

$$\|x(\gamma)\|^2 = \frac{1}{1 - g'(\gamma)\frac{v^T B v}{v^T B^2 v}} \tag{3.51}$$

and noting that in the case where $\gamma = 1 \in \Gamma$, the relation (3.49) holds as well by the continuity of the functions g' and $\|x(\gamma)\|^2$. □

We are now ready to answer, with the next two theorems, how optimums of the function g are related to local-nonglobal minimizers of problem (1.2). The first one states that if a local-nonglobal minimizer of problem (1.2) exists, then there exists γ^* such that the first and second order optimality conditions for a local minimizer of g are satisfied. The second one is almost its converse: if the first and second order sufficient optimality conditions for a local minimizer of g are satisfied at some γ^*, then $x(\gamma^*)$ is the local-nonglobal minimizer of problem (1.2).

THEOREM 3.18. *Suppose x^* is a local-nonglobal minimizer of problem* (1.2) *with a corresponding Lagrange multiplier λ^* that satisfies* (2.1). *Let γ^* be the unique solution to $\lambda(\gamma) + \bar{\lambda} = \lambda^*$. Then $g'(\gamma^*) = 0$ and $g''(\gamma^*) \geq 0$.*

Proof. By Lemma 2.2, $\lambda(\gamma^*) \in (\lambda_1(B), \lambda_2(B))$ and thus $\gamma^* \in \Gamma$. Theorem 3.8 gives $x^* = x(\gamma^*)$. The fact that $g'(\gamma^*) = 0$ follows from the feasibility of x^* and from Lemma 3.17.

By (3.35) and (3.38), and since, by Theorem 2.5, $\varphi'(\lambda^*) \leq 0$, we obtain

$$\frac{d\|x(\gamma^*)\|^2}{d\gamma} = \frac{d\varphi(\lambda(\gamma^*) + \bar{\lambda})}{d\lambda}\,\lambda'(\gamma^*) \geq 0. \tag{3.52}$$

If $g''(\gamma^*) < 0$, then $g'(\gamma^* + h) < g'(\gamma^*) = 0$ for $h > 0$ small enough, and, using Lemma 3.17, we deduce $\|x(\gamma^* + h)\| < 1$. Thus, since $\|x(\gamma^*)\| = 1$,

$$\frac{d\|x(\gamma^*)\|^2}{d\gamma} = \lim_{h \to 0} \frac{\|x(\gamma^* + h)\|^2 - \|x(\gamma^*)\|^2}{h} \leq 0. \tag{3.53}$$

Inequalities (3.52) and (3.53) give

$$\frac{d\|x(\gamma^*)\|^2}{d\gamma} = 0.$$

It follows then from (3.52) and $\lambda'(\gamma^*) < 0$ that $\varphi'(\lambda(\gamma^*) + \bar{\lambda}) = 0$. From (2.6c), φ is strictly convex over the interval $(\lambda_1(A), \lambda_2(A))$ and thus $\lambda(\gamma^*) + \bar{\lambda}$ is its strict minimizer. By (3.37), the following inequality thus holds:

$$\|x(\gamma)\| \geq \|x(\gamma^*)\| = 1 \text{ for } \gamma \in \Gamma.$$

This contradicts $\|x(\gamma^* + h)\| < 1$ for $h > 0$ small enough. Thus $g''(\gamma^*) \geq 0$. □

Theorem 3.19. *Suppose $\gamma^* \in \mathbb{R}$ satisfies $g'(\gamma^*) = 0$ and $g''(\gamma^*) > 0$; then $x(\gamma^*)$ is a strict local-nonglobal minimizer of* (1.2) *with Lagrange multiplier $\lambda^* := \lambda(\gamma^*) + \bar{\lambda}$.*

Proof. From Lemma 3.6, $g''(\gamma^*) \neq 0$ implies $\lambda(\gamma^*) \in (\lambda_1(B), \lambda_2(B))$. Therefore $x(\gamma^*)$ is well defined, and if we let $x^* := x(\gamma^*)$ and $\lambda^* := \lambda(\gamma^*) + \bar{\lambda}$, then by (3.18), the stationarity condition (2.1) is satisfied. Feasibility of x^* follows from Lemma 3.17. If we can further show that $\varphi'(\lambda^*) < 0$, then the result follows from item 3 of Theorem 2.5.

Since $g''(\gamma^*) > 0$, then $g'(\gamma^* - h) < g'(\gamma^*) = 0$ for $h > 0$ small enough. By Lemma 3.17, this implies $\|x(\gamma^* - h)\| < 1$, and thus

$$\frac{d\|x(\gamma^*)\|^2}{d\gamma} = \lim_{h \to 0} \frac{\|x(\gamma^*)\|^2 - \|x(\gamma^* - h)\|^2}{h} \geq 0. \tag{3.54}$$

By an argument similar to that in the proof of Theorem 3.18, we conclude that the inequality in (3.54) holds strictly. From (3.35) and (3.38), we deduce $\varphi'(\lambda^*) < 0$. □

3.3. Computing a local-nonglobal minimizer: Second method. The ideas involved in deriving our second method for computing a local-nonglobal minimizer are similar to those of section 3.2. Our analysis relies on the functions $\lambda_2(D(t))$, $\|x(t)\|^2 - 1$, and $m(t)$. The results of this section closely follow those of section 3.2. Hence we start by investigating the function $\lambda_2(D(t))$.

We will make use in our analysis of the function

$$p(\lambda) := \lambda + \sum_{j \in \mathcal{J}} \frac{\bar{a}_j^2}{\lambda_j(A) - \lambda}$$

and of the sets $\mathcal{J}$ and $\mathcal{K}$ defined in (3.28).

Lemma 3.20.

1. *If $\mathcal{K} \cap \mathcal{J} \neq \emptyset$, $\lambda_2(D(t))$ is infinitely differentiable and satisfies $p(\lambda_2(D(t))) = t$. Moreover,*

$$\lambda_2'(D(t)) = \frac{1}{p'(\lambda_2(D(t)))} \quad \text{and} \quad \lambda_2''(D(t)) = \frac{-p''(\lambda_2(D(t)))}{[p'(\lambda_2(D(t)))]^3} \tag{3.55}$$

for all $t \in \mathbb{R}$.

2. *If $\mathcal{K} \cap \mathcal{J} = \emptyset$, $\lambda_2(D(t))$ is continuous and infinitely differentiable for $t \in \mathbb{R} \setminus \{p(\lambda_2(A))\}$.*

(i) *For $t < p(\lambda_2(A))$, $\lambda_2(D(t))$ satisfies $p(\lambda_2(D(t))) = t$ and*

$$\lambda_2'(D(t)) = \frac{1}{p'(\lambda_2(D(t)))} \quad \text{and} \quad \lambda_2''(D(t)) = \frac{-p''(\lambda_2(D(t)))}{[p'(\lambda_2(D(t)))]^3}. \tag{3.56}$$

(ii) *For $t > p(\lambda_2(A))$, $\lambda_2(D(t)) = \lambda_2(A)$, $\lambda_2'(D(t)) = 0$, and $\lambda_2''(D(t)) = 0$.*

(iii) *For $t = p(\lambda_2(A))$, $\lambda_2(D(t)) = \lambda_2(A)$ and satisfies $p(\lambda_2(D(t))) = t$; the right- and left-hand side derivatives are given, respectively, by*

$$\lambda_2'(D(t^+)) = \frac{1}{p'(\lambda_2(A))}, \quad \lambda_2''(D(t^+)) = \frac{-d''(\lambda_2(A))}{[d'(\lambda_2(A))]^3}, \tag{3.57a}$$

$$\lambda_2'(D(t^-)) = 0, \qquad \lambda_2''(D(t^-)) = 0. \tag{3.57b}$$

Proof. 1. Expanding the determinant of the matrix $D(t) - \lambda I$ with respect to its first column gives

$$\text{(3.58a)}\quad \det(D(t) - \lambda I) = (t - \lambda) \prod_{j=1}^{n} (\lambda_j(A) - \lambda) - \sum_{k=1}^{n} \left(\bar{a}_k^2 \prod_{j \neq k}^{n} (\lambda_j(A) - \lambda) \right)$$

$$\text{(3.58b)}\qquad = (t - p(\lambda)) \prod_{j=1}^{n} (\lambda_j(A) - \lambda) \quad \text{for } \lambda \notin \{\lambda_j(A) | j \in J\}.$$

Since $\mathcal{K} \cap \mathcal{J} \neq \emptyset$ and $\bar{a}_1 \neq 0$,

$$\lim_{\lambda \searrow \lambda_1(A)} p(\lambda) = -\infty \ \text{ and } \ \lim_{\lambda \nearrow \lambda_2(A)} p(\lambda) = \infty.$$

Furthermore,

$$p'(\lambda) = 1 + \sum_{j \in \mathcal{J}} \frac{\bar{a}_j^2}{(\lambda_j(A) - \lambda)^2} > 0.$$

Therefore, for all $t \in \mathbb{R}$, $p^{-1}(t)$ is well defined, where $p^{-1}(t) \in (\lambda_1(A), \lambda_2(A))$. Moreover, (3.58b) shows it is an eigenvalue of $D(t)$. From Lemma 3.1, this shows $p^{-1}(t) = \lambda_2(D(t))$. Hence, $\lambda_2(D(t))$ is infinitely differentiable and equations (3.55) are obtained by implicit differentiation.

2. Let $t \in \mathbb{R}$. Since $\mathcal{K} \cap \mathcal{J} = \emptyset$, then $p(\lambda_2(A))$ is well defined. By (3.58b), $\lambda_2(A)$ is an eigenvalue of $D(t)$. Assumptions 1 and 2 imply that $\lambda_1(A)$ is not an eigenvalue of $D(t)$, since from (3.58a) we obtain

$$\det(D(t) - \lambda_1(A) I) = -\bar{a}_1^2 \prod_{j=2}^{n} (\lambda_j(A) - \lambda_1(A)) \neq 0.$$

Combining this last inequality with Lemma 3.1 gives

$$\text{(3.59)}\qquad \lambda_2(D(t)) \in (\lambda_1(A), \lambda_2(A)].$$

Note again that $p(\lambda)$ is strictly increasing for $\lambda \in (\lambda_1(A), \lambda_2(A)]$ and therefore

$$\text{(3.60)}\qquad p(\lambda) < p(\lambda_2(A)) \ \text{ for } \ \lambda \in (\lambda_1(A), \lambda_2(A)).$$

(i) If $t < p(\lambda_2(A))$, then $p^{-1}(t)$ is well defined, where $p^{-1}(t) \in (\lambda_1(A), \lambda_2(A))$. The rest of the proof is similar to the proof of item 1.

(ii) and (iii) If $t \geq p(\lambda_2(A))$, by (3.60), $p(\lambda) < t$ for $\lambda \in (\lambda_1(A), \lambda_2(A))$. Therefore, from (3.58b), there are no eigenvalues of $D(t)$ in the interval $[\lambda_1(A), \lambda_2(A))$ and, using the expression (3.59), we obtain $\lambda_2(A) = \lambda_2(D(t))$. In particular, the derivatives of $\lambda_2(D(t))$ for $t > p(\lambda_2(A))$ are zero and equations (3.57b) hold. Note finally that $t = p(\lambda_2(D(t)))$ for $t \leq p(\lambda_2(A))$ and thus (3.57a) holds. □

COROLLARY 3.21. *For $t \in \mathbb{R}$, $\lambda_2(D(t)) > \lambda_1(A)$ and $\lim_{t \to -\infty} \lambda_2(D(t)) = \lambda_1(A)$. Moreover,*

1. *if $\mathcal{K} \cap \mathcal{J} \neq \emptyset$, then $\lambda_2(D(t)) < \lambda_2(A)$ and $\lim_{t \to \infty} \lambda_2(D(t)) = \lambda_2(A)$;*
2. *if $\mathcal{K} \cap \mathcal{J} = \emptyset$, then*
   (i) *$\lambda_2(D(t)) = \lambda_2(A)$ for $t \geq p(\lambda_2(A))$,*
   (ii) *$\lambda_2(D(t)) < \lambda_2(A)$ for $t < p(\lambda_2(A))$.* □

Define

$$\mathcal{T} := \begin{cases} \mathbb{R} & \text{if } \mathcal{K} \cap \mathcal{J} \neq \emptyset, \\ (-\infty, p(\lambda_2(A)) & \text{if } \mathcal{K} \cap \mathcal{J} = \emptyset. \end{cases}$$

Note, from Lemma 3.20 and Corollary 3.21, that

$$\mathcal{T} = \{t : \lambda_2(D(t)) \in (\lambda_1(A), \lambda_2(A))\}, \tag{3.61}$$

$$\lambda_2'(D(t)) > 0 \quad \text{for } t \in \mathcal{T}. \tag{3.62}$$

For $t \in \mathcal{T}$, let the vector $y \in \mathbb{R}^n$ (which depends on t) be a unit norm eigenvector of $D(t)$ for the eigenvalue $\lambda_2(D(t))$. Let $y := (y_0, z)^T$, where $y_0 \in \mathbb{R}$ and $z \in \mathbb{R}^n$. Similarly to equations (3.4) we have

$$ty_0 - a^T z = \lambda_2(D(t))y_0, \tag{3.63a}$$

$$-y_0 a + Az = \lambda_2(D(t))z. \tag{3.63b}$$

Note that $y_0 \neq 0$ for $t \in \mathcal{T}$; otherwise, by (3.63b), this would imply that $\lambda_2(D(t))$ is an eigenvalue of A. Hence, for $t \in \mathcal{T}$, we may define

$$x(t) := 1/y_0 \, z. \tag{3.64}$$

Recall that $x(t)$ satisfies (3.7) and notice that (3.61) implies that $A - \lambda_2(D(t))I$ is invertible. Thus using (2.3) we may write $\|x(t)\|^2$ as

$$\|x(t)\|^2 = \varphi(\lambda_2(D(t))). \tag{3.65}$$

It follows that

$$\frac{d\|x(t)\|^2}{dt} = \frac{d\varphi(\lambda_2(D(t)))}{d\lambda} \lambda_2'(D(t)). \tag{3.66}$$

Again, the algorithm of this section is based on finding a root of the function $\|x(t)\|^2 - 1$ in the interval $\mathcal{T}$. A theorem similar to Theorem 2.5 holds.

THEOREM 3.22.

1. *If x^* is a local-nonglobal minimizer of problem* (1.2), *then* (2.1) *holds with $\lambda^* \in (\lambda_1(A), \lambda_2(A))$. Let t^* be the unique solution to $\lambda_2(D(t)) = \lambda^*$; then $x^* = x(t^*)$ and $\frac{d\|x(t^*)\|^2}{dt} \leq 0$.*
2. *If, for $t^* \in \mathcal{T}$, $\|x(t^*)\| = 1$ and $\frac{d\|x(t^*)\|^2}{dt} < 0$, then $x(t^*)$ is a strict local-nonglobal minimizer of* (1.2).
3. *For $t \in \mathcal{T} \cap \{t : \frac{d\|x(t)\|^2}{dt} < 0\}$, $x(t)$ is a strict local-nonglobal minimizer of*

$$\begin{array}{ll} \min\limits_{x} & x^T A x - 2a^T x \\ s.t. & \|x\| = \|x(t)\| \end{array} \tag{3.67}$$

with Lagrange multiplier $\lambda_2(D(t))$.

Proof. Since $t^* \in \mathcal{T}$, the proofs of items 1 and 2 follow from Theorem 2.5 and (3.62), (3.65), and (3.66). To prove item 3, fix $t \in \mathcal{T}$ and let $\delta := \|x(t)\|$. Note that $x(t)$ is a local-nonglobal minimizer of problem (3.67) if and only if problem (3.40) is solved by $x(t; \delta) := x(t)/\delta$.

For $t \in \mathcal{T} \cap \{t : \frac{d\|x(t)\|^2}{dt} < 0\}$, it follows from (3.62) and (3.66) that

$$\frac{d\varphi(\lambda_2(D(t)))}{d\lambda} < 0. \tag{3.68}$$

By item 3 of Theorem 2.5, $\|x(t;\delta)\|$ solves problem (3.40) since

$$((\delta^2 A) - (\delta^2 \lambda_2(D(t)))I)x(t;\delta) = \delta a,$$

since $\lambda_1(\delta^2 A) < \delta^2 \lambda_2(D(t)) < \lambda_2(\delta^2 A)$, and since, from (3.41) and (3.68),

$$\varphi'(\delta^2 \lambda_2(D(t));\delta) = \varphi'(\lambda_2(D(t))) < 0. \qquad \square$$

As for Algorithm 3.1, a key property for the algorithm of this section is that $\|x(t)\|^2$ is a strictly convex function.

LEMMA 3.23. *Consider the function* $\|x(t)\|^2$ *with domain* $\mathcal{T}$. *Then it is an infinitely differentiable strictly convex function and* $\lim_{t\to-\infty} \|x(t)\|^2 = \infty$.

Proof. Since $\varphi(\lambda)$ and $\lambda_2(D(t))$ are infinitely differentiable, respectively, on the intervals $(\lambda_1(A), \lambda_2(A))$ and $\mathcal{T}$, and $\lambda_2(D(t)) \in (\lambda_1(A), \lambda_2(A))$, then infinite differentiability follows from (3.65). By Corollary 3.21, $\lim_{t\to-\infty} \lambda_2(D(t)) = \lambda_1(A)$ and $\lambda_2(D(t)) > \lambda_1(A)$, and, by Assumption 2, $\lim_{\lambda \searrow \lambda_1(A)} \varphi(\lambda) = \infty$. Thus, using (3.65), $\lim_{t\to-\infty} \|x(t)\|^2 = \infty$.

All that is left to prove is strict convexity. For simplicity, let $\lambda_i = \lambda_i(A)$ for $i = 1, \ldots, n$, let $\lambda_t = \lambda_2(D(t))$, and let $\lambda_t' = \lambda_2'(D(t))$. There are two cases to consider.

Case 1: $\bar{a}_1 \neq 0$ *and* $\bar{a}_j = 0$ *for* $j = 2, \ldots, n$. We have in this case

$$\begin{aligned}
\|x(t)\|^2 &= \frac{\bar{a}_1^2}{(\lambda_1 - \lambda_t)^2}, \\
\frac{d\|x(t)\|^2}{dt} &= \frac{2\bar{a}_1^2}{(\lambda_1 - \lambda_t)^3}\lambda_t' = -\frac{2\bar{a}_1^2}{(\lambda_1 - \lambda_t)^3 + \bar{a}_1^2(\lambda_1 - \lambda_t)},
\end{aligned}$$

where we have used (3.55) and (3.56) to obtain the first derivative. Thus, using (3.62),

$$\frac{d^2\|x(t)\|^2}{dt^2} = \frac{2\bar{a}_1^2(3(\lambda_1 - \lambda_t)^2 + \bar{a}_1^2)}{((\lambda_1 - \lambda_t)^3 + \bar{a}_1^2(\lambda_1 - \lambda_t))^2}\lambda_t' > 0.$$

Case 2: $\exists j \geq 2$ *such that* $\bar{a}_1 \bar{a}_j \neq 0$. We have, using once again (3.55) and (3.56),

$$\begin{aligned}
\|x(t)\|^2 &= \sum_{i=1}^n \frac{\bar{a}_i^2}{(\lambda_i - \lambda_t)^2}, \\
\frac{d\|x(t)\|^2}{dt} &= 2\sum_{i=1}^n \frac{\bar{a}_i^2}{(\lambda_i - \lambda_t)^3}\lambda_t' = \frac{2\sum_{i=1}^n \frac{\bar{a}_i^2}{(\lambda_i - \lambda_t)^3}}{1 + \sum_{i=1}^n \frac{\bar{a}_i^2}{(\lambda_i - \lambda_t)^2}}, \\
\frac{d^2\|x(t)\|^2}{dt^2} &= \frac{\left(6\sum_{i=1}^n \frac{\bar{a}_i^2}{(\lambda_i - \lambda_t)^4}\left(1 + \sum_{i=1}^n \frac{\bar{a}_i^2}{(\lambda_i - \lambda_t)^2}\right) - 4\sum_{i=1}^n \frac{\bar{a}_i^2}{(\lambda_i - \lambda_t)^3}\sum_{i=1}^n \frac{\bar{a}_i^2}{\lambda_i(\lambda_i - \lambda_t)^3}\right)\lambda_t'}{\left(1 + \sum_{i=1}^n \frac{\bar{a}_i^2}{(\lambda_i - \lambda_t)^2}\right)^2}.
\end{aligned}$$

From (3.61) and (3.62), our result is proved if we can show for all $\lambda \in (\lambda_1, \lambda_2)$ that

$$3\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda)^4} \left(1 + \sum_{j=1}^{n} \frac{\bar{a}_j^2}{(\lambda_j - \lambda)^2} \right) - 2\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda)^3} \sum_{j=1}^{n} \frac{\bar{a}_j^2}{(\lambda_j - \lambda)^3}$$

is strictly positive. In fact, we prove the stronger statement, for $\lambda \in (\lambda_1, \lambda_2)$, that

$$\sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda)^4} \sum_{j=1}^{n} \frac{\bar{a}_j^2}{(\lambda_j - \lambda)^2} - \sum_{i=1}^{n} \frac{\bar{a}_i^2}{(\lambda_i - \lambda)^3} \sum_{j=1}^{n} \frac{\bar{a}_j^2}{(\lambda_j - \lambda)^3} \tag{3.69}$$

is strictly positive. We may rewrite (3.69) as

$$\begin{aligned}\sum_{i,j=1}^{n} \frac{\bar{a}_i^2 \bar{a}_j^2}{(\lambda_i - \lambda)^4 (\lambda_j - \lambda)^2} \left(1 - \frac{\lambda_i - \lambda}{\lambda_j - \lambda} \right) &= \sum_{i,j=1}^{n} \frac{\bar{a}_i^2 \bar{a}_j^2}{(\lambda_i - \lambda)^4 (\lambda_j - \lambda)^2} \left(\frac{\lambda_j - \lambda_i}{\lambda_j - \lambda} \right) \\ &= \sum_{i,j=1, i\neq j}^{n} \frac{\bar{a}_i^2 \bar{a}_j^2}{(\lambda_i - \lambda)^4 (\lambda_j - \lambda)^2} \left(\frac{\lambda_j - \lambda_i}{\lambda_j - \lambda} \right).\end{aligned}$$

The previous sum may be rewritten as

$$\begin{aligned}&\sum_{j=2}^{n} \left\{ \frac{\bar{a}_1^2 \bar{a}_j^2}{(\lambda_1 - \lambda)^4 (\lambda_j - \lambda)^2} \left(\frac{\lambda_j - \lambda_1}{\lambda_j - \lambda} \right) + \frac{\bar{a}_j^2 \bar{a}_1^2}{(\lambda_j - \lambda)^4 (\lambda_1 - \lambda)^2} \left(\frac{\lambda_1 - \lambda_j}{\lambda_1 - \lambda} \right) \right\} \\ &+ \sum_{i=2}^{n} \sum_{j>i} \left\{ \frac{\bar{a}_i^2 \bar{a}_j^2}{(\lambda_i - \lambda)^4 (\lambda_j - \lambda)^2} \left(\frac{\lambda_j - \lambda_i}{\lambda_j - \lambda} \right) + \frac{\bar{a}_j^2 \bar{a}_i^2}{(\lambda_j - \lambda)^4 (\lambda_i - \lambda)^2} \left(\frac{\lambda_i - \lambda_j}{\lambda_i - \lambda} \right) \right\}.\end{aligned}$$

Recall, from Assumption 2, that $\lambda_1 < \lambda_2$. Thus, the first sum is strictly positive for $\lambda \in (\lambda_1, \lambda_2)$, where we use the fact that there exists $j \geq 2$ such that $\bar{a}_1 \bar{a}_j \neq 0$. We next claim, for $2 \leq i \leq n$ and $i < j \leq n$, that

$$\frac{\bar{a}_i^2 \bar{a}_j^2}{(\lambda_i - \lambda)^4 (\lambda_j - \lambda)^2} \left(\frac{\lambda_j - \lambda_i}{\lambda_j - \lambda} \right) + \frac{\bar{a}_j^2 \bar{a}_i^2}{(\lambda_j - \lambda)^4 (\lambda_i - \lambda)^2} \left(\frac{\lambda_i - \lambda_j}{\lambda_i - \lambda} \right) \tag{3.70}$$

is positive. Indeed, if $\bar{a}_i \bar{a}_j = 0$ or $\lambda_i = \lambda_j$, it is trivial. Otherwise, $\bar{a}_i \bar{a}_j \neq 0$ and $\lambda_i < \lambda_j$, and (3.70) is positive if and only if

$$\frac{1}{(\lambda_i - \lambda)^4 (\lambda_j - \lambda)^3} - \frac{1}{(\lambda_j - \lambda)^4 (\lambda_i - \lambda)^3}$$

is positive. Rewriting the last expression, we obtain

$$\frac{\lambda_j - \lambda_i}{(\lambda_i - \lambda)^4 (\lambda_j - \lambda)^4},$$

which is positive. Thus (3.69) is strictly positive and $\|x(t)\|^2$ is a strictly convex function for $t \in \mathcal{T}$. □

3.3.1. Bounds on λ^* and t^*. For this section, we assume a local-nonglobal minimizer of problem (1.2) exists. Our algorithm is based on finding a root t^* to $\|x(t)\|^2 - 1$ and we need initial bounds on t^*.

LEMMA 3.24. *Suppose x^* is a local-nonglobal minimizer of problem* (1.2) *with a corresponding Lagrange multiplier λ^* that satisfies* (2.1)*. Let t^* be the unique solution to $\lambda_2(D(t)) = \lambda^*$. Then*

$$t^* \in (-\|a\| + \lambda_1(A), \|a\| + \lambda_2(A)).$$

Proof. By definition of t^*, $\lambda_2(D(t^*)) \in (\lambda_1(A), \lambda_2(A))$ and thus $t^* \in \mathcal{T}$. Hence, for $t = t^*$, dividing (3.63a) by y_0 gives

$$t^* = a^T x^* + \lambda_2(D(t^*)) = a^T x^* + \lambda^*.$$

The conclusion follows from $\lambda^* \in (\lambda_1(A), \lambda_2(A))$ and $-\|a\| \leq a^T x^* \leq \|a\|$. □

COROLLARY 3.25. *Suppose x^* is a local-nonglobal minimizer of problem* (1.2)*; then* (2.1) *holds with $\lambda^* \in [\lambda_2(D(-\|a\| + \lambda_1(A))), \lambda_2(D(\|a\| + \lambda_2(A)))]$.*

Proof. Recall, from Lemma 3.20, that $\lambda_2(D(t))$ is an increasing function. □

3.3.2. The algorithm. We now describe our second algorithm for either computing a possible local-nonglobal minimizer of problem (1.2) or declaring that such a candidate does not exist. From Theorem 3.22, in order to compute the local-nonglobal minimizer we need to find the smallest root t^* of $\|x(t)\|^2 - 1$. Since the latter function is strictly convex for $t \in \mathcal{T}$ and since we have a lower bound on t^*, the algorithm we propose is similar to Algorithm 3.1 and is essentially the secant method. To simplify our analysis, let $r(t) := \|x(t)\|^2 - 1$.

```
ALGORITHM 3.2.
1. INITIALIZATION.
  1.1. Let $t_L = -\|a\| + \lambda_1(A)$, $t_U = \|a\| + \lambda_2(A)$, $t_0 = t_L - 0.1$, $t_1 = t_L$, $k = 1$.
  1.2. If $\lambda_2(D(t_L)) = \lambda_2(A)$ or if $\frac{r(t_1)-r(t_0)}{t_1-t_0} \geq 0$, LNGM $= 0$, else LNGM $= 1$.
2. ITERATION. While LNGM $= 1$ and $\|x(t_k)\| \neq 1$, do
  2.1. $t_{k+1} = t_k - \frac{r(t_k)(t_k-t_{k-1})}{r(t_k)-r(t_{k-1})}$.
  2.2. If $\lambda_2(D(t_{k+1})) = \lambda_2(A)$, $\frac{r(t_{k+1})-r(t_k)}{t_{k+1}-t_k} \geq 0$, or $t_{k+1} > t_U$, then LNGM $= 0$.
  2.3. $k = k + 1$.
```

The convergence results of Algorithm 3.2 and their proofs are identical to those of Theorem 3.14 and Corollary 3.15. We again suppose Assumption 3 holds.

THEOREM 3.26. *The sequence $\{t_k\}$ produced by Algorithm* 3.1 *either converges to t^* such that $x(t^*)$ is a local-nonglobal minimizer of problem* (1.2) *or there does not exist a local-nonglobal minimizer of problem* (1.2) *and LNGM is set to* 0. □

COROLLARY 3.27. *Suppose x^* is a local-nonglobal minimizer of problem* (1.2) *with a corresponding Lagrange multiplier λ^* that satisfies* (2.1)*. Let t^* be the unique solution to $\lambda_2(D(t)) = \lambda^*$. Then if $r'(t^*) > 0$, the sequence $\{t_k\}$ produced by Algorithm* 3.2 *converges to t^* superlinearly and $x(t^*)$ is a strict local-nonglobal minimizer of problem* (1.2)*.* □

3.3.3. The relation between a local-nonglobal minimizer and $m(t)$. Recall that

$$m(t) := 2\lambda_2(D(t)) - t.$$

Let $t \in \mathcal{T}$. From Lemma 3.1 and (3.61), $\lambda_2(D(t))$ has multiplicity one. Hence m is differentiable for $t \in \mathcal{T}$. Therefore (see, e.g., [12])

$$m'(t) = 2y_0^2 - 1. \tag{3.71}$$

Just as in Lemma 3.17, our next lemma shows that $\|x(t)\|$ is related to the first derivative of m.

Lemma 3.28. *Let $t \in \mathcal{T}$. Then*

$$\|x(t)\| > \ (=,<)\ 1 \iff m'(t) < \ (=,>)\ 0. \tag{3.72}$$

Proof. For $t \in \mathcal{T}$, we have

$$\|x(t)\|^2 = \frac{z^2}{y_0^2} = \frac{1-y_0^2}{y_0^2} = \frac{1-\frac{m'(t)+1}{2}}{\frac{m'(t)+1}{2}} = \frac{1-m'(t)}{1+m'(t)}, \tag{3.73}$$

where the second equality follows from y being the unit norm and where the third equality follows from (3.71). This proves our result since, for $t \in \mathcal{T}$, $m'(t) \in (-1,1]$ and since the function $w(x) := \frac{1-x}{1+x}$ is strictly decreasing with $w(0) = 1$. □

Following are two theorems, analogous to Theorems 3.18 and 3.19, which relate a local-nonglobal minimizer to the function m. The first one states that if a local-nonglobal minimizer of problem (1.2) exists, then there exists t^* such that the first and second order optimality conditions for a local minimizer of m are satisfied. The second one is almost its converse: if the first and second order sufficient optimality conditions for a local minimizer of m are satisfied at some t^*, then $x(t^*)$ is the local-nonglobal minimizer of problem (1.2).

Theorem 3.29. *Suppose x^* is a local-nonglobal minimizer of problem* (1.2) *with a corresponding Lagrange multiplier λ^* that satisfies* (2.1). *Let t^* be the unique solution to $\lambda_2(D(t)) = \lambda^*$. Then $m'(t^*) = 0$ and $m''(t^*) \geq 0$.*

Proof. By Lemma 2.2, $\lambda_2(D(t^*)) \in (\lambda_1(A), \lambda_2(A))$ and thus $t^* \in \mathcal{T}$. Theorem 3.22 gives $x^* = x(t^*)$. The fact that $m'(t^*) = 0$ follows from the feasibility of x^* and from Lemma 3.28.

By (3.62) and (3.66), and since, by Theorem 2.5, $\varphi'(\lambda^*) \leq 0$, we obtain

$$\frac{d\|x(t^*)\|^2}{dt} = \frac{d\varphi(\lambda_2(D(t^*)))}{d\lambda}\ \lambda_2'(D(t^*)) \leq 0. \tag{3.74}$$

If $m''(t^*) < 0$, then $m'(t^* - h) > 0$ for $h > 0$ small enough and, using Lemma 3.28, we deduce $\|x(t^* - h)\| < 1$. Thus, since $\|x(t^*)\| = 1$,

$$\frac{d\|x(t^*)\|^2}{dt} = \lim_{h \to 0} \frac{\|x(t^*)\|^2 - \|x(t^* - h)\|^2}{h} \geq 0. \tag{3.75}$$

Inequalities (3.74) and (3.75) give

$$\frac{d\|x(t^*)\|^2}{dt} = 0.$$

It follows then from (3.74) and $\lambda_2'(D(t^*)) < 0$ that $\varphi'(\lambda_2(D(t^*))) = 0$. From (2.6c), φ is strictly convex over the interval $(\lambda_1(A), \lambda_2(A))$ and thus $\lambda_2(D(t^*))$ is its strict minimizer. By (3.65), the following inequality thus holds:

$$\|x(t)\| \geq \|x(t^*)\| = 1 \ \text{ for } t \in \mathcal{T}.$$

This contradicts $\|x(t^* - h)\| < 1$ for $h > 0$ small enough. Thus $m''(t^*) \geq 0$. □

THEOREM 3.30. *Suppose $t^* \in \mathbb{R}$ satisfies $m'(t^*) = 0$ and $m''(t^*) > 0$; then $x(t^*)$ is a strict local-nonglobal minimizer of* (1.2) *with Lagrange multiplier $\lambda^* := \lambda_2(D(t^*))$.*

Proof. From Lemma 3.20, $m''(t^*) \neq 0$ implies $\lambda_2(D(t^*)) \in (\lambda_1(A), \lambda_2(A))$, i.e., $t^* \in \mathcal{T}$. Therefore $x(t^*)$ is well defined, and if we let $x^* := x(t^*)$ and $\lambda^* := \lambda_2(D(t^*))$, then by (3.18), the stationarity condition (2.1) is satisfied. Feasibility of x^* follows from Lemma 3.28. If we can further show that $\varphi'(\lambda^*) < 0$, then the result follows from item 3 of Theorem 2.5.

Since $m''(t^*) > 0$, then $m'(t^* + h) > m'(t^*) = 0$ for $h > 0$ small enough. By Lemma 3.28, this implies $\|x(t^* + h)\| < 1$ and thus

$$\frac{d\|x(t^*)\|^2}{dt} = \lim_{h\to 0} \frac{\|x(t^* + h)\|^2 - \|x(t^*)\|^2}{h} \leq 0. \tag{3.76}$$

By an argument similar to one that appears in the proof of Theorem 3.29, we conclude that the inequality in (3.76) holds strictly. From (3.62) and (3.66), we deduce $\varphi'(\lambda^*) < 0$. □

4. Conclusion.

We have considered two algorithms for computing the local-nonglobal minimizer of problem (1.2). The first algorithm builds on an extremal ellipsoid-based approach that recasts problem (1.2) as problem (3.16). This approach has the particularity of bringing a geometric interpretation common to both types of minimizers (local-nonglobal and global): each minimizer lies in the intersection of the unit sphere with an ellipsoid locally contained in the unit ball at the minimizer. As we have seen in section 3.1, this allows a geometric view between local-nonglobal and global minimizers when the trust-region radius tends to infinity. The algorithm was largely motivated by Corollary 3.4 on the interlacing of eigenvalues between a parametric matrix and the matrix B. Similarly, Cauchy's inequalities, here Lemma 3.1, motivated the second algorithm, which builds on the approach of Rendl and Wolkowicz [21] that recasts problem (1.2) as problem (3.2).

Both algorithms are based on applying the secant method, respectively, to the strictly convex functions $\|x(\gamma)\|^2 - 1$ and $\|x(t)\|^2 - 1$. Although the author was not able to prove it, it seems the functions $\|x(\gamma)\| - 1$ and $\|x(t)\| - 1$ are strictly convex as well, and some preliminary results (which are not reported here) tend to show convergence is enhanced when the secant method is used to find a root of the latter functions. This is probably due to the fact that the functions are in some sense less nonlinear. Obviously, this constitutes material for future research.

For each algorithm, the main computing effort at each step lies in approximating the first two eigenvalues of a parametric matrix. We have coded Algorithms 3.1 and 3.2 using MATLAB 6.1 and used as a black box MATLAB's implementation of ARPACK [13, 14, 24], the function `eigs`, to compute the required eigenvalues and corresponding eigenvectors. All that is required are matrix-vector multiplications, and thus our algorithms are able to exploit the sparsity of the matrix A in large scale problems.

We compared the computation times of each algorithms on randomly constructed problems of the type (1.2) for which we made sure a local-nonglobal minimizer existed. The algorithms computed, respectively, a root of the functions $\|x(\gamma)\|^2 - 1$ and $\|x(t)\|^2 - 1$. We stopped when the image of an iterate had an absolute value less than `1e-12`. For example, Algorithm 3.1 stops when $|\|x(\gamma_k)\|^2 - 1| < 1e - 12$. We tested our algorithms on random problems of dimensions varying from $n = 125$ to $n = 2000$

Table 4.1

Average number of iterations, computation times, and matrix-vector multiplications of Algorithms 3.1 *and* 3.2 *in function of the dimension n of the problem* (1.2). *The density of the matrix A is* $\log(n)/n$, *a function of its dimension.*

	Iterations		Computation time		Matrix-vector ×	
Dimension	Alg. 3.1	Alg. 3.2	Alg. 3.1	Alg. 3.2	Alg. 3.1	Alg. 3.2
125	17.4	16.4	2.25	1.84	1899.8	1569.0
250	15.8	15.0	3.23	2.92	2645.6	2356.6
500	17.2	16.2	5.19	3.72	3579.2	2488.0
1000	20.4	19.6	8.09	8.13	4082.6	3986.4
2000	20.4	19.4	17.38	15.39	5246.2	4446.0

using a Pentium 4 at 1.8GHz with 256MB of memory. For each given dimension we computed the average number of iterations, matrix-vector multiplications, and computation time (cpu in seconds), taken over five random problems, in order to compute a local-nonglobal minimizer up to the accuracy above mentioned. The results are illustrated in Table 4.1.

The results reveal that Algorithm 3.2 takes in average one less iteration to converge and clearly needs fewer matrix-vector multiplications. This results in smaller computation times for Algorithm 3.2. Thus the results tend to indicate we should prefer the latter algorithm to Algorithm 3.1. Somehow this is not surprising since for each iteration of Algorithm 3.1 a generalized eigenvalue problem needs to be solved, whereas Algorithm 3.2 requires only the solution of a standard eigenvalue problem.

We conclude with two final remarks. First, the algorithms proposed in this paper are a first step toward solving large problems of the type (1.3), in particular in the case where the constraints are two Euclidean balls. However, the case where the two constraints are binding at the optimum remains a challenge, even for small dimensions. Second, we have seen that the theory behind both algorithms we presented is induced by the theory behind algorithms for computing the global minimizer of trust-region subproblems. It would be interesting to see if this could be done for other trust-region subproblem algorithms as well.

Appendix. We briefly discuss how we can modify Algorithms 3.1 and 3.2 in order to compute a local-nonglobal minimizer of problem (1.1).

It is easy to see that if x^* is a local-nonglobal minimizer of problem (1.1), then $\|x^*\| = 1$ must hold. Hence x^* is necessarily a local-nonglobal minimizer of problem (1.2). From the standard necessary optimality conditions it has a negative Lagrange multiplier $\lambda^* \leq 0$. Furthermore, it is shown in [15] that in fact strict inequality holds, i.e., $\lambda^* < 0$. Similarly to item 1 of Theorem 3.8 we have the following theorem.

Theorem A.1. *If x^* is a local-nonglobal minimizer of problems* (1.1), *then* (2.1) *holds with $\lambda^* \in (\lambda_1(A), \lambda_2(A))$. Let γ^* be the unique solution to $\lambda(\gamma) + \bar{\lambda} = \lambda^*$; then $x^* = x(\gamma^*)$, $\frac{d\|x(\gamma^*)\|^2}{d\gamma} \geq 0$, and $\lambda(\gamma^*) + \bar{\lambda} < 0$.* □

Recall that as long as $LNGM = 1$ the sequence $\{\gamma_k\}$ generated by Algorithm 3.1 is decreasing and that the function $\lambda(\gamma)+\bar{\lambda}$ is decreasing. Therefore, in Algorithm 3.1, if at some iteration k,

$$\lambda(\gamma_k) + \bar{\lambda} \geq 0, \tag{A.1}$$

we may deduce from Theorem A.1 that problem (1.1) does not have a local-nonglobal minimizer. Thus in order to modify Algorithm 3.1 for computing a local-nonglobal minimizer of problem (1.1), we may set the boolean parameter $LNGM$ to 0 whenever (A.1) holds. A similar remark holds for Algorithm 3.2.

Acknowledgment. The author thanks the anonymous referees for their valuable comments which have improved the presentation of the paper.

REFERENCES

[1] K. Anstreicher and H. Wolkowicz, *On Lagrangian relaxation of quadratic matrix constraints*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 41–55.

[2] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, SIAM, Philadelphia, 2001.

[3] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM Stud. Appl. Math. 15, SIAM, Philadelphia, 1994.

[4] A. L. Cauchy, *Sur l'équation à l'aide de laquelle on détermine les ingalités séculaires des mouvements des planètes*, in Oeuvres Complètes (II^e Série), Vol. 9, 1829.

[5] M. R. Celis, J. E. Dennis, and R. A. Tapia, *A trust region strategy for nonlinear equality constrained optimization*, in Numerical Optimization, 1984 (Boulder, CO, 1984), SIAM, Philadelphia, 1985, pp. 71–82.

[6] C. Fortin and H. Wolkowicz, *The trust region subproblem and semidefinite programming*, Optim. Methods Softw., 19 (2004), pp. 41–67.

[7] D. M. Gay, *Computing optimal locally constrained steps*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 186–197.

[8] G. H. Golub and C. F. Van Loan, *Matrix Computation*, 3rd ed., The Johns Hopkins University Press, Baltimore, MA, 1996.

[9] N. I. M. Gould, S. Lucidi, M. Roma, and P. L. Toint, *Solving the trust-region subproblem using the Lanczos method*, SIAM J. Optim., 9 (1999), pp. 504–525.

[10] M. Heinkenschloss, *Mesh independence for nonlinear least squares problems with norm constraints*, SIAM J. Optim., 3 (1993), pp. 81–117.

[11] M. Heinkenschloss, *On the solution of a two ball trust region subproblem*, Math. Programming, 64 (1994), pp. 249–276.

[12] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[13] R. B. Lehoucq and D. C. Sorensen, *Deflation techniques for an implicitly restarted Arnoldi iteration*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 789–821.

[14] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide*, Software Environ. Tools 6, SIAM, Philadelphia, 1998.

[15] S. Lucidi, L. Palagi, and M. Roma, *On some properties of quadratic programs with a convex quadratic constraint*, SIAM J. Optim., 8 (1998), pp. 105–122.

[16] J. M. Martínez, *Local minimizers of quadratic functions on Euclidean balls and spheres*, SIAM J. Optim., 4 (1994), pp. 159–176.

[17] J. M. Martínez and S. A. Santos, *A trust-region strategy for minimization on arbitrary domains*, Math. Programming, 68 (1995), pp. 267–301.

[18] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[19] A. Neumaier, *Introduction to Numerical Analysis*, Cambridge University Press, Cambridge, UK, 2001.

[20] M. J. D. Powell and Y. Yuan, *A trust region algorithm for equality constrained optimization*, Math. Programming, 49 (1990/91), pp. 189–211.

[21] F. Rendl and H. Wolkowicz, *A semidefinite framework for trust region subproblems with applications to large scale minimization*, Math. Programming, 77 (1997), pp. 273–299.

[22] M. Rojas, S. A. Santos, and D. C. Sorensen, *A new matrix-free algorithm for the large-scale trust-region subproblem*, SIAM J. Optim., 11 (2000), pp. 611–646.

[23] D. C. Sorensen, *Newton's method with a model trust region modification*, SIAM J. Numer. Anal., 19 (1982), pp. 409–426.

[24] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.

[25] D. C. Sorensen, *Minimization of a large-scale quadratic function subject to a spherical constraint*, SIAM J. Optim., 7 (1997), pp. 141–161.

[26] J. F. Sturm and S. Zhang, *On cones of nonnegative quadratic functions*, Math. Oper. Res., 28 (2003), pp. 246–267.

[27] P. D. Tao and L. T. H. An, *Difference of convex functions optimization algorithms (DCA) for globally minimizing nonconvex quadratic forms on Euclidean balls and spheres*, Oper. Res. Lett., 19 (1996), pp. 207–216.

[28] P. D. Tao and L. T. H. An, *A d.c. optimization algorithm for solving the trust-region subproblem*, SIAM J. Optim., 8 (1998), pp. 476–505.

[29] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.

[30] Y. Ye and S. Zhang, *New Results on Quadratic Optimization*, Technical report, Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, 2001.

[31] Y. Yuan, *On a subproblem of trust region algorithms for constrained optimization*, Math. Programming, 47 (1990), pp. 53–63.

SIAM J. OPTIM.
Vol. 16, No. 1, pp. 297–309

ERROR ESTIMATES IN THE OPTIMIZATION OF DEGREE TWO POLYNOMIALS ON A DISCRETE HYPERCUBE[*]

M. MARSHALL[†]

Abstract. This paper considers the distribution of values $Q(x)$, $x \in \{-1,1\}^n$, where Q is a quadratic form in n variables with real coefficients. Error estimates are established for approximations of the maximum and minimum values of Q on $\{-1,1\}^n$ which can be obtained by semidefinite programming. Bounds are given involving the sum of the absolute values of the off-diagonal entries. Other bounds are given which are useful in the case of extreme skewness. Used in conjunction with earlier bounds of Nesterov in [*Optim. Methods Softw.*, 9 (1998), pp. 141–160], these new bounds lead to improvements on the bound given by the trace. The trigonometric description of the maximum and minimum given in [*Optim. Methods Softw.*, 9 (1998), pp. 141–160], which is based on the rounding argument introduced by Goemans and Williamson in [*J. Assoc. Comput. Mach.*, 6 (1995), pp. 1115–1145], is a major tool in obtaining these bounds.

Key words. semidefinite programming, MAX-CUT algorithm

AMS subject classifications. 14P99, 15A63, 90C22

DOI. 10.1137/030602228

1. Introduction.

Let Q be a polynomial in n variables with real coefficients, and let $S \subseteq \mathbb{R}^n$ be a basic closed semialgebraic set. Lasserre's algorithm [3] produces an increasing sequence of lower bounds for Q on S computable via semidefinite programming which, in case S is compact, converges to the exact minimum of Q on S. In [4] a refinement of Lasserre's algorithm is described which takes into account the fact that S may have dimension less than n. This involves computation in the factor ring $\mathbb{R}[x]/\mathfrak{a}$ using Gröbner basis techniques, where $\mathfrak{a}$ is the ideal of polynomials vanishing on S. Many times the sequence converges rapidly. Often the first or second term in the sequence is already close to the exact minimum. Still, there is no general theory in this regard.

In particular, one would like to be able to estimate the accuracy of the first term in the sequence. In the present paper we consider this question in case Q is of degree 2 and S is the discrete hypercube $\{-1,1\}^n$. In this case encouraging results have been obtained already by a number of people [5], [6], [8], [9], [10], [11], following up on the ground-breaking work of Goemans and Williamson in [1].

When S is $\{-1,1\}^n$ the ideal $\mathfrak{a}$ is generated by x_i^2-1, $i=1,\dots,n$. The factor ring $\mathbb{R}[x]/\mathfrak{a}$ is 0-dimensional with basis as a vector space over $\mathbb{R}$ consisting of all products $\prod_{i\in I} x_i$, where I is a subset of $\{1,\dots,n\}$. We consider the problem of minimizing (or maximizing) a degree two polynomial $Q \in \mathbb{R}[x]$ on $\{-1,1\}^n$. More precisely, we consider an approximation to this minimum (or maximum) which, in terms of the algorithm described in [4], is just the first term in the sequence of approximations converging to the exact value, and we examine the accuracy of this approximation.

Since minimizing (or maximizing) Q on $\{-1,1\}^n$ is equivalent to minimizing (or maximizing) the associated quadratic form $x_0^2 Q(\frac{x_1}{x_0},\dots,\frac{x_n}{x_0})$ in $n+1$ variables

[*] Received by the editors December 4, 2003; accepted for publication (in revised form) January 10, 2005; published electronically September 15, 2005. This research was supported by NSERC of Canada and by the universities of Konstanz, Paris 6, and Paris 7.
http://www.siam.org/journals/siopt/16-1/60222.html

[†] Research Unit Algebra & Logic, Department of Computer Science, University of Saskatchewan, Saskatoon, SK Canada, S7N 5E6 (marshall@math.usask.ca).

$x_0, \dots, x_n$ on $\{-1,1\}^{n+1}$, we may as well assume from the start that Q is a quadratic form. We identify Q with its associated symmetric matrix, so $Q(x) = x^tQx$. We define

$$Q_* = \min\{Q(x) \mid x \in \{-1,1\}^n\},$$
$$Q^* = \max\{Q(x) \mid x \in \{-1,1\}^n\}.$$

One can associate to Q the graph with vertices $V = \{1,\dots,n\}$ and edges $E = \{(i,j) : i < j, Q_{ij} \neq 0\}$. Computation of Q_* (resp., Q^*) can be viewed as a "weighted" version of the MAX-CUT problem considered in [1], where Q_{ij} is the "weight" attached to the edge (i,j).

The approximations of Q_* and Q^* that we are dealing with are

$$Q_+ := \min\{\langle Q, X\rangle \mid X \text{ is PSD}, \ X_{ii} = 1, i = 1,\dots,n\},$$
$$Q^+ := \max\{\langle Q, X\rangle \mid X \text{ is PSD}, \ X_{ii} = 1, i = 1,\dots,n\}.$$

Here, $\langle Q, X\rangle := \sum_{i,j} Q_{ij}X_{ij}$. It is clear that $Q_+ \leq Q_*$ (and, similarly, that $Q^* \leq Q^+$): If $x \in \{-1,1\}^n$, define X by $X_{ij} = x_ix_j$. Then X is positive semidefinite (PSD), $X_{ii} = 1$ for $i = 1,\dots,n$, and $\langle Q, X\rangle = \sum_{i,j} Q_{ij}x_ix_j = x^tQx = Q(x)$. In summary we have

$$Q_+ \leq Q_* \leq Q^* \leq Q^+.$$

We also have the following dual description of Q_+ (and of Q^+).

Theorem 1.1. *$Q_+ = \overline{Q}_+$ and $Q^+ = \overline{Q}^+$, where*

$$\overline{Q}_+ = \max\left\{\lambda \mid \exists y \in \mathbb{R}^n, \ \lambda = \sum y_i, \ Q - \mathrm{Diag}(y) \textit{ is PSD}\right\},$$
$$\overline{Q}^+ = \min\left\{\lambda \mid \exists y \in \mathbb{R}^n, \ \lambda = \sum y_i, \ \mathrm{Diag}(y) - Q \textit{ is PSD}\right\}.$$

Here, $\mathrm{Diag}(y)$ denotes the diagonal matrix with diagonal entries $y_1,\dots,y_n$. Computation of Q_+ (resp., Q^+) is a semidefinite programming problem. Computation of $\overline{Q}_+$ (resp., of $\overline{Q}^+$) is the dual semidefinite programming problem. The inequality $Q_+ \geq \overline{Q}_+$ (resp., $\overline{Q}^+ \geq Q^+$) is based on the fact that if A, B are PSD, then $\langle A, B\rangle \geq 0$: Theorem 1.1 asserts that the duality gap $Q_+ - \overline{Q}_+$ (resp., $\overline{Q}^+ - Q^+$) is zero. This can be proved in various ways; e.g., see [4], [5], or [6] for more general results.

In [5], [6] Nesterov obtains bounds for Q_* (resp., Q^*) in terms of Q_+, Q^+, and $\mathrm{tr}(Q) := \sum_{i=1}^n Q_{ii}$. We recall these bounds in section 3; see Theorems 3.2 and 3.3. The bound given by Theorem 3.3 is always better than the bound given by Theorem 3.2. In section 4 we give bounds which involve $\sum_{i\neq j} |Q_{ij}|$. The first (see Theorem 4.1) is a simple generalization of the result in [1]. A particular case of this already appears in [11]. The second (see Theorem 4.2) is new. The bound provided by Theorem 4.2 is better than the bound provided by Theorem 3.3 in cases where $Q^+ - \mathrm{tr}(Q)$ (resp., $\mathrm{tr}(Q) - Q_+$) is "sufficiently close" to $\sum_{i\neq j} |Q_{ij}|$. This occurs, for example, if $Q_{ij} \geq 0$ (resp., $Q_{ij} \leq 0$) for $i \neq j$. We also compare Theorems 4.1 and 4.2 and explain how Theorem 4.2 predicts better accuracy of the MAX-CUT algorithm in [1] when the output is either more than ≈ 86.6% of the total number of edges or less than ≈ 67.0% of the total number of edges.

For Q not diagonal, the ratio $r = \frac{Q^+ - \mathrm{tr}(Q)}{Q^+ - Q_+}$ lies somewhere in the closed interval $[\frac{1}{n}, \frac{n-1}{n}]$. This follows from Theorem 2.6. When r is not too large, the bound for Q_* given by Theorem 3.3 is significantly better than the trivial bound $Q_* \leq \mathrm{tr}(Q)$. If

$Q_{ij} \geq 0$ for $i \neq j$, one can improve on this using Theorem 4.2. At the other extreme, when r is sufficiently close to $\frac{n-1}{n}$, other bounds come into play (see Corollary 5.5) which are also significantly better than the trivial bound. In the intermediate case nothing much seems to be known. It may be that no significant improvement on the trivial bound is possible in this case; see Example 5.3. The best we are able to show in general is that, for large n, $\frac{\operatorname{tr}(Q)-Q_*}{\operatorname{tr}(Q)-Q_+}$ is bounded away from zero by a function of the form $\frac{C}{\sqrt[3]{n}}$, where C is a constant; see Theorem 5.6.

2. Elementary observations. In studying the distribution $Q(x)$, $x \in \{-1,1\}^n$, it is natural to consider the mean and standard deviations. Denote by $\operatorname{tr}(Q)$ the trace of Q, i.e., $\operatorname{tr}(Q) = \langle Q, I\rangle = \sum_i Q_{ii}$.

LEMMA 2.1. *The mean value of Q on $\{-1,1\}^n$ is equal to* $\operatorname{tr}(Q)$.

Proof. The proof is trivial. In the sum

$$\sum_{x\in\{-1,1\}^n} Q(x) = \sum_{x\in\{-1,1\}^n} \sum_{i,j} Q_{ij}x_ix_j$$

the terms with $i \neq j$ cancel. □

It follows from Lemma 2.1 (also see [5, Cor. 2.4]) that $Q_* \leq \operatorname{tr}(Q) \leq Q^*$. We refer to the bound $Q_* \leq \operatorname{tr}(Q)$ (resp., $Q^* \geq \operatorname{tr}(Q)$) as the *trivial bound* for Q_* (resp., Q^*).

LEMMA 2.2. *The standard deviation of Q on $\{-1,1\}^n$ is*

$$2\sqrt{\sum_{i<j} Q_{ij}^2} = \sqrt{2\sum_{i\neq j} Q_{ij}^2}.$$

Proof. The proof is similar to the proof of Lemma 2.1 and is omitted. □

We also have the following lower (resp., upper) bound for Q_+ (resp., Q^+).

THEOREM 2.3.

$$Q_+ \geq \operatorname{tr}(Q) - \sum_{i\neq j} |Q_{ij}|,$$
$$Q^+ \leq \operatorname{tr}(Q) + \sum_{i\neq j} |Q_{ij}|.$$

Proof. Adding $\frac{|Q_{ij}|}{2}(x_i^2 + x_j^2)$ to $Q_{ij}x_ix_j$ for $j \neq i$ yields the perfect square

$$\frac{|Q_{ij}|}{2}(x_i \pm x_j)^2.$$

Consequently the quadratic form $Q(x) - \sum_{i=1}^n (Q_{ii} - \sum_{j\neq i} |Q_{ij}|)x_i^2$ is PSD, so $\overline{Q}_+ \geq \operatorname{tr}(Q) - \sum_{i\neq j} |Q_{ij}|$. The first assertion follows from this using $Q_+ \geq \overline{Q}_+$ (the easy half of Theorem 1.1). The second assertion follows from the first by replacing Q by $-Q$. □

COROLLARY 2.4 (see [11, Thm. 2]). *If $Q_{ij} \geq 0$ (resp., $Q_{ij} \leq 0$) for $i \neq j$, then*

$$Q^* = Q^+ = \operatorname{tr}(Q) + \sum_{i\neq j} |Q_{ij}|,$$
$$\left(\textit{resp., } Q_* = Q_+ = \operatorname{tr}(Q) - \sum_{i\neq j} |Q_{ij}|\right).$$

In particular, if Q is diagonal, then $Q_+ = Q_* = \operatorname{tr}(Q) = Q^* = Q^+$.

Proof. Since $Q(1,\dots,1)=\sum_{i,j}Q_{ij}$, this is immediate from Theorem 2.3. □

Lemma 2.5.

$$Q_* \le \operatorname{tr}(Q) - 2\max\{|Q_{ij}| \mid i\neq j\},$$
$$Q^* \ge \operatorname{tr}(Q) + 2\max\{|Q_{ij}| \mid i\neq j\}.$$

In particular, Q not diagonal $\Rightarrow Q_* < \operatorname{tr}(Q) < Q^*$.

Proof. We prove that if $Q_* \ge \operatorname{tr}(Q)-\delta$ (resp., $Q^* \le \operatorname{tr}(Q)+\delta$), then $|Q_{ij}| \le \frac{\delta}{2}$ for all $i\neq j$. Replacing Q by $Q' = Q - \operatorname{Diag}(Q_{11},\dots,Q_{nn})$, we can assume the diagonal entries of Q are 0. We can assume $n\ge 2$ and, after reindexing, that $i=1$, $j=2$. If $n=2$, then $Q(x) = 2Q_{12}x_1x_2$ and the hypothesis $Q_* = -2|Q_{12}| \ge -\delta$ implies $|Q_{12}| \le \frac{\delta}{2}$, as required. If $n\ge 3$, use the identity

$$Q(x_1,\dots,x_{n-1},0) = \frac{1}{2}(Q(x_1,\dots,x_{n-1},x_n) + Q(x_1,\dots,x_{n-1},-x_n))$$

and proceed by induction on n. □

The invariants Q_+ and Q^+ provide not only upper and lower bounds for the distribution $Q(x)$, $x\in\{-1,1\}^n$, but also, by comparing the relative magnitude of $Q^+ - \operatorname{tr}(Q)$ and $\operatorname{tr}(Q) - Q_+$, some rough measure of the skewness of the distribution. Lemma 2.5 implies that the ratio $\frac{Q^+-\operatorname{tr}(Q)}{\operatorname{tr}(Q)-Q_+}$ is well-defined and positive for Q nondiagonal.

Theorem 2.6. *For Q nondiagonal,*

$$\frac{1}{n-1} \le \frac{Q^+ - \operatorname{tr}(Q)}{\sum_{i\neq j}|Q_{ij}|} \le \frac{Q^+ - \operatorname{tr}(Q)}{\operatorname{tr}(Q) - Q_+} \le \frac{\sum_{i\neq j}|Q_{ij}|}{\operatorname{tr}(Q) - Q_+} \le n-1.$$

Proof. The middle inequalities are immediate from Theorem 2.3. The first inequality follows from the last, replacing Q by $-Q$, so we concentrate on the last inequality. One reduces easily to the case where $Q_+ = 0$ and Q is PSD. This is just a matter of replacing Q by $Q' = Q - \operatorname{Diag}(y)$, where $y\in\mathbb{R}^n$ is chosen so that $\sum_i y_i = Q_+$ and Q' is PSD. Scaling, we can assume $\operatorname{tr}(Q)=1$. We want to show $\sum_{i\neq j}|Q_{ij}| \le n-1$. Since Q is PSD there exist vectors $w_1,\dots,w_n$ in $\mathbb{R}^n$ such that $Q_{ij} = \langle w_i, w_j\rangle$. (Use the spectral theorem to decompose Q as $Q = D^tD$ and take $w_1,\dots,w_n$ to be the columns of D.) Thus $\sum_i \|w_i\|^2 = \sum_i Q_{ii} = \operatorname{tr}(Q) = 1$. By the Cauchy–Schwarz inequality, $|\langle w_i,w_j\rangle| \le \|w_i\|\|w_j\|$. Thus

$$\sum_{i\neq j}|Q_{ij}| \le \sum_{i\neq j}\|w_i\|\|w_j\| = \left(\sum_i \|w_i\|\right)^2 - \sum_i \|w_i\|^2 = \left(\sum_i \|w_i\|\right)^2 - 1.$$

We know from calculus that the maximum value of $f(x) = (\sum_i x_i)^2$ on the sphere $\sum_i x_i^2 = 1$ is n. The maximum is achieved at $x = \pm(\frac{1}{\sqrt{n}},\dots,\frac{1}{\sqrt{n}})$. This proves $\sum_{i\neq j}|Q_{ij}| \le n-1$ and completes the proof. □

Later we establish similar bounds for the ratio $\frac{Q^*-\operatorname{tr}(Q)}{\operatorname{tr}(Q)-Q_*}$; see section 5, Theorem 5.4, for the precise statement. The following example shows that the bounds given by Theorems 2.6 and 5.4 are best possible.

Example 2.7. Take

$$Q(x) = (x_1 + \cdots + x_n)^2.$$

Then $\mathrm{tr}(Q) = n$, $Q^+ = Q^* = n^2$, and Q_* is either 0 or 1 depending on whether n is even or odd. We claim that $Q_+ = 0$. Since Q is PSD we see that $Q_+ \geq 0$. Choose vectors $v_1, \ldots, v_n \in \mathbb{R}^n$ such that $\|v_i\| = 1$ and $v_1 + \cdots + v_n = 0$ (always possible if $n \geq 2$) and define X by $X_{ij} = \langle v_i, v_j \rangle$. Then X is PSD, $X_{ii} = 1$, and $\langle Q, X \rangle = \sum_{i,j} \langle v_i, v_j \rangle = \|v_1 + \cdots + v_n\|^2 = 0$. This proves the upper bounds given by Theorems 2.6 and 5.4 are the best possible. Similarly, looking at

$$Q(x) = -(x_1 + \cdots + x_n)^2,$$

we see that the lower bounds are also the best possible.

It follows by continuity that each number $t \in [\frac{1}{n-1}, n-1]$ is equal to $\frac{Q^+ - \mathrm{tr}(Q)}{\mathrm{tr}(Q) - Q_+}$ for some nondiagonal symmetric $n \times n$ matrix Q. For example, if $t \in [1, n-1]$, we can choose Q of the form $Q(x) = a(x_1 + \cdots + x_n)^2 + (1-a)(x_1 + x_2)^2$ for suitable $a \in [0,1]$. An analogous result holds for the ratio $\frac{Q^* - \mathrm{tr}(Q)}{\mathrm{tr}(Q) - Q_*}$.

3. Nesterov's error bounds.

The main technical tool used in establishing bounds for Q_* and Q^* is an alternate description of Q_* and Q^* proved in [5], which, in turn, is motivated by the probabilistic argument in [1].

THEOREM 3.1 (see [5, Thm. 3.1]).

$$Q_* = \min \left\{ \frac{2}{\pi} \langle Q, \arcsin[X] \rangle \mid X \textit{ is PSD, } X_{ii} = 1, i = 1, \ldots, n \right\},$$
$$Q^* = \max \left\{ \frac{2}{\pi} \langle Q, \arcsin[X] \rangle \mid X \textit{ is PSD, } X_{ii} = 1, i = 1, \ldots, n \right\}.$$

Here, $\arcsin[X]$ denotes the matrix with ij entry $\arcsin(X_{ij})$.

In [5] and [6] Nesterov uses Theorem 3.1 to obtain the following result as a special case of a more general result.

THEOREM 3.2 (see [5, Thm. 3.3]).

$$Q_* \leq \left(1 - \frac{2}{\pi}\right) Q^+ + \frac{2}{\pi} Q_+,$$
$$Q^* \geq \left(1 - \frac{2}{\pi}\right) Q_+ + \frac{2}{\pi} Q^+.$$

As explained in [5] and [6], it is possible to improve on Theorem 3.2 with a bound that takes the trivial bound into account. In the case we are considering here, the improvement reads as follows. For Q nondiagonal, define

$$r := \frac{Q^+ - \mathrm{tr}(Q)}{Q^+ - Q_+}, \quad r' := \frac{\mathrm{tr}(Q) - Q_+}{Q^+ - Q_+}. \tag{3.1}$$

Note: $r + r' = 1$ and $\frac{r}{r'} = \frac{Q^+ - \mathrm{tr}(Q)}{\mathrm{tr}(Q) - Q_+}$. By Theorem 2.6, $\frac{1}{n-1} \leq \frac{r}{r'} \leq n-1$, so $r, r' \in [\frac{1}{n}, \frac{n-1}{n}]$.

THEOREM 3.3 (see [5, Thm. 3.5]). *For Q nondiagonal,*

$$Q_* \leq (1 - \omega(r))Q^+ + \omega(r)Q_+,$$
$$Q^* \geq (1 - \omega(r'))Q_+ + \omega(r')Q^+.$$

Here r, r' are defined by (3.1), *and* $\omega(y) := \frac{2}{\pi}(\sqrt{1-y^2} + y\arcsin(y))$.

The function ω is increasing on $[0,1]$, $\omega(0) = \frac{2}{\pi}$, $\omega(1) = 1$. In particular, the bound given by Theorem 3.3 is always better than that given by Theorem 3.2. The bound given by Theorem 3.3 also improves on the trivial bound. This follows from the proof of Theorem 3.3 given in [5]. Also see section 5, Corollary 5.1(1).

4. Error bounds involving $\sum_{i\neq j}|Q_{ij}|$. Denote by μ the minimal value of the function $\frac{2}{\pi}\frac{x}{1-\cos x}$ on the interval $(0,\pi)$. $\mu \approx 0.8786$ is the well-known Goemans–Williamson approximation ratio. The proof of the following bound copies the argument given in [1].

THEOREM 4.1.

$$Q_* \leq (1-\mu)\Big(\operatorname{tr}(Q) + \sum_{i\neq j}|Q_{ij}|\Big) + \mu Q_+,$$

$$Q^* \geq (1-\mu)\Big(\operatorname{tr}(Q) - \sum_{i\neq j}|Q_{ij}|\Big) + \mu Q^+.$$

Proof. Replacing Q by $Q' = Q - \operatorname{Diag}(Q_{11},\dots,Q_{nn})$ we are reduced to the case where the diagonal entries of Q are zero. We apply Theorem 3.1. Fix X PSD with $X_{ii} = 1$, $i = 1,\dots,n$, such that $\langle Q, X\rangle = Q_+$. Choose $\epsilon_{ij} \in \{-1,1\}$ such that $Q_{ij} = \epsilon_{ij}|Q_{ij}|$. Then

$$\begin{aligned}
Q_* \leq \frac{2}{\pi}\langle Q, \arcsin[X]\rangle &= \frac{2}{\pi}\sum_{i\neq j} Q_{ij}\arcsin(X_{ij}) \\
&= \frac{2}{\pi}\sum_{i\neq j}|Q_{ij}|\arcsin(\epsilon_{ij}X_{ij}) \\
&= -\frac{2}{\pi}\sum_{i\neq j}|Q_{ij}|\left(\frac{\pi}{2} - \arcsin(\epsilon_{ij}X_{ij})\right) + \sum_{i\neq j}|Q_{ij}| \\
&\leq -\mu\sum_{i\neq j}|Q_{ij}|\left(1-\cos\left(\frac{\pi}{2} - \arcsin(\epsilon_{ij}X_{ij})\right)\right) + \sum_{i\neq j}|Q_{ij}| \\
&= -\mu\sum_{i\neq j}|Q_{ij}|(1-\epsilon_{ij}X_{ij}) + \sum_{i\neq j}|Q_{ij}| \\
&= -\mu\sum_{i\neq j}|Q_{ij}| + \mu\sum_{i\neq j}Q_{ij}X_{ij} + \sum_{i\neq j}|Q_{ij}| \\
&= (1-\mu)\Big(\sum_{i\neq j}|Q_{ij}|\Big) + \mu Q_+.
\end{aligned}$$

This proves the first assertion. The second assertion follows from the first by replacing Q by $-Q$. □

See [11, Theorem 3] for a proof of Theorem 4.1 in the special case where $Q_{ij} \geq 0$ (resp., $Q_{ij} \leq 0$) for $i \neq j$. In this case, $\operatorname{tr}(Q)+\sum_{i\neq j}|Q_{ij}|$ (resp., $\operatorname{tr}(Q)-\sum_{i\neq j}|Q_{ij}|$) coincides with Q^+ (resp., Q_+); see Corollary 2.4. One obtains the Goemans–Williamson result in [1] by applying Theorem 4.1 to the quadratic form $Q(x) := \sum_{(i,j)\in E}\frac{(x_i-x_j)^2}{4}$, where E is some set of ordered pairs (i,j), $i,j \in \{1,\dots,n\}$, with $i < j$ (the set of edges of a graph with vertices $1,\dots,n$). The examples considered by Karloff in [2] show that the constant μ in Theorem 4.1 is the best possible.

We now give another bound involving $\sum_{i\neq j}|Q_{ij}|$, in some sense complementary to Theorem 4.1, which takes the trivial bound into account. For Q nondiagonal, define

$$(4.1) \qquad s := \frac{\sum_{i\neq j}|Q_{ij}|}{\operatorname{tr}(Q)+\sum_{i\neq j}|Q_{ij}| - Q_+}, \quad s' := \frac{\sum_{i\neq j}|Q_{ij}|}{Q^+ - \operatorname{tr}(Q) + \sum_{i\neq j}|Q_{ij}|}.$$

Note: $\frac{s}{1-s} = \frac{\sum_{i\neq j}|Q_{ij}|}{\mathrm{tr}(Q)-Q_+}$. By Theorems 2.3 and 2.6, $1 \leq \frac{s}{1-s} \leq n-1$, so $s \in [\frac{1}{2}, \frac{n-1}{n}]$. Also, $\frac{s}{1-s} \geq \frac{Q^+-\mathrm{tr}(Q)}{\mathrm{tr}(Q)-Q_+} = \frac{r}{r'} = \frac{r}{1-r}$, so $s \geq r$ and $s = r$ iff $Q^+ - \mathrm{tr}(Q) = \sum_{i\neq j}|Q_{ij}|$. A similar argument shows that $s' \in [\frac{1}{2}, \frac{n-1}{n}]$, $s' \geq r'$, and $s' = r'$ iff $\mathrm{tr}(Q) - Q_+ = \sum_{i\neq j}|Q_{ij}|$.

THEOREM 4.2. *For Q nondiagonal,*

$$Q_* \leq (1-\beta(s))\,\mathrm{tr}(Q) + \beta(s)Q_+,$$
$$Q^* \geq (1-\beta(s'))\,\mathrm{tr}(Q) + \beta(s')Q^+.$$

Here s, s' are defined by (4.1), *and* $\beta(y) := \frac{2}{\pi}\max_{t\in(0,1]}\{\arcsin(t) - g(t)\frac{y}{1-y}\}$, *where* $g(t) := \sqrt{(\frac{\arcsin(t)}{t})^2 - 1} - \arctan(\sqrt{(\frac{\arcsin(t)}{t})^2-1})$.

Proof. Replacing Q by $Q' = Q - \mathrm{Diag}(Q_{11}, \ldots, Q_{nn})$ we are reduced to the case where the diagonal entries of Q are zero. Pick X PSD with $X_{ii} = 1$, $i = 1, \ldots, n$, and $\langle Q, X\rangle = Q_+$. Let $X_t := tX + (1-t)I$, $t \in [0,1]$. Then $\arcsin[X_t] = (\frac{\pi}{2} - \arcsin(t))I + \arcsin[tX]$ and, by Theorem 3.1,

$$\begin{aligned} Q_* &\leq \frac{2}{\pi}\langle Q, \arcsin[X_t]\rangle \\ &= \left(1 - \frac{2}{\pi}\arcsin(t)\right)\langle Q, I\rangle + \frac{2}{\pi}\langle Q, \arcsin[tX]\rangle \\ &= \frac{2}{\pi}\langle Q, \arcsin[tX]\rangle \\ &= \frac{2}{\pi}\langle Q, \arcsin[tX] - \arcsin(t)X\rangle + \frac{2}{\pi}\arcsin(t)\langle Q, X\rangle \\ &= \frac{2}{\pi}\sum_{i\neq j} Q_{ij}(\arcsin(tX_{ij}) - \arcsin(t)X_{ij}) + \frac{2}{\pi}\arcsin(t)Q_+. \end{aligned}$$

Write $Q_{ij} = \epsilon_{ij}|Q_{ij}|$, $\epsilon_{ij} \in \{-1, 1\}$, and consider the individual terms

$$Q_{ij}(\arcsin(tX_{ij}) - \arcsin(t)X_{ij}) = |Q_{ij}|(\arcsin(t\epsilon_{ij}X_{ij}) - \arcsin(t)\epsilon_{ij}X_{ij})$$

in the sum. Since $\arcsin(t)x \geq \arcsin(tx)$ for $x \in (0,1]$, the terms with $\epsilon_{ij}X_{ij} \geq 0$ contribute negatively. Terms with $\epsilon_{ij}X_{ij} < 0$ contribute at most $g(t)|Q_{ij}|$, where $g(t)$ denotes the maximum of the function

$$h_t(x) = \arcsin(t)x - \arcsin(tx)$$

on the interval $(0,1]$. One checks that the maximum is achieved at $x = \frac{\sqrt{(\frac{\arcsin(t)}{t})^2-1}}{\arcsin(t)}$, so

$$\begin{aligned} g(t) &= \arcsin(t)x - \arcsin(tx) \\ &= \sqrt{\left(\frac{\arcsin(t)}{t}\right)^2 - 1} - \arcsin\left(\frac{\sqrt{(\frac{\arcsin(t)}{t})^2 - 1}}{\frac{\arcsin(t)}{t}}\right) \\ &= \sqrt{\left(\frac{\arcsin(t)}{t}\right)^2 - 1} - \arctan\left(\sqrt{\left(\frac{\arcsin(t)}{t}\right)^2 - 1}\right). \end{aligned}$$

This proves

$$Q_* \leq \frac{2}{\pi}g(t)\sum_{i\neq j}|Q_{ij}| + \frac{2}{\pi}\arcsin(t)Q_+.$$

Table 1

y	$\omega(y)$	$\delta(y)$
0.50	0.718	0.895
0.55	0.736	0.884
0.60	0.755	0.875
0.65	0.777	0.871
0.70	0.800	0.873
0.75	0.826	0.879
0.80	0.854	0.891
0.85	0.885	0.908
0.90	0.919	0.931
0.95	0.957	0.961

Finally, using $\frac{s}{1-s} = \frac{\sum_{i\neq j}|Q_{ij}|}{\mathrm{tr}(Q)-Q_+} = \frac{\sum_{i\neq j}|Q_{ij}|}{-Q_+}$, this yields

$$\begin{aligned} Q_* &\leq \frac{2}{\pi}g(t)\sum_{i\neq j}|Q_{ij}| + \frac{2}{\pi}\arcsin(t)Q_+ \\ &= -\frac{2}{\pi}g(t)\frac{s}{1-s}Q_+ + \frac{2}{\pi}\arcsin(t)Q_+ = \frac{2}{\pi}\left(\arcsin(t) - g(t)\frac{s}{1-s}\right)Q_+. \end{aligned}$$

Since this holds for any $t \in (0,1]$, the first assertion is now clear. The second assertion follows from the first. □

Theorem 4.2 can also be formulated as follows.

Corollary 4.3. *For Q nondiagonal,*

$$\begin{aligned} Q_* &\leq (1-\delta(s))\Big(\mathrm{tr}(Q) + \sum_{i\neq j}|Q_{ij}|\Big) + \delta(s)Q_+, \\ Q^* &\geq (1-\delta(s'))\Big(\mathrm{tr}(Q) - \sum_{i\neq j}|Q_{ij}|\Big) + \delta(s')Q^+, \end{aligned}$$

where $\delta(y) := y + \beta(y)(1-y)$ and $\beta(y)$ is defined as in Theorem 4.2.

Proof. By Theorem 4.2,

$$\begin{aligned} \frac{\mathrm{tr}(Q) + \sum_{i\neq j}|Q_{ij}| - Q_*}{\mathrm{tr}(Q) + \sum_{i\neq j}|Q_{ij}| - Q_+} &\geq \frac{\mathrm{tr}(Q) + \sum_{i\neq j}|Q_{ij}| - (1-\beta(s))\,\mathrm{tr}(Q) - \beta(s)Q_+}{\mathrm{tr}(Q) + \sum_{i\neq j}|Q_{ij}| - Q_+} \\ &= \frac{\sum_{i\neq j}|Q_{ij}| + \beta(s)(\mathrm{tr}(Q) - Q_+)}{\mathrm{tr}(Q) + \sum_{i\neq j}|Q_{ij}| - Q_+} = s + \beta(s)(1-s). \end{aligned}$$

This proves the first assertion. The second assertion follows from the first. □

Theorem 4.2 improves on Theorem 3.3 when Q^+ (resp., Q_+) is "sufficiently close" to $\mathrm{tr}(Q) + \sum_{i\neq j}|Q_{ij}|$ (resp., $\mathrm{tr}(Q) - \sum_{i\neq j}|Q_{ij}|$); see Corollary 4.3 and Table 1. This occurs, for example, when $Q_{ij} \geq 0$ (resp., $Q_{ij} \leq 0$) for $i \neq j$; see Corollary 2.4.

It is possible to show that $\delta(y) \leq \mu \Leftrightarrow y \in [a,b]$, where $a \approx 0.5777$, $b \approx 0.7457$, so Theorem 4.2 improves on Theorem 4.1 when $s \in [0.5, 0.5777) \cup (0.7457, 1]$ (resp., when $s' \in [0.5, 0.5777) \cup (0.7457, 1]$). It is important to note that this does not contradict the result in [2]. The examples in [2] showing that the Goemans–Williamson approximation ratio is the best possible have s (resp., s') close to 0.5920, i.e., well within the interval $[0.5777, 0.7457]$.

Thus Theorem 4.2 allows one to predict better accuracy of the output of (the natural generalization of) the Goemans–Williamson MAX-CUT algorithm in certain cases, depending on s. Specifically, in terms of the original Goemans–Williamson MAX-CUT algorithm, since $\frac{1}{(2)(0.5777)} \approx 0.8655$ and $\frac{1}{(2)(0.7467)} \approx 0.6705$ one can predict better accuracy when the output value is either $\geq 86.6\%$ of the total number of edges or $\leq 67.0\%$ of the total number of edges. By way of comparison, $\frac{1}{(2)(0.5920)} \approx 0.8446$, so the Karloff examples have cut size $\approx 84.5\%$ of the total number of edges.

5. Improvements on the trivial bound. To simplify the presentation, we focus our attention now on Q_*. The reader will have no difficulty at this point in formulating the corresponding results for Q^*.

If n is large and r (resp., s) is relatively close to $\frac{n-1}{n}$, then $\operatorname{tr}(Q) - Q_+$ is relatively small compared to $Q^+ - \operatorname{tr}(Q)$ (resp., $\sum_{i\neq j}|Q_{ij}|$). So, in this sense, the trivial bound $Q_* \leq \operatorname{tr}(Q)$ for Q_* is already a good bound. Whether it is the best possible is another question. This latter question is the one we are concerned with in this section.

To allow comparison with the trivial bound, we note that Theorems 3.3 and 4.1 can also be formulated as follows.

COROLLARY 5.1. *For Q nondiagonal,*

(1) $Q_* \leq (1-\alpha(r))\operatorname{tr}(Q) + \alpha(r)Q_+$,

(2) $Q_* \leq (1-\gamma(s))\operatorname{tr}(Q) + \gamma(s)Q_+$,

where r is defined by (3.1), *s is defined by* (4.1), $\alpha(y) := \frac{2}{\pi}\frac{\sqrt{1-y^2} - y\arccos(y)}{1-y}$, *and* $\gamma(y) := \frac{\mu - y}{1-y}$.

Proof. (1) By Theorem 3.3,

$$\begin{aligned}
\frac{\operatorname{tr}(Q) - Q_*}{\operatorname{tr}(Q) - Q_+} &\geq \frac{\operatorname{tr}(Q) - (1-\omega(r))Q^+ - \omega(r)Q_+}{\operatorname{tr}(Q) - Q_+} \\
&= \omega(r)\frac{Q^+ - Q_+}{\operatorname{tr}(Q) - Q_+} - \frac{Q^+ - \operatorname{tr}(Q)}{\operatorname{tr}(Q) - Q_+} \\
&= \omega(r)\frac{1}{1-r} - \frac{r}{1-r} = \frac{\omega(r) - r}{1-r} \\
&= \frac{\frac{2}{\pi}(\sqrt{1-r^2} + r\arcsin(r)) - r}{1-r} \\
&= \frac{2}{\pi}\frac{\sqrt{1-r^2} - r(\frac{\pi}{2} - \arcsin(r))}{1-r} \\
&= \frac{2}{\pi}\frac{\sqrt{1-r^2} - r\arccos(r)}{1-r}.
\end{aligned}$$

(2) By Theorem 4.1,

$$\begin{aligned}
\frac{\operatorname{tr}(Q) - Q_*}{\operatorname{tr}(Q) - Q_+} &\geq \frac{\operatorname{tr}(Q) - (1-\mu)(\operatorname{tr}(Q) + \sum_{i\neq j}|Q_{ij}|) - \mu Q_+}{\operatorname{tr}(Q) - Q_+} \\
&= \mu\frac{\operatorname{tr}(Q) + \sum_{i\neq j}|Q_{ij}| - Q_+}{\operatorname{tr}(Q) - Q_+} - \frac{\sum_{i\neq j}|Q_{ij}|}{\operatorname{tr}(Q) - Q_+} \\
&= \mu\frac{1}{1-s} - \frac{s}{1-s} = \frac{\mu - s}{1-s}. \qquad \square
\end{aligned}$$

Clearly Theorem 4.1 improves on the trivial bound iff $s < \mu$. The function α is positive and decreasing on $[0,1)$, $\alpha(0) = \frac{2}{\pi}$, $\lim_{y\to 1^-}\alpha(y) = 0$. The function β defined

Table 2

y	$\alpha(y)$	$\beta(y)$	$\gamma(y)$
0.50	0.436	0.789	0.757
0.55	0.412	0.743	0.730
0.60	0.388	0.688	0.696
0.65	0.361	0.632	0.653
0.70	0.334	0.576	0.595
0.75	0.304	0.517	0.514
0.80	0.271	0.456	0.393
0.85	0.234	0.389	0.190
0.90	0.191	0.314	−0.214
0.95	0.135	0.219	−1.429

in the statement of Theorem 4.2 is positive and decreasing on $[\frac{1}{2},1)$, $\beta(\frac{1}{2}) \approx 0.7895$, $\lim_{y\to 1^-} \beta(y) = 0$. Thus Theorems 3.3 and 4.2 both improve on the trivial bound, but for r (resp., s) close to $\frac{n-1}{n}$ with n large, the improvement is only marginal. See Table 2. The next result describes more exactly the behavior of $\alpha(y)$ and $\beta(y)$ for y close to 1.

Theorem 5.2.
(1) $\lim_{y\to 1^-} \frac{\alpha(y)}{\sqrt{1-y}} = (\frac{2}{\pi})(\frac{2\sqrt{2}}{3}) \approx (\frac{2}{\pi})(0.9428)$.
(2) $\lim_{y\to 1^-} \frac{\beta(y)}{\sqrt{1-y}} = (\frac{2}{\pi})(\frac{2}{3^{1/4}}) \approx (\frac{2}{\pi})(1.5197)$.

Proof. The proof of (1) is straightforward, e.g., use l'Hôpital's rule. For (2), denote by $t = t_y$ the value of t that maximizes the function in the formula for $\beta(y)$. Clearly $t_y \to 0$ as $y \to 1^-$. Using power series approximations, one checks that, for y close to 1, $t_y \approx 3^{3/4}\sqrt{1-y}$. The rest of the computation is standard. □

For fixed n, there exists a positive real number $\epsilon \le 1$ (depending on n), such that $\frac{\operatorname{tr}(Q)-Q_*}{\operatorname{tr}(Q)-Q_+} \ge \epsilon$ holds for all nondiagonal symmetric $n \times n$ matrices Q, i.e., such that $Q_* \le (1-\epsilon)\operatorname{tr}(Q) + \epsilon Q_+$ holds for *all* symmetric $n \times n$ matrices Q. For example, we can take $\epsilon = \alpha(\frac{n-1}{n})$ or $\beta(\frac{n-1}{n})$. Alternatively, one can prove the result directly, using Lemma 2.5 and a compactness argument.

Denote by ρ_n the largest $\epsilon \le 1$ such that $Q_* \le (1-\epsilon)\operatorname{tr}(Q) + \epsilon Q_+$ holds for all symmetric $n \times n$ matrices Q. One checks easily that $\rho_1 = \rho_2 = 1$. Recent computations by Pereira [7] show that $\rho_3 = \rho_4 = \rho_5 = \rho_6 = \frac{2}{3}$. Of course, ρ_n is a nonincreasing function, so $\rho_n \le \frac{2}{3}$ for $n \ge 7$. Nothing else seems to be known about upper bounds for ρ_n. The quadratic form $Q(x) = (x_1+x_2+x_3)^2$ gives $\frac{\operatorname{tr}(Q)-Q_*}{\operatorname{tr}(Q)-Q_+} = \frac{2}{3}$; see Example 2.7. The author knows of no example worse than this.[1] The following question appears to be open.

Example 5.3. Question: Is it true that $\lim_{n\to\infty} \rho_n = 0$?

Our goal here is to find a better lower bound for ρ_n. According to Theorem 5.2 (2), for large n,

$$\beta\left(\frac{n-1}{n}\right) \approx \frac{2}{\pi}1.5197\sqrt{1-\frac{n-1}{n}} \approx \frac{0.9675}{\sqrt{n}};$$

i.e., we have a lower bound for ρ_n which approaches zero like $\frac{1}{\sqrt{n}}$. We proceed to improve on this, obtaining a lower bound for ρ_n which approaches zero like $\frac{1}{\sqrt[3]{n}}$. We begin by proving the analogue of Theorem 2.6 referred to earlier.

[1] If Conjecture 2.12 at the end of Karloff's paper [2] is true, then one can find examples with $\frac{\operatorname{tr}(Q)-Q_*}{\operatorname{tr}(Q)-Q_+}$ close to $\frac{2}{\pi}$ (i.e., just slightly worse than $\frac{2}{3}$) for large n.

THEOREM 5.4. *For Q nondiagonal,*

(1) *if n is even, then $\frac{1}{n-1} \leq \frac{Q^*-\mathrm{tr}(Q)}{\mathrm{tr}(Q)-Q_*} \leq n-1$;*

(2) *if n is odd, then $\frac{1}{n} \leq \frac{Q^*-\mathrm{tr}(Q)}{\mathrm{tr}(Q)-Q_*} \leq n$.*

Proof. We can assume the diagonal entries of Q are 0. For any subset I of $\{1,\dots,n\}$, denote by $Q_I(x)$ the quadratic form obtained from $Q(x)$ by replacing x_i by $-x_i$ for each $i \in I$. For each pair of indices $i < j$, if either both of i and j are in I or both of i and j are not in I, then the coefficient of x_ix_j in $Q_I(x)$ is $2Q_{ij}$. In the remaining cases, i.e., where one of i, j is in I and the other is not, the coefficient of x_ix_j in $Q_I(x)$ is $-2Q_{ij}$. For $1 \leq k < n$ denote by $Q_k(x)$ the sum of the $Q_I(x)$, I running through all k-element subsets of $\{1,\dots,n\}$. For fixed $i < j$, x_ix_j appears with coefficient $-2Q_{ij}$ in $2\binom{n-2}{k-1}$ of the $Q_I(x)$ and with coefficient $2Q_{ij}$ in the remainder of the $Q_I(x)$. It follows that

$$Q_k(x) = \binom{n}{k} Q(x) - 4\binom{n-2}{k-1} Q(x) = \frac{(n-2k)^2-n}{n(n-1)} \binom{n}{k} Q(x).$$

Let $Q_* = -\delta$. Then $Q_I(x) \geq -\delta$ for each $x \in \{-1,1\}^n$, so $Q_k(x) \geq -\binom{n}{k}\delta$, i.e., $-Q_k(x) \leq \binom{n}{k}\delta$. If $n > (n-2k)^2$, this yields $Q(x) \leq \frac{n(n-1)}{n-(n-2k)^2}\delta$. For n odd, say $n = 2\ell - 1$, apply this with $k = \ell$ to obtain $Q(x) \leq n\delta$. Similarly, for n even, say $n = 2\ell$, apply this with $k = \ell$ to obtain $Q(x) \leq (n-1)\delta$. A similar argument shows that if $Q^* = \delta$, then for any $x \in \{-1,1\}^n$, $Q(x) \geq -n\delta$ if n is odd and $Q(x) \geq -(n-1)\delta$ if n is even. □

COROLLARY 5.5.

(1) *If Q is nondiagonal, then*

$$Q_* \leq \left(1 - \frac{r\alpha(r')}{r'(2\ell-1)}\right)\mathrm{tr}(Q) + \frac{r\alpha(r')}{r'(2\ell-1)}Q_+.$$

(2) *If, in addition, $Q_{ij} \geq 0$ for $i \neq j$, then*

$$Q_* \leq \left(1 - \frac{r}{r'(2\ell-1)}\right)\mathrm{tr}(Q) + \frac{r}{r'(2\ell-1)}Q_+.$$

Here, r, r' are defined by (3.1) *and ℓ is defined by $n = 2\ell$ if n is even, $n = 2\ell - 1$ if n is odd.*

Proof. (1) After shifting and scaling, we can assume that $\mathrm{tr}(Q) = 0$ and $Q_+ = -1$, so $Q^+ = \frac{r}{r'}$. Applying Corollary 5.1(1) to $-Q$ yields $Q^* \geq (1-\alpha(r'))\,\mathrm{tr}(Q) + \alpha(r')Q^+ = \alpha(r')Q^+$. By Theorem 5.4, $2\ell - 1 \geq \frac{Q^*}{-Q_*}$. This implies $Q_* \leq -\frac{Q^*}{2\ell-1} \leq -\frac{\alpha(r')Q^+}{2\ell-1} = -\frac{r\alpha(r')}{r'(2\ell-1)}$. The proof of (2) is similar, except that now, by Corollary 2.4, we have $Q^* = Q^+$. □

Note: As y increases to $\frac{n-1}{n}$, $\frac{y\alpha(1-y)}{(1-y)(2\ell-1)}$ increases to $\alpha(\frac{1}{n})$ if n is even and to $\frac{n-1}{n}\alpha(\frac{1}{n})$ if n is odd, and, similarly, $\frac{y}{(1-y)(2\ell-1)}$ increases to 1 if n is even and to $\frac{n-1}{n}$ if n is odd. In particular, the bounds provided by Corollary 5.5 are appreciably better than the trivial bound if r is sufficiently close to $\frac{n-1}{n}$.

Now consider

$$\overline{\alpha}_n := \min_{y\in[\frac{1}{n},\frac{n-1}{n}]} \left\{ \max\left\{ \alpha(y), \frac{y\alpha(1-y)}{(1-y)(2\ell-1)} \right\} \right\},$$

Table 3

n	$\overline{\alpha}_n$	$\beta(\frac{n-1}{n})$	$\gamma(\frac{n-1}{n})$	$\overline{\beta}_n$
2	0.4360	0.7895	0.7572	0.7895
3	0.3526	0.6134	0.6358	0.6440
4	0.3497	0.5174	0.5144	0.6440
5	0.3093	0.4560	0.3930	0.5467
0	0.2653	0.3137		0.4524
15	0.2300	0.2539		0.3857
20	0.2148	0.2190		0.3579
25	0.1980	0.1954		0.3278
50	0.1615	0.1389		0.2639
100	0.1293	0.0975		0.2097
200	0.1035	0.0685		0.1667

where ℓ is defined as in Corollary 5.5. This is a lower bound for ρ_n. See Table 3 for a comparison of the lower bounds $\overline{\alpha}_n$, $\beta(\frac{n-1}{n})$, and $\gamma(\frac{n-1}{n})$. The bound given by $\beta(\frac{n-1}{n})$ is best for n in the range $2 \le n \le 23$, $n \neq 3$. For $n = 3$ the bound given by $\gamma(\frac{n-1}{n})$ is best. For $n \ge 24$ the bound given by $\overline{\alpha}_n$ is best. It is pretty clear that these lower bounds are nowhere near optimal.

The fifth column in Table 3 gives lower bounds for the function $\rho'_n :=$ the maximum $\epsilon \le 1$ such that $Q_* \le (1-\epsilon)\operatorname{tr}(Q) + \epsilon Q_+$ holds for all symmetric $n \times n$ matrices Q such that $Q_{ij} \ge 0$ for $i \neq j$. The function $\overline{\beta}_n$ is defined by

$$\overline{\beta}_n := \min_{y \in [\frac{1}{2}, \frac{n-1}{n}]} \left\{ \max \left\{ \beta(y), \frac{y}{(1-y)(2\ell-1)}, \gamma(y) \right\} \right\}.$$

Again, it is pretty clear that this lower bound for ρ'_n is nowhere near optimal.

Theorem 5.6. *For large n,*

(1) $\overline{\alpha}_n \approx \frac{0.6121}{\sqrt[3]{n}}$;

(2) $\overline{\beta}_n \approx \frac{0.9782}{\sqrt[3]{n}}$.

Proof. (1) On the interval $[\frac{1}{n}, \frac{n-1}{n}]$, $\alpha(y)$ is decreasing and $\frac{y\alpha(1-y)}{(1-y)(2\ell-1)}$ is increasing. Thus the minimum occurs when $\alpha(y) = \frac{y\alpha(1-y)}{(1-y)(2\ell-1)}$. It follows that $y \to 1$ as $n \to \infty$ and

$$\begin{aligned} n\alpha(y)^3 &= n\frac{y\alpha(1-y)}{(1-y)(2\ell-1)}\alpha(y)^2 \\ &= \frac{n}{2\ell-1} y\alpha(1-y)\left(\frac{\alpha(y)}{\sqrt{1-y}}\right)^2 \to (1)(1)\left(\frac{2}{\pi}\right)\left(\frac{2}{\pi}\right)^2\left(\frac{2\sqrt{2}}{3}\right)^2 \text{ as } n \to \infty. \end{aligned}$$

Here, we use Theorem 5.2(1). This implies $\lim_{n\to\infty} \sqrt[3]{n}\overline{\alpha}_n = \frac{2}{\pi}(\frac{2\sqrt{2}}{3})^{2/3} \approx 0.6121$.

(2) The proof is similar. For large n, the minimum on the interval $[\frac{1}{2}, \frac{n-1}{n}]$ occurs when $\beta(y) = \frac{y}{(1-y)(2\ell-1)}$ and $\gamma(y)$ is negative. Also $y \to 1$ as $n \to \infty$ and

$$\begin{aligned} n\beta(y)^3 &= n\frac{y}{(1-y)(2\ell-1)}\beta(y)^2 \\ &= \frac{n}{2\ell-1} y\left(\frac{\beta(y)}{\sqrt{1-y}}\right)^2 \to (1)(1)\left(\frac{4}{3^{1/4}\pi}\right)^2 \text{ as } n \to \infty. \end{aligned}$$

Here, we use Theorem 5.2(2). This implies $\lim_{n\to\infty} \sqrt[3]{n}\overline{\beta}_n = (\frac{4}{3^{1/4}\pi})^{2/3} \approx 0.9782$. □

6. Conclusion. This paper gives error estimates when Q_* is approximated by Q_+. The bounds provided by Theorems 3.2 and 3.3 come from [5]. Theorem 4.1 is a generalization of the result in [1]. A special case of Theorem 4.1 already appears in [11]. The bound given in Theorem 4.2 is new, improves on Theorem 3.3 when s is close to r, and provides improved estimates in the MAX-CUT algorithm in [1] for $s \notin [0.5777, 0.7457]$. The bounds given by Corollary 5.5 are also new, but are only useful when r is sufficiently close to $\frac{n-1}{n}$. The situation regarding ρ_n is unsatisfactory. One would like to know the limiting value of ρ_n as $n \to \infty$. Is it zero, or is it positive? The best we are able to show is that, for large n, ρ_n is bounded away from zero by a function of the form $\frac{C}{\sqrt[3]{n}}$, where C is a constant.

Acknowledgment. The author wishes to thank Markus Schweighofer for sharing his expertise.

REFERENCES

[1] M. Goemans and D. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. Assoc. Comput. Mach., 6 (1995), pp. 1115–1145.

[2] H. Karloff, *How good is the Goemans–Williamson MAX CUT algorithm?*, SIAM J. Comput., 29 (1999), pp. 336–350.

[3] J. B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM J. Optim., 11 (2001), pp. 796–817.

[4] M. Marshall, *Optimization of polynomial functions*, Canad. Math. Bull., 46 (2003), pp. 575–587.

[5] Y. Nesterov, *Semidefinite relaxation and nonconvex quadratic optimization*, Optim. Methods Softw., 9 (1998), pp. 141–160.

[6] Y. Nesterov, H. Wolkowicz, and Y. Ye, *Semidefinite programming relaxations of nonconvex quadratic optimization*, in Handbook of Semidefinite Programming, H. Wolkowicz, R. Saigal, and L. Vandenberghe, eds., Kluwer Academic, Boston, MA, 2000, pp. 361–419.

[7] R. Pereira, *The Convex Hull of Rank-One Correlation Matrices*, preprint.

[8] D. Xu and S. Zhang, *Approximation Bounds for Quadratic Maximization with Semidefinite Programming Relaxation*, Report, Department of Systems Engineering and Engineering Management, Chinese University of Hong Kong, 2002.

[9] Y. Ye, *Approximating quadratic programming with bound and quadratic constraints*, Math. Program., 84 (1999), pp. 219–226.

[10] Y. Ye, *Approximating global quadratic optimization with convex quadratic constraints*, J. Global Optim., 15 (1999), pp. 1–17.

[11] S. Zhang, *Quadratic maximization and semidefinite relaxation*, Math. Program., 87 (2000), pp. 453–465.

SIAM J. OPTIM.
Vol. 16, No. 2, pp. 311–340

STRONG CHIP FOR INFINITE SYSTEM OF CLOSED CONVEX SETS IN NORMED LINEAR SPACES[*]

CHONG LI[†] AND K. F. NG[‡]

Abstract. For a general (possibly infinite) system of closed convex sets in a normed linear space we provide several sufficient conditions for ensuring the strong conical hull intersection property. One set of sufficient conditions is given in terms of the finite subsystems while the other sets are in terms of the relaxed interior-point conditions together with appropriate continuity of the associated set-valued function on the (topologized) index set I. In the special case when I is finite and X is finite dimensional, one of these results reduces to a classical result of Rockafellar.

Key words. system of closed convex sets, interior-point condition, strong conical hull intersection property

AMS subject classifications. Primary, 90C34, 90C25; Secondary, 52A05, 41A29

DOI. 10.1137/040613238

1. Introduction.

The notion of the strong CHIP (conical hull intersection property) was introduced by Deutsch, Li, and Ward in [12, 13] for a finite family of closed convex sets in a Euclidean space (or a Hilbert space) and has been successfully applied in the reformulation of some best approximation problems. This notion closely relates other fundamental concepts such as bounded linear regularity, G-property of Jameson, error bounds in convex optimization [1, 3], and the BCQ (basic constraint qualification) as well as the perturbations for finite convex systems of inequalities. See [5, 6, 8, 12, 13, 14, 18, 19, 24, 25] and references therein, especially in [20], where the strong CHIP was defined for an arbitrary family of closed convex sets in a Banach space and utilized in the study of general systems of infinite convex inequalities, such as the system that naturally arises from the problem of best restricted range approximation in the space $C(Q)$ of complex-valued continuous functions on a compact metric space Q under quite general constraints. This problem was first presented and formulated by Smirnov and Smirnov in [31, 32], where each Ω_t was assumed to be a disk in $\mathbb{C}$. Later in [33, 34, 35] and also more recently in [17, 20], the constraint sets Ω_t have been relaxed but still remain to assume the strong interior-point condition (in particular, $\operatorname{int}\Omega_t \neq \emptyset$ for each $t \in Q$). This unfortunately excludes the interesting case when some Ω_t is a line segment or a singleton in $\mathbb{C}$. As demonstrated in an accompanying paper [22], the results obtained in the present paper have enabled us to study the restricted range approximation problem under much less restrictive assumptions by allowing the case that $\operatorname{int}\Omega_t = \emptyset$ for some $t \in Q$. The present paper is devoted to providing sufficient conditions for a (finite or infinite) family $\{C, C_i : i \in I\}$ of closed convex sets in a Banach (or normed linear) space to have the strong CHIP.

In expanding and improving the known results on the sufficient conditions for the

[*] Received by the editors August 11, 2004; accepted for publication (in revised form) February 15, 2005; published electronically October 7, 2005.

http://www.siam.org/journals/siopt/16-2/61323.html

[†] Department of Mathematics, Zhejiang University, Hangzhou 310027, People's Republic of China (cli@zju.edu.cn). This author was supported in part by the National Natural Science Foundation of China (grant 10271025) and the Program for New Century Excellent Talents in University.

[‡] Department of Mathematics, Chinese University of Hong Kong, Hong Kong, People's Republic of China (kfng@math.cuhk.edu.hk). This author was supported by a direct grant (CUHK) and an Earmarked Grant from the Research Grant Council of Hong Kong.

strong CHIP for $\{C, C_i : i \in I\}$ from the case when the index set I is finite to the case when I may be infinite, this paper presents two types of results. One type is on the natural approach to answer the question of whether or not the following implication is valid:

$$\begin{aligned}&\{C, C_j : j \in J\} \text{ has the strong CHIP for each finite subset } J \text{ of } I\\ &\Longrightarrow \{C, C_i : i \in I\} \text{ has the strong CHIP.}\end{aligned}$$

While the answer to this question is negative in general (see [13, Example 1]), we provide some reasonable conditions in section 5 to ensure the validity of the above implication. Another type of sufficient conditions presented in this paper is given more directly (in terms of the system itself rather than via its finite subsystems). In this connection, the starting point of our study is the following theorem. "DLW" refers to the authors Deutsch, Li, and Ward of [12, 13], where the assertions regarding the sufficiency for (a) and for (b) were stated and proved in the Hilbert space setting, but the arguments can be modified to suit the Banach space setting. For the sake of completeness and also for more convenient applications, we will present a direct proof for a slightly more general form in the next section (see also [26] for another approach).

Theorem DLW. *Let I be a finite index set and $\{C, C_i : i \in I\}$ be a finite family of nonempty closed convex sets in a Banach space X. Let $x_0 \in C \cap (\cap_{i\in I} C_i)$. Then the family $\{C, C_i : i \in I\}$ has the strong CHIP at x_0 provided that at least one of the following conditions is satisfied:*

(a) $C \bigcap (\operatorname{int} \bigcap_{i\in I} C_i) \neq \emptyset$.

(b) $\operatorname{ri} C \bigcap (\bigcap_{i\in I} C_i) \neq \emptyset$ *and each C_i is a polyhedron (where "*ri*" means "relative interior").*

(c) *There exists a subset I_0 of I such that C_i is a polyhedron for each $i \in I \setminus I_0$ and*

$$\operatorname{ri} C \bigcap \left(\operatorname{int} \bigcap_{i\in I_0} C_i\right) \bigcap \left(\bigcap_{i\in I\setminus I_0} C_i\right) \neq \emptyset. \tag{1.1}$$

The sufficiency of (c) follows directly from (a) and (b). The condition (a) is sometimes referred to as the strong interior-point condition (or Slater condition; see, e.g., [13]) which is equivalent (as I is finite) to the following interior-point condition:

(a′) $C \bigcap (\bigcap_{i\in I} \operatorname{int} C_i) \neq \emptyset$.

As shown in [20], when I is infinite, the above (a), (b), and (c) are no longer sufficient for the strong CHIP. A natural condition that one would like to impose is the continuity assumption for the set-valued mapping $i \mapsto C_i$; thus it is judicious for us to assume henceforth that

$$\text{the set } I \text{ is a compact metric space.} \tag{1.2}$$

(When I is finite, it will be regarded as a compact metric space under the discrete metric; needless to say, in this case the continuity assumption is automatically satisfied.)

Under an appropriate continuity assumption we show in Theorem 4.1 that (a) implies the strong CHIP at $x_0 \in C \cap (\cap_{i\in I} C_i)$ provided that C is finite dimensional or the set $I_C^{\mathrm{rb}}(x_0)$ of "C-relative boundary indices" for x_0 is finite. We remark that even in the case when C is finite dimensional, our results are genuinely an extension of Theorem DLW as some (or all) sets C_i can be infinite dimensional. In a similar fashion other parts of Theorem DLW are extended in section 4. In fact, we use the following condition, somewhat weaker than (c), to establish a sufficient condition result in Theorem 4.3. The family $\{C, C_i : i \in I\}$ is said to satisfy the weak-strong interior-point condition with the pair (I_1, I_2) if there exist disjoint finite subsets I_1, I_2 of I satisfying the following two properties:

$$\operatorname{ri} C \bigcap \left(\operatorname{int} \bigcap_{i \in I \setminus (I_1 \cup I_2)} C_i \right) \bigcap \left(\bigcap_{i \in I_1} \operatorname{ri} C_i \right) \bigcap_{i \in I_2} C_i \neq \emptyset; \tag{1.3}$$

$$C_i \text{ is a polyhedron for each } i \in I_2. \tag{1.4}$$

This condition, in contrast to the interior-point condition, enables us to consider the case when some C_i neither is a polyhedron nor has an interior point. Specializing to the case when $I = I_1 \cup I_2$ (thus $\operatorname{int}(\cap_{i \in I \setminus (I_1 \cup I_2)} C_i)$, to be read as X by convention), a corollary of Theorem 4.3 is the following infinite dimension extension of a result of Rockafellar [27, Corollary 23.8.1, p. 223]:

Let $I = J \cup K$ be finite such that C_k is a polyhedron for each $k \in K$ and suppose that

$$\operatorname{ri} C \bigcap \left(\bigcap_{j \in J} \operatorname{ri} C_j \right) \bigcap \left(\bigcap_{k \in K} C_k \right) \neq \emptyset. \tag{1.5}$$

Then the system $\{C, C_i : i \in I\}$ has the strong CHIP if at least one of the following conditions is satisfied.

(a) At least one of $\{C, C_j : j \in J\}$ is finite dimensional.

(b) C_j is finite codimensional for each $j \in J$.

2. Notations and preliminary results.

The notations used in the present paper are standard (cf. [7, 16]). In particular, we assume that X is a normed linear space throughout the whole paper, unless we explicitly state otherwise. We use $\mathbf{B}(x, \epsilon)$ to denote the closed ball with center x and radius ϵ. For a set Z in X (or in $\mathbb{R}^n$), the interior (resp., relative interior, closure, convex hull, convex cone hull, linear hull, affine hull, boundary, relative boundary) of Z is denoted by $\operatorname{int} Z$ (resp., $\operatorname{ri} Z$, $\overline{Z}$, $\operatorname{conv} Z$, $\operatorname{cone} Z$, $\operatorname{span} Z$, $\operatorname{aff} Z$, $\operatorname{bd} Z$, $\operatorname{rb} Z$), and the negative polar cone $Z^{\ominus}$ is the set defined by

$$Z^{\ominus} = \{x^* \in X^* \ : \ \langle x^*, z \rangle \leq 0 \quad \text{for all } z \in Z\}.$$

The normal cone of Z at z_0 is denoted by $N_Z(z_0)$ and defined by $N_Z(z_0) = (Z - z_0)^{\ominus}$. For convenience of printing we sometimes use $N(z_0; Z)$ in place of $N_Z(z_0)$. Let A be a closed convex nonempty subset of X. The interior and boundary of Z relative to A are denoted by $\operatorname{rint}_A Z$ and $\operatorname{bd}_A Z$, respectively; they are defined to be, respectively, the interior and boundary of the set $\operatorname{aff} A \cap Z$ in the metric space $\operatorname{aff} A$. Thus, a point $z \in \operatorname{rint}_A Z$ if and only if there exists $\varepsilon > 0$ such that

$$z \in (\operatorname{aff} A) \cap \mathbf{B}(z, \varepsilon) \subseteq Z, \tag{2.1}$$

while $z \in \operatorname{bd}_A Z$ if and only if $z \in \operatorname{aff} A$ and, for any $\varepsilon > 0$, $(\operatorname{aff} A) \cap \mathbf{B}(z, \varepsilon)$ intersects Z and its complement. Let $\mathbb{R}_-$ denote the subset of $\mathbb{R}$ consisting of all nonpositive real numbers. For a proper extended real-valued convex function on X, the subdifferential of f at $x \in X$ is denoted by $\partial f(x)$ and defined by

$$\partial f(x) = \{z^* \in X^* : \ f(x) + \operatorname{Re}\langle z^*, y - x\rangle \leq f(y) \quad \text{for all } y \in X\},$$

where $\langle z^*, x\rangle$ denotes the value of a functional z^* in X^* at $x \in X$, i.e., $\langle z^*, x\rangle = z^*(x)$.

For simplicity of notations, we will usually assume (with the exception of Proposition 2.1 and section 5) that the scalar field of X is $\mathbb{R}$ and that $\operatorname{Re}\langle x^*, x\rangle$ is to be replaced by $\langle x^*, x\rangle$.

Remark 2.1. (a) Let f be a continuous convex function on X and $x \in X$ with $f(x) = 0$. It is easy to see that $\operatorname{cone}(\partial f(x)) \subseteq N_{f^{-1}(\mathbb{R}_-)}(x)$, and that the equality holds if f is an affine function or if x is not a minimizer of f; see [7, Corollary 1, p. 56].

(b) The directional derivative of the function f at x in the direction d is denoted by $f'_+(x, d)$:

$$f'_+(x, d) := \lim_{t \to 0+} \frac{f(x + td) - f(x)}{t}. \tag{2.2}$$

We recall [7, Proposition 2.2.7] that, if x is a continuity point of f,

$$\partial f(x) = \{z^* \in X^* : \ \langle z^*, d\rangle \leq f'_+(x, d) \quad \text{for all } d \in X\} \tag{2.3}$$

and

$$f'_+(x, d) = \max\{\langle z^*, d\rangle : \ z^* \in \partial f(x)\}. \tag{2.4}$$

Let $\{A_i : \ i \in J\}$ be a family of subsets of X. The set $\sum_{i \in J} A_i$ is defined by

$$\sum_{i \in J} A_i = \begin{cases} \left\{\sum_{i \in J_0} a_i : \quad a_i \in A_i, \quad J_0 \subseteq J \text{ being finite}\right\} & \text{if } J \neq \emptyset, \\ \{0\} & \text{if } J = \emptyset. \end{cases} \tag{2.5}$$

The following concept of the strong CHIP plays an important role in optimization theory (see [1, 3, 8, 10, 11, 30]) and is due to [12, 13] in the case when I is finite and to [20] in the case when I is infinite.

Definition 2.1. *Let $\{C_i : \ i \in I\}$ be a collection of convex subsets of X and $x \in \bigcap_{i \in I} C_i$. The collection is said to have*

(a) *the strong CHIP at x if $N_{\bigcap_{i \in I} C_i}(x) = \sum_{i \in I} N_{C_i}(x)$, that is,*

$$\left(\bigcap_{i \in I} C_i - x\right)^{\ominus} = \sum_{i \in I} (C_i - x)^{\ominus}; \tag{2.6}$$

(b) *the strong CHIP if it has the strong CHIP at each point of $\cap_{i \in I} C_i$.*

Consider a convex inequality system (CIS) defined by

$$g_i(x) \leq 0, \quad i \in I, \tag{2.7}$$

where $x \in X$ and each g_i is a real continuous convex function on X. We always assume that the solution set S of the system (CIS) is nonempty, i.e.,

$$S := \{x \in X : \ g_i(x) \leq 0 \quad \text{for all } i \in I\} \neq \emptyset. \tag{2.8}$$

Let $G(\cdot)$ denote the sup-function [16] of $\{g_i\}$:

$$G(x) := \sup_{i\in I} g_i(x) \quad \text{for all } x \in X.$$

Then S is also the solution set of the convex inequality

$$G(x) \leq 0. \tag{2.9}$$

In this paper we assume throughout that

$$G(x) < +\infty \quad \text{for all } x \in X \tag{2.10}$$

and that G is continuous on X. These blanket assumptions are automatically satisfied if $\{g_i : \ i \in I\}$ is locally uniformly bounded. Moreover, the continuity of G automatically follows from (2.10) if X is finite dimensional.

Let $I(x)$ denote the set of all active indices i: $I(x) = \{i \in I : \ g_i(x) = G(x)\}$. Following [15, 23], we define

$$D'(x) := \operatorname{conv} \bigcup_{i\in I(x)} \partial g_i(x), \quad x \in X. \tag{2.11}$$

Note that, by (2.5), $D'(x) = \{0\}$ if $I(x) = \emptyset$.

The following theorem will play a key role in section 4. It is a known result; see, for example, [16, 23] for the special case when X is finite dimensional and [21] for the general case (the proof presented in [21] is valid for normed linear spaces though the result was stated in the Banach space setting).

Theorem 2.1. *Suppose that I is a compact metric space and that the function $i \mapsto g_i(x)$ is upper semicontinuous for each $x \in X$. Let $x_0 \in C$. Then $I(x_0) \neq \emptyset$ and the following assertions hold.*

(i) *If* $\operatorname{span} C$ *is finite dimensional, then*

$$N_C(x_0) + \partial G(x_0) = N_C(x_0) + D'(x_0). \tag{2.12}$$

(ii) $\partial G(x_0) = D'(x_0)$ *provided that $I(x_0)$ is finite.*

Theorem 2.2 below is a slight extension (applicable to convex, but not necessarily closed, sets in a normed space). To prepare for the proof we begin with a simple lemma.

Lemma 2.1. *Assume that C is a polyhedron in X defined by*

$$C = \bigcap_{i=1}^{k} \{x \in X : \ \langle h_i, x\rangle \leq d_i\}, \tag{2.13}$$

where $h_i \in X^* \setminus \{0\}$ and d_i is a real number for each $i = 1, \ldots, k$. Let $x_0 \in \operatorname{bd} C$ and let $I(x_0) = \{i \in \{1, \ldots, k\} : \ \langle h_i, x_0\rangle = d_i\}$. Then

$$N_C(x_0) = \operatorname{cone}\{h_i : \ i \in I(x_0)\}. \tag{2.14}$$

Consequently, $\{C_i : \ i = 1, 2, \ldots, n\}$ has the strong CHIP if each C_i is a polyhedron of X.

Proof. We need only prove that the set on the left-hand side of (2.14) is contained in that on the right-hand side. To do this, suppose on the contrary that

$y^* \in N_C(x_0) \setminus \operatorname{cone}\{h_i : i \in I(x_0)\}$. Since $\operatorname{cone}\{h_i : i \in I(x_0)\}$ is closed as I is finite, by the separation theorem, there exists an element $x^{**} \in X^{**}$ such that

$$\langle x^{**}, y^* \rangle > 0 \geq \sup\{\langle x^{**}, h\rangle : h \in \operatorname{cone}\{h_i : i \in I(x_0)\}\}. \tag{2.15}$$

Moreover, as $I(x_0)$ is a finite set, there exists $x \in X$ such that

$$\langle y^*, x\rangle = \langle x^{**}, y^*\rangle \quad \text{and} \quad \langle h_i, x\rangle = \langle x^{**}, h_i\rangle \quad \text{for each } i \in I(x_0). \tag{2.16}$$

Hence, by (2.15) and (2.16), we have that $tx + x_0 \in C$ for some $t > 0$ small enough. But $\langle y^*, (tx + x_0) - x_0\rangle = t\langle y^*, x\rangle > 0$, which contradicts that $y^* \in N_C(x_0)$, and the lemma is proved. □

Let Y be a subspace of X. We use N^Y to represent the normal cone operator taken in Y; namely, for any subset A of Y, $N_A^Y(x)$ is the set

$$N_A^Y(x) = \{y^* \in Y^* : \langle y^*, z - x\rangle \leq 0 \quad \text{for all } z \in A\}. \tag{2.17}$$

COROLLARY 2.1. *Let $Z \subseteq X$ be a closed subspace and $C \subseteq X$ a polyhedron. Let $x_0 \in C \cap Z$ and let $N_{C\cap Z}^Z(x_0)$ denote the normal cone of $C \cap Z$ at x_0 taken in Z. Then, for each $x_0^* \in N_{C\cap Z}^Z(x_0)$, there exists $x^* \in N_C(x_0)$ such that x^* is an extension of x_0^* (an extension of x_0^* obtained from Lemma* 2.1 *will be referred to as a natural extension of x_0^*).*

THEOREM 2.2. *Let I be a finite index set and $\{C, C_i : i \in I\}$ be a finite family of nonempty convex sets in a normed linear space X. Let $x_0 \in C \cap (\cap_{i\in I} C_i)$. Then the family $\{C, C_i : i \in I\}$ has the strong CHIP at x_0 provided that at least one of conditions* (a), (b), *and* (c) *in Theorem DLW is satisfied.*

Proof. In the case (a), if X is a Banach space and $\{C, C_i : i \in I\}$ is a family of nonempty closed convex sets, the proof is the same as that given in [13], except that here we apply [2, Theorem 2.6, p. 189] instead of [2, Corollary 2.5, p. 113]. Note further that the result is valid for any normed linear space X and any family $\{C, C_i : i \in I\}$ of nonempty convex sets. However, this observation does not constitute a genuine extension. Indeed, let U denote the open unit ball of $\overline{X}$. Let $\overline{C}$ and $\overline{C}_i$, respectively, denote the closures of C and C_i in $\overline{X}$. By (a), there exist $c \in C$ and $\epsilon > 0$ such that

$$(c + \epsilon U) \cap X \subseteq C_i \quad \text{for each } i \in I. \tag{2.18}$$

We claim that

$$c + \epsilon U \subseteq \overline{C}_i \quad \text{for each } i \in I. \tag{2.19}$$

Indeed, let $u \in U$. Then there exists a sequence $\{x_n\} \subseteq U \cap X$ convergent to u. Then by (2.18) $\{c + \epsilon x_n\} \subseteq C_i$, which implies that $c + \epsilon u \in \overline{C}_i$. Thus (2.19) is clear. Let $x_0 \in C \bigcap (\bigcap_{i\in I} C_i)$. Then one can apply the Banach space version of (a) in Theorem 2.2 to conclude that

$$N_{\overline{C}\cap(\cap_{i\in I}\overline{C}_i)}(x_0) \subseteq N_{\overline{C}}(x_0) + \sum_{i\in I} N_{\overline{C}_i}(x_0)$$

and hence

$$N_{C\cap(\cap_{i\in I} C_i)}(x_0) = N_{\overline{C\cap(\cap_{i\in I} C_i)}}(x_0) \subseteq N_C(x_0) + \sum_{i\in I} N_{C_i}(x_0)$$

because $\overline{C \cap (\cap_{i\in I} C_i)} = \overline{C} \cap (\cap_{i\in I} \overline{C}_i)$ thanks to (2.19). This completes the proof of (a).

Now let us verify the conclusion in the case (b). Let $x_0 \in C \bigcap (\bigcap_{i\in I} C_i)$ and let Z denote the subspace spanned by $C - x_0$. Since $\operatorname{ri} C \bigcap (\bigcap_{i\in I} C_i) \neq \emptyset$, the intersection of the interior of $C - x_0$ in the subspace Z and the set $\bigcap_{i\in I}(C_i - x_0)$ is nonempty. By the case (a) and Lemma 2.1 (applied to Z in place of X), we obtain that

$$\begin{aligned} N^Z_{(C-x_0)\cap(\cap_{i\in I}(C_i-x_0))}(0) &= N^Z_{C-x_0}(0) + N^Z_{Z\cap(\cap_{i\in I}(C_i-x_0))}(0) \\ &= N^Z_{C-x_0}(0) + \sum_{i\in I} N^Z_{Z\cap(C_i-x_0)}(0). \end{aligned} \tag{2.20}$$

Let $x^* \in N_{C\cap(\cap C_i)}(x_0)$. Then $x^*|_Z \in N^Z_{(C-x_0)\cap(\cap_{i\in I}(C_i-x_0))}(0)$; consequently, by (2.20), there exist $\widetilde{x}^*_0 \in N^Z_{C-x_0}(0)$ and $\widetilde{x}^*_i \in N^Z_{Z\cap(C_i-x_0)}(0)$ for each $i \in I$ such that

$$x^*|_Z = \widetilde{x}^*_0 + \sum_{i\in I} \widetilde{x}^*_i \quad \text{on } Z. \tag{2.21}$$

Let $x^*_0 \in X^*$ be an extension of $\widetilde{x}^*_0$. Then, as $C - x_0 \subseteq Z$, $x^*_0 \in N_{C-x_0}(0) = N_C(x_0)$. Also, for each $i \in I$, as $C_i - x_0$ is a polyhedron in X, there exists a natural extension $x^*_i \in N_{C_i-x_0}(0)$ of $\widetilde{x}^*_i$ by Corollary 2.1, and hence $x^*_i \in N_{C_i}(x_0)$ with $x^*_i|_Z = \widetilde{x}^*_i$ for each $i \in I$. Do this for each $i \in I$ and let $y^* = x^* - x^*_0 - \sum_{i\in I} x^*_i$. Then $y^* \in N_C(x_0)$ by (2.21). Hence, $x^* = y^* + x^*_0 + \sum_{i\in I} x^*_i \in N_C(x_0) + \sum_{i\in I} N_{C_i}(x_0)$ and the conclusion in the case (b) is proved. Therefore the proof of Theorem 2.2 is complete, as (c) follows from (a) and (b). □

For a closed convex subset W of X, let P_W denote the projection operator defined by

$$P_W(x) = \{y \in W : \|x - y\| = d_W(x)\},$$

where $d_W(x)$ denotes the distance from x to W. Recall that the duality map J from X to 2^{X^*} is defined by

$$J(x) := \{x^* \in X^* : \langle x^*, x\rangle = \|x\|^2, \ \|x^*\| = \|x\|\}. \tag{2.22}$$

In fact, $J(x) = \partial\phi(x)$, where $\phi(x) := \frac{1}{2}\|x\|^2$. Thus a Banach space X is smooth if and only if for each $x \in X$ the duality map is single-valued. We also need the following proposition, which was established independently by Deutsch [9] and Rubenstein [28] (see also [4]).

Proposition 2.1. *Let W be a convex set in X. Then for any $x \in X$, $z_0 \in P_W(x)$ if and only if $z_0 \in W$ and there exists $x^* \in J(x - z_0)$ such that* $\operatorname{Re}\langle x^*, z - z_0\rangle \leq 0$ *for any $z \in W$, that is, $J(x - z_0) \cap N_W(z_0) \neq \emptyset$. In particular, when X is smooth, $z_0 \in P_W(x)$ if and only if $z_0 \in W$ and $J(x - z_0) \in N_W(z_0)$.*

3. Extended Minkowski functional, interior-point condition, and continuity condition.

Recall that I denotes an index set which is assumed to be a compact metric space. For convenience, a family $\{C, C_i : i \in I\}$ is called a closed convex set system with base-set C (CCS-system with base-set C) if C and C_i are nonempty closed convex subsets of X for each $i \in I$.

Definition 3.1. *A CCS-system $\{C, C_i : i \in I\}$ with base-set C is said to satisfy*

(i) *the C-interior-point condition if*

$$C \bigcap \left(\bigcap_{i\in I} \operatorname{rint}_C C_i \right) \neq \emptyset; \tag{3.1}$$

(ii) *the strong C-interior-point condition if*

$$C\bigcap\left(\operatorname{rint}_C\bigcap_{i\in I}C_i\right)\neq\emptyset; \tag{3.2}$$

(iii) *the weak-strong C-interior-point condition with the pair (I_1,I_2) if there exist two disjoint finite subsets I_1 and I_2 of I such that each C_i ($i\in I_2$) is a polyhedron and*

$$\operatorname{ri}C\bigcap\left(\operatorname{rint}_C\bigcap_{i\in I\setminus(I_1\cup I_2)}C_i\right)\bigcap\left(\bigcap_{i\in I_1}\operatorname{ri}C_i\right)\bigcap_{i\in I_2}C_i\neq\emptyset; \tag{3.3}$$

(iv) *the interior-point condition (resp., the strong interior-point condition, the weak-strong interior-point condition with the pair (I_1,I_2)) if the operation "rint_C" in* (3.1) *(resp.,* (3.2), (3.3)*) is replaced with "*int*".*

Any point $\bar{x}$ belonging to the set on the left-hand side of (3.1) *(resp.,* (3.2), (3.3)*) is called a C-interior point (resp., a strong C-interior point, a weak-strong C-interior point with the pair (I_1,I_2)) of the CCS-system $\{C,C_i:\ i\in I\}$. Similarly, the notion of an interior point (resp., a strong interior point, a weak-strong interior point with the pair (I_1,I_2)) of the CCS-system $\{C,C_i:\ i\in I\}$ is defined.*

It is trivial that (3.2) $\Longrightarrow$ (3.1). The converse also holds in some cases, one of which will be described in terms of continuity of some set-valued functions. For set-valued functions there are many different notions of continuity. In Definitions 3.2 and 3.3 below, we recall two frequently used ones. We assume that Q is a compact metric space.

Definition 3.2. *Let Y be a normed linear space. Then the set-valued function $F:Q\to 2^Y\setminus\{\emptyset\}$ is said to be*

(i) *lower semicontinuous at $t_0\in Q$ if, for any $y_0\in F(t_0)$ and any $\epsilon>0$, there exists an open neighborhood $U(t_0)$ of t_0 such that for each $t\in U(t_0)$, $\mathbf{B}(y_0,\epsilon)\cap F(t)\neq\emptyset$;*

(ii) *locally uniform lower semicontinuous at $t_0\in Q$ if, for any $y_0\in F(t_0)$, there exists an open neighborhood $V(y_0)$ of y_0 such that for any $\epsilon>0$, there exists an open neighborhood $U(t_0)$ of t_0 such that $\mathbf{B}(y,\epsilon)\cap F(t)\neq\emptyset$ holds for each $t\in U(t_0)$ and each $y\in V(y_0)\cap F(t_0)$;*

(iii) *upper semicontinuous at $t_0\in Q$ if, for any open neighborhood V of $F(t_0)$, there exists an open neighborhood $U(t_0)$ of t_0 such that $F(t)\subseteq V$ for each $t\in U(t_0)$;*

(iv) *lower semicontinuous (resp., locally uniform lower semicontinuous, upper semicontinuous) on Q if it is lower semicontinuous (resp., locally uniform lower semicontinuous, upper semicontinuous) at each $t\in Q$.*

Definition 3.3 (cf. [29, p. 55]). *Let $F:Q\to 2^Y$ be a set-valued function defined on Q and let $t_0\in Q$. Then F is said to be*

(i) *upper Kuratowski semicontinuous at t_0 if, for any sequence $\{t_k\}\subseteq Q$, the relations $\lim_{k\to\infty}t_k=t_0$, $\lim_{k\to\infty}x_{t_k}=x_{t_0}$, $x_{t_k}\in F(t_k)$, $k=1,2,\ldots$, imply $x_{t_0}\in F(t_0)$;*

(ii) *lower Kuratowski semicontinuous at t_0 if, for any sequence $\{t_k\}\subseteq Q$, the relations $\lim_{k\to\infty}t_k=t_0$, $y_0\in F(t_0)$ imply $\lim_{k\to\infty}d_{F(t_k)}(y_0)=0$;*

(iii) *Kuratowski continuous at t_0 if F is both upper Kuratowski semicontinuous and lower Kuratowski semicontinuous at t_0;*

(iv) *Kuratowski continuous on Q if it is Kuratowski continuous at each point of Q.*

Remark 3.1. Clearly,

(i) F is upper semicontinuous $\Longrightarrow$ F is upper Kuratowski semicontinuous,

(ii) F is lower semicontinuous $\Longleftrightarrow$ F is lower Kuratowski semicontinuous.

Moreover, the converse of (i) holds provided that the union set $\cup_{t\in Q} F(t)$ is compact.

The following two propositions provide some useful reformulations regarding various lower semicontinuities. Since the proofs are similar, we shall only prove the first proposition.

PROPOSITION 3.1. *Let $F : Q \to 2^Y \setminus \{\emptyset\}$ be a set-valued function. Let $t_0 \in Q$. Then the following statements are equivalent.*

(i) *F is lower semicontinuous at t_0.*

(ii) *For any $y_0 \in F(t_0)$, there exists $y_t \in F(t)$ for each $t \in Q$ such that $\lim_{t\to t_0} \|y_t - y_0\| = 0$.*

(iii) *For any $y_0 \in F(t_0)$, $\lim_{t\to t_0} d_{F(t)}(y_0) = 0$.*

Proof. (i) $\Longrightarrow$ (ii). Let $y_0 \in F(t_0)$. Then, by (i), for each positive k there exists an open neighborhood $U_k(t_0)$ of t_0 such that

$$\mathbf{B}\left(y_0, \frac{1}{k}\right) \cap F(t) \neq \emptyset \quad \text{for each } t \in U_k(t_0). \tag{3.4}$$

Without loss of generality, we may assume that $U_{k+1}(t_0) \subseteq U_k(t_0)$ for each k and $\bigcap_{k\geq 1} U_k(t_0) = \{t_0\}$ because Q is a metric space. Now we construct $y_t \in F(t)$ for each $t \in Q$ as follows:

$$\begin{array}{ll} y_t \in F(t) & \text{if } t \in Q \setminus U_1(t_0), \\ y_t \in \mathbf{B}\left(y_0, \frac{1}{k}\right) \cap F(t) & \text{if } t \in U_k(t_0) \setminus U_{k+1}(t_0), \quad k = 1, 2, \dots, \\ y_0 & \text{if } t = t_0. \end{array} \tag{3.5}$$

Then $\lim_{t\to t_0} y_t = y_0$.

(ii) $\Longrightarrow$ (iii). It is trivial.

(iii) $\Longrightarrow$ (i). Let $y_0 \in F(x_0)$ and $\epsilon > 0$. By (iii), there exists an open neighborhood $U(t_0)$ of t_0 such that for each $t \in U(t_0)$, one has that $d_{F(t)}(y_0) < \epsilon$ and thus $\mathbf{B}(y_0, \epsilon) \cap F(t) \neq \emptyset$. Therefore (i) holds. The proof is complete. □

The following proposition can be proved similarly.

PROPOSITION 3.2. *Let $F : Q \to 2^Y \setminus \{\emptyset\}$ be a set-valued function. Let $t_0 \in Q$. Then the following statements are equivalent.*

(i) *F is locally uniform lower semicontinuous at t_0.*

(ii) *For any $y_0 \in F(t_0)$, there exists an open neighborhood $V(y_0)$ of y_0 such that for any $y \in V(y_0) \cap F(t_0)$, there exists $z_t(y) \in F(t)$ for each $t \in Q$ such that $\lim_{t\to t_0} \|z_t(y) - y\| = 0$ holds uniformly on $V(y_0) \cap F(t_0)$.*

(iii) *For any $y_0 \in F(t_0)$, there exists an open neighborhood $V(y_0)$ of y_0 such that*

$$\lim_{t\to t_0} \sup_{y\in V(y_0)\cap F(t_0)} d_{F(t)}(y) = 0.$$

(iv) *For any $y_0 \in F(t_0)$, there exists a neighborhood $V(y_0)$ of y_0 such that for any $\epsilon > 0$, there exists a neighborhood $U(t_0)$ of t_0 satisfying*

$$V(y_0) \cap F(t_0) \subseteq \bigcap_{t\in U(t_0)} F(t)^{\varepsilon},$$

where A^{ε} is defined by

$$A^{\varepsilon} = \{y \in Y : \ d_A(y) < \varepsilon\}.$$

Recalling our blanket assumption (1.2), and the definition of CCS-systems made at the beginning of this section, we state our first main result of this section.

THEOREM 3.1. *Let $\{C, C_i : i \in I\}$ be a CCS-system with base-set C, and let $\bar{x} \in C$. Suppose that the set-valued function $i \mapsto (\operatorname{aff} C) \cap C_i$ is locally uniform lower semicontinuous on I. Then $\bar{x}$ is a C-interior point of the system if and only if it is a strong C-interior point of the system.*

Proof. We need to prove only the necessity part. Assume without loss of generality that

$$0 = \bar{x} \in C \bigcap \left(\bigcap_{i \in I} \operatorname{rint}_C C_i \right). \tag{3.6}$$

Then $Z := \operatorname{aff} C$ is a vector subspace of X. It suffices to show that

$$0 \in C \bigcap \left(\operatorname{rint}_C \bigcap_{i \in I} C_i \right). \tag{3.7}$$

Clearly, we need only to show that

$$\inf_{i \in I} d_{\operatorname{bd}_Z C_i}(0) > 0. \tag{3.8}$$

(Indeed, if $d_{\operatorname{bd}_Z C_i}(0) > \gamma$, then $Z \cap \mathbf{B}(0, \gamma) \subseteq C_i$.) Suppose on the contrary that (3.8) does not hold. Then, by the compactness of I, there exist a convergent sequence $(i_n) \subseteq I$ (say with limit $i_0 \in I$) and a sequence (y_{i_n}) with $y_{i_n} \in \operatorname{bd}_Z C_{i_n}$ for each n such that $\lim_n \|y_{i_n}\| = 0$. Write

$$\widehat{C}_i = Z \cap C_i, \quad i \in I. \tag{3.9}$$

By assumptions, $i \mapsto \widehat{C}_i$ is locally uniform lower semicontinuous at i_0. By (iv) of Proposition 3.2 (applied to $0, i_0$ in place of y_0, t_0), there exists a $\delta \in (0, 1)$ such that for any $\varepsilon > 0$ there exists a neighborhood $U_\varepsilon(i_0)$ of i_0 such that

$$\mathbf{B}(0, \delta) \cap \widehat{C}_{i_0} \subseteq \bigcap_{i \in U_\varepsilon(i_0)} \widehat{C}_i^\varepsilon. \tag{3.10}$$

In view of (3.6), we may assume in addition that

$$\mathbf{B}(0, \delta) \cap Z \subseteq \widehat{C}_{i_0} \tag{3.11}$$

(take a smaller $\delta > 0$ if necessary). Combining the above two inclusions, we have

$$\mathbf{B}(0, \delta) \cap Z \subseteq \bigcap_{i \in U_\varepsilon(i_0)} \widehat{C}_i^\varepsilon. \tag{3.12}$$

Let us fix an $\varepsilon \in (0, \frac{\delta}{3})$ and take $\alpha > 0$ such that $\frac{3}{2}\varepsilon < \alpha < \frac{\delta}{2}$ (hence $\alpha < \frac{1}{2}$ as $\delta < 1$). We fix a natural number n which is large enough so that

$$\|y_{i_n}\| < \frac{\delta}{2} \quad \text{and} \quad i_n \in U_\varepsilon(i_0). \tag{3.13}$$

For simplicity of notations, we henceforth write i for the i_n with the above n. Since y_i is a (relative) boundary point of $C_i \cap Z = \widehat{C}_i$ in the vector subspace Z of X and since

$\widehat{C}_i$ has a nonempty relative interior (containing the origin) by (3.6), the separation theorem implies that $N_{\widehat{C}_i}(y_i)|_Z \neq \{0\}$. Hence, by the Hahn–Banach theorem, there exists $x_i^* \in N_{\widehat{C}_i}(y_i)$ such that $\|x_i^*|_Z\|$ is of norm 1. Take $x_i^\varepsilon \in Z$ such that $\|x_i^\varepsilon\| = 1$ and $\langle x_i^*, x_i^\varepsilon\rangle \geq 1 - \frac{\varepsilon}{2}$. Define $z_i := y_i + \alpha x_i^\varepsilon$. Then $z_i \in Z$,

$$\|(y_i + x_i^\varepsilon) - z_i\| = \|(1-\alpha)x_i^\varepsilon\| = 1 - \alpha,$$

and it follows from the triangle inequality that, for any $y \in \widehat{C}_i$,

$$\begin{aligned}\|z_i - y\| &\geq \|(y_i + x_i^\varepsilon) - y\| - (1-\alpha)\\ &\geq \langle x_i^*, y_i - y\rangle + \langle x_i^*, x_i^\varepsilon\rangle - (1-\alpha)\\ &\geq \langle x_i^*, x_i^\varepsilon\rangle - (1-\alpha)\\ &\geq (1 - \tfrac{\varepsilon}{2}) - (1-\alpha)\\ &= \alpha - \tfrac{\varepsilon}{2} > \varepsilon.\end{aligned}$$

Therefore $z_i \notin \widehat{C}_i^\varepsilon$. This contradicts (3.12) and (3.13) because

$$\|z_i\| \leq \|z_i - y_i\| + \|y_i\| = \|\alpha x_i^\varepsilon\| + \|y_i\| < \alpha + \frac{\delta}{2} \leq \delta.$$

Thus (3.8) must hold and the proof is complete. □

Let A and $\widehat{C}$ be two closed convex subsets of X with $0 \in \operatorname{rint}_{\widehat{C}} A$. In the following we will show that A admits a "$\widehat{C}$-extended Minkowski functional" p_A in the sense that p_A is a continuous sublinear functional on X such that its restriction $p_A|_{\operatorname{aff}\widehat{C}}$ equals the Minkowski functional of $A \cap \operatorname{aff}\widehat{C}$ in the vector subspace $\operatorname{aff}\widehat{C}$ of X. Note that in this case one has, for each $z \in \operatorname{aff}\widehat{C}$,

$$p_A(z) \leq 1 \Longleftrightarrow z \in A, \tag{3.14}$$

$$p_A(z) = 1 \Longleftrightarrow z \in \operatorname{bd}_{\widehat{C}} A. \tag{3.15}$$

Lemma 3.1. *Suppose that* $0 \in \operatorname{rint}_{\widehat{C}} A$*; that is,*

$$0 \in \mathbf{B}(0,\alpha) \cap \operatorname{aff}\widehat{C} \subseteq A \tag{3.16}$$

for some $\alpha > 0$*, where* A *and* $\widehat{C}$ *are closed convex subsets of* X*. Denote the closure of* $\operatorname{aff}\widehat{C}$ $(= \operatorname{span}\widehat{C})$ *by* Z *and let* $\widetilde{A}$ *denote the closed convex hull of the set* $(A \cap Z) \cup (\mathbf{B}(0,\alpha))$*. Then*

(i) $\widetilde{A}$ *is a closed convex set in* X *with nonempty interior such that*

$$\widetilde{A} \cap Z = A \cap Z \quad \textit{and} \quad 0 \in \operatorname{int}\widetilde{A}. \tag{3.17}$$

(ii) *The corresponding Minkowski functional* $q_{\widetilde{A}}$ *(in the usual sense) on* X *has the properties*

$$q_{\widetilde{A}}(x) \leq \frac{1}{\alpha}\|x\| \quad \textit{for each } x \in X \tag{3.18}$$

and

$$q_{\widetilde{A}}(x) = \inf\{\lambda \geq 0 :\ x \in \lambda(A \cap Z)\} \quad \textit{for each } x \in Z \tag{3.19}$$

(that is, $q_{\widetilde{A}}$ *is a* $\widehat{C}$*-extended Minkowski functional of* A*).*

Proof. Let D denote the convex hull of the set $(A\cap Z)\cup(\mathbf{B}(0,\alpha))$. Then $\widetilde{A}=\overline{D}$ and $\mathbf{B}(0,\alpha)\subseteq D$. Hence, by elementary functional analysis, the Minkowski functional of $\widetilde{A}$ coincides with that of D and (3.18) holds. Hence, to prove (3.19) it suffices to show that

$$D\cap Z=A\cap Z. \tag{3.20}$$

Let $x=\lambda_1 a+\lambda_2 b\in D\cap Z$ with $a\in A\cap Z$, $b\in\mathbf{B}(0,\alpha)$, and $\lambda_1,\lambda_2\in(0,1)$ such that $\lambda_1+\lambda_2=1$. Then $b\in Z$. We claim that $b\in A$. In fact, since $b\in Z$, there exists a sequence $\{b_k\}\subset\operatorname{aff}\widehat{C}$ such that $b_k\to b$. If $b\in\operatorname{int}\mathbf{B}(0,\alpha)$, then $b_k\in\mathbf{B}(0,\alpha)$ for all k large enough. This implies that $b_k\in\mathbf{B}(0,\alpha)\cap\operatorname{aff}\widehat{C}$ for all such k. Therefore, by (3.16), $b_k\in A$ and hence $b\in A$. Thus assume that $b\in\operatorname{bd}\mathbf{B}(0,\alpha)$. Define $\tilde{b}_k=\frac{\alpha b_k}{\|b_k\|}$ for each k. Then $\tilde{b}_k\in\mathbf{B}(0,\alpha)\cap\operatorname{aff}\widehat{C}$ and $\tilde{b}_k\to b$ for each k. Hence $b\in A$ by (3.16). Therefore our claim stands and $x\in A$ as A is convex. This shows that (3.20), and hence (3.19), are true. To verify (3.17), let $z\in\widetilde{A}\cap Z$. Then $q_{\widetilde{A}}(z)\le 1$ and one can apply (3.19) to conclude that $z\in A$ because A is closed and $z\in Z$. Thus (3.17) is seen to be true and the proof is complete. ◻

Note. The set $\widetilde{A}$ will be referred to as a $\widehat{C}$-Minkowski extension of A (though it also depends on α in (3.16)).

For the remainder of this paper, $\{C,C_i:\ i\in I\}$ denotes a CCS-system with base-set C as defined at the beginning of this section. Now we state the second main result of this section.

THEOREM 3.2. *Let $\bar{x}\in C\cap(\cap_{i\in I}C_i)$ and let $\widehat{C}:=C-\bar{x}$, $\widehat{C}_i:=C_i-\bar{x}$ for each $i\in I$. Then the following statements are equivalent.*

(i) *$\bar{x}$ is a strong C-interior point of the CCS-system $\{C,C_i:\ i\in I\}$, namely,*

$$\bar{x}\in C\cap\operatorname{rint}_C\left(\cap_{i\in I}C_i\right). \tag{3.21}$$

(ii) *For each $i\in I$, there exists a $\widehat{C}$-extended Minkowski functional $p_{\widehat{C}_i}$ of the set $\widehat{C}_i$ such that the sup-function $P(\cdot)$ of $\{p_{\widehat{C}_i}(\cdot)\}$ defined by*

$$P(x):=\sup_{i\in I}p_{\widehat{C}_i}(x),\quad x\in X \tag{3.22}$$

is continuous on X.

Moreover, if we add an additional assumption that the set-valued map $i\mapsto(\operatorname{aff}C)\cap C_i$ is lower semicontinuous, then (ii) *above can be replaced by a stronger one, as follows:*

$\widetilde{\text{(ii)}}$ (ii) *holds and $i\mapsto p_{\widehat{C}_i}(x)$ is upper semicontinuous for each $x\in X$.*

Proof. (ii) $\Longrightarrow$ (i). Let $Z:=\operatorname{aff}\widehat{C}=\operatorname{span}\widehat{C}$. By (3.14), $Z\cap\widehat{C}_i=\{z\in Z:\ p_{\widehat{C}_i}(z)\le 1\}$ and hence

$$Z\bigcap\left(\bigcap_{i\in I}\widehat{C}_i\right)=\{z\in Z:\ P(z)\le 1\}.$$

By the continuity assumption on P, it follows that $0\in\operatorname{rint}_Z(\cap_{i\in I}\widehat{C}_i)$ and hence $\bar{x}\in C\cap\operatorname{rint}_C(\cap_{i\in I}C_i)$. Thus (3.21) is seen to hold.

(i) $\Longrightarrow$ (ii). By (3.21), there exists $\alpha>0$ such that

$$\mathbf{B}(0,\alpha)\cap Z\subseteq\widehat{C}_i\quad\text{for each } i\in I. \tag{3.23}$$

Then, by Lemma 3.1, there exists a $\widehat{C}$-extended Minkowski functional $p_{\widehat{C}_i}$ of $\widehat{C}_i$ such that $p_{\widehat{C}_i}$ is the Minkowski functional of the closed convex hull of $(\widehat{C}_i \cap Z) \cup (\mathbf{B}(0,\alpha))$ and, in particular,

$$p_{\widehat{C}_i}(x) \leq \frac{1}{\alpha}\|x\| \quad \text{for each } x \in X. \tag{3.24}$$

Hence, by definition of P,

$$P(x) \leq \frac{1}{\alpha}\|x\| \quad \text{for each } x \in X, \tag{3.25}$$

and thus P is continuous by an elementary argument. This establishes the implication (i) $\Longrightarrow$ (ii).

For the remainder of the proof we assume, in addition, that the set-valued map $i \mapsto (\operatorname{aff} C) \cap C_i$ is lower semicontinuous and hence the set-valued map $i \mapsto Z \cap \widehat{C}_i$ is lower semicontinuous. Then, to prove (i) $\Longrightarrow$ $\widetilde{\text{(ii)}}$ it remains to show for any $i_0 \in I$ and any $x \in X$ that

$$\limsup_{i \to i_0} p_{\widehat{C}_i}(x) \leq p_{\widehat{C}_{i_0}}(x). \tag{3.26}$$

Suppose not. Then there exist $i_0 \in I$ and $x \in X$ such that

$$\limsup_{i \to i_0} p_{\widehat{C}_i}(x) > 1 > p_{\widehat{C}_{i_0}}(x). \tag{3.27}$$

Then $x \in \operatorname{co}((\widehat{C}_{i_0} \cap Z) \cup (\mathbf{B}(0,\alpha)))$ and so $x = \lambda_1 b + \lambda_2 z$ for some $b \in \mathbf{B}(0,\alpha)$, $z \in \widehat{C}_{i_0} \cap Z$, and some $\lambda_1, \lambda_2 \in [0,1]$ with $\lambda_1 + \lambda_2 = 1$. Since the set-valued function $i \mapsto \widehat{C}_i \cap Z$ is lower semicontinuous at i_0, there exists $z_i \in \widehat{C}_i \cap Z$ for each $i \in I$ such that $z_i \to z$ as $i \to i_0$. Define $x_i = \lambda_1 b + \lambda_2 z_i$ for each $i \in I$. Then $x_i \in \operatorname{co}((\widehat{C}_i \cap Z) \cup (\mathbf{B}(0,\alpha)))$ and thus $p_{\widehat{C}_i}(x_i) \leq 1$. Consequently, it follows from (3.25) that

$$p_{\widehat{C}_i}(x) \leq p_{\widehat{C}_i}(x - x_i) + p_{\widehat{C}_i}(x_i) \leq \frac{1}{\alpha}\|x - x_i\| + 1$$

and hence that $\limsup_{i \to i_0} p_{\widehat{C}_i}(x) \leq 1$. This contradicts (3.27). Therefore (3.26) must hold for each $i_0 \in I$ and $x \in X$. □

The following proposition deals with a special case in Theorem 3.2 by deleting the words "relative" and "$\widehat{C}$-extended," respectively, from (i) and (ii). We will omit the proof as it is similar to that of Theorem 3.2.

THEOREM 3.3. *Let $\bar{x} \in C \cap (\cap_{i\in I} C_i)$. Then the following statements are equivalent:*

(i) *$\bar{x}$ is a strong interior point of the CCS-system $\{C, C_i : i \in I\}$; namely,*

$$\bar{x} \in C \cap \operatorname{int}(\cap_{i\in I} C_i).$$

(ii) *$\bar{x} \in \operatorname{int} C_i$ for each $i \in I$ and the sup-function $P(\cdot)$ of $\{p_{\widehat{C}_i}(\cdot)\}$ is continuous on X, where $p_{\widehat{C}_i}$ is the Minkowski functional of the set $\widehat{C}_i := C - \bar{x}$.*

Moreover, if we add an additional assumption that the set-valued map $i \mapsto C_i$ is lower semicontinuous, then the above (ii) *can be replaced by a stronger one, as follows:*

$\widetilde{\text{(ii)}}$ (ii) *holds and $i \mapsto p_{\widehat{C}_i}(x)$ is upper semicontinuous for each $x \in X$.*

4. Interior-point condition and the strong CHIP. For the remainder of this paper, we assume that I is a compact metric space and that $\{C, C_i :\ i \in I\}$ is a CCS-system with base-set C as in the beginning of the preceding section. Our main results are to provide sufficient conditions for ensuring the strong CHIP. For $x_0 \in C \cap (\cap_{i \in I} C_i)$, let $I_C^{\mathrm{rb}}(x_0) = \{i \in I :\ x_0 \in \mathrm{bd}_C\, C_i\}$. Since $\mathrm{bd}_C\, C_i = \mathrm{bd}\, C_i \setminus \mathrm{int}_C\, C_i$,

$$I_C^{\mathrm{rb}}(x_0) \subseteq \{i \in I :\ x_0 \in \mathrm{bd}\, C_i\}. \tag{4.1}$$

THEOREM 4.1. *Let $x_0 \in C \cap (\cap_{i \in I} C_i)$. Then the CCS-system $\{C, C_i :\ i \in I\}$ has the strong CHIP at x_0 if the following conditions are satisfied.*

(a) *The system $\{C, C_i :\ i \in I\}$ satisfies the strong C-interior-point condition.*

(b) *The set-valued mapping $i \mapsto (\mathrm{aff}\, C) \cap C_i$ is lower semicontinuous on I.*

(c) *The pair $\{\mathrm{aff}\, C, C_i\}$ has the strong CHIP at x_0 for each $i \in I$.*

(d) *Either C is finite dimensional or $I_C^{\mathrm{rb}}(x_0)$ is a finite set.*

Moreover, the same conclusion also holds if (a), (b), *and* (c) *are replaced simultaneously by* (a*) *and* (b*).

(a*) *The same as* (a) *but delete the word "relative."*

(b*) *The set-valued mapping $i \mapsto C_i$ is lower semicontinuous on I.*

Remark 4.1. In view of Theorem 2.2, (a*) $\Longrightarrow$ (c) because (a*) implies $\mathrm{int}\, C_i \supseteq \mathrm{int}(\cap_{i \in I} C_i) \cap C \neq \emptyset$.

Proof of Theorem 4.1. By the assumptions (a) and (b), let $\bar{x}$, $\widehat{C}$, $\widehat{C}_i$, $p_{\widehat{C}_i}$, and P be as in parts (i), $\widetilde{\text{(ii)}}$ of Theorem 3.2. In particular, P is continuous, the function $i \mapsto p_{\widehat{C}_i}(x)$ is upper semicontinuous for each $x \in X$, and for each $x \in \mathrm{aff}\, C$ and $i \in I$ it holds that

$$p_{\widehat{C}_i}(x - \bar{x}) \leq 1 \Longleftrightarrow x \in C_i, \tag{4.2}$$

$$p_{\widehat{C}_i}(x - \bar{x}) = 1 \Longleftrightarrow x \in \mathrm{bd}_C\, C_i, \tag{4.3}$$

where (4.3) holds by (3.15). [*Note.* The above considerations are valid by Lemma 3.1 if one assumes (a*) + (b*) instead of (a) + (b); in fact, in this case, (4.2) and (4.3) hold for all $x \in X$ (not only those x in $\mathrm{aff}\, C$).]

Define, for each $i \in I$,

$$g_i(x) = p_{\widehat{C}_i}(x - \bar{x}) - 1 \quad \text{for each } x \in X$$

and let $G : X \to \mathbb{R}$ be defined by $G(x) := \sup_{i \in I} g_i(x)$ for each $x \in X$. Then G is continuous and the function $i \mapsto g_i(x)$ is upper semicontinuous for each $x \in X$. Thus, one can apply Theorem 2.1 to conclude that

$$N_C(x_0) + \partial G(x_0) = N_C(x_0) + \mathrm{cone} \sum_{i \in I(x_0)} \partial\, g_i(x_0), \tag{4.4}$$

provided that the following condition (d′) is satisfied:

(d′) C is finite dimensional or $I(x_0)$ is finite.

To prove the theorem, we need only to prove the inclusion

$$N_{C \cap (\cap_{i \in I} C_i)}(x_0) \subseteq N_C(x_0) + \sum_{i \in I} N_{C_i}(x_0), \tag{4.5}$$

as the reverse inclusion is evident. Note that, by (4.2),

$$\begin{aligned} N_{C\bigcap(\bigcap_{i\in I} C_i)}(x_0) &= N_{C\bigcap(\bigcap_{i\in I} g_i^{-1}(\mathbb{R}_-))}(x_0) \\ &= N_{C\bigcap(G^{-1}(\mathbb{R}_-))}(x_0) \\ &= N_C(x_0) + N_{G^{-1}(\mathbb{R}_-)}(x_0) \end{aligned} \tag{4.6}$$

thanks to Theorem DLW because $\bar{x} \in C \cap \operatorname{int} G^{-1}(\mathbb{R}_-)$ as $G(\bar{x}) = -1$. Thus (4.5) is seen to hold if $N_{G^{-1}(\mathbb{R}_-)}(x_0) = 0$. Therefore we may henceforth assume that $G(x_0) = 0$. Then, referring to the corresponding definition stated in (2.11), $I(x_0) = \{i \in I : \ g_i(x_0) = 0\}$. Since $x_0 \in C$ it follows from (4.3) that

$$I(x_0) = I_C^{\mathrm{rb}}(x_0). \tag{4.7}$$

Thus, by assumption (d), the condition (d′) holds. Recall from [7, Corollary 1, p. 56] that

$$N_{g_i^{-1}(\mathbb{R}_-)}(x_0) = \begin{cases} \operatorname{cone} \partial g_i(x_0), & i \in I(x_0), \\ 0, & i \notin I(x_0) \end{cases} \tag{4.8}$$

and, similarly,

$$N_{G^{-1}(\mathbb{R}_-)}(x_0) = \operatorname{cone} \partial G(x_0). \tag{4.9}$$

Hence, by (4.6) and (4.4), we have

$$\begin{aligned} N_{C\bigcap(\bigcap_{i\in I} C_i)}(x_0) &= N_C(x_0) + \operatorname{cone} \partial G(x_0) \\ &= N_C(x_0) + \operatorname{cone} \sum_{i\in I(x_0)} \partial\, g_i(x_0) \\ &= N_C(x_0) + \sum_{i\in I(x_0)} N_{g_i^{-1}(\mathbb{R}_-)}(x_0) \\ &\subseteq N_C(x_0) + \sum_{i\in I(x_0)} N_{C_i\cap \operatorname{aff} C}(x_0) \end{aligned} \tag{4.10}$$

because $g_i^{-1}(\mathbb{R}_-) \supseteq C_i \cap \operatorname{aff} C$. This implies that (4.5) holds because for each $i \in I(x_0) = I_C^{\mathrm{rb}}(x_0)$, one has from assumption (c) that

$$N_{C_i\cap \operatorname{aff} C}(x_0) = N_{C_i}(x_0) + N_{\operatorname{aff} C}(x_0)$$

(note also that $N_C(x_0) = N_C(x_0) + N_{\operatorname{aff} C}(x_0)$ as $C \subseteq \operatorname{aff} C$). Thus the first part of the theorem is proved. The proof of the second part is almost the same because if the assumptions (a) + (b) + (c) are replaced by (a*) + (b*), then (4.7) remains true (noting that $\bar{x} \in C \cap \operatorname{int} C_i$ and $x_0 \in C \cap C_i$ for each $i \in I$ and that $N_{C_i\cap \operatorname{aff} C}(x_0)$ in (4.10) is to be replaced by $N_{C_i}(x_0)$). □

Remark 4.2. As assumption (c) is used only at the end of the proof, (4.10) is valid even if (c) is not assumed in Theorem 4.1.

Remark 4.3. The result in the first part of Theorem 4.1 is not true if the assumptions are replaced by (a), (c), (d), and (b*) (instead of (b)), as shown by the following example.

Example 4.1. Let $X = \mathbb{R}^2$ and let $I = \{0, 1, \frac{1}{2}, \ldots\}$. For each $i \in I$, let C_i be the closed convex subset of $\mathbb{R}^2$ that is bounded by four line segments l_1, l_2, l_3, l_4 and the curve defined by

$$t_2 = i(t_1 + 1 + i)^2 \quad \text{for all } t_1 \in [-2, -1 - i],$$

where l_1, l_2, l_3, and l_4 are defined as follows:

$$\begin{aligned} l_1: &\quad t_1 = -2 \quad \text{for all } t_2 \in [i(1-i)^2, 1],\\ l_2: &\quad t_1 = 1 \quad \text{for all } t_2 \in [0, 1],\\ l_3: &\quad t_2 = 1 \quad \text{for all } t_1 \in [-2, 1],\\ l_4: &\quad t_2 = 0 \quad \text{for all } t_1 \in [-1 - i, 1]. \end{aligned}$$

Let $C = \{(t_1, 0) : t_1 \in [-2, 1]\}$. Clearly, $\bar{x} = (0, 0)$ is a strong C-interior point of the system $\{C, C_i : i \in I\}$. Note that $i_0 = 0$ is the only limit point of I and that C_0 is the rectangle $[-2, 1] \times [0, 1]$. Hence it is easy to see that the set-valued function $i \mapsto C_i$ is lower semicontinuous on I. However, since $(\operatorname{aff} C) \cap C_i$ is the segment $[-1-i, 1] \times \{0\}$ for each $i \in I \backslash \{0\}$ and $(\operatorname{aff} C) \cap C_0 = [-2, 1] \times \{0\}$, the set-valued function $i \mapsto (\operatorname{aff} C) \cap C_i$ is not lower semicontinuous on I. Let $x_0 = (-1, 0)$. It is clear that $\{\operatorname{aff} C, C_i\}$ has the strong CHIP at x_0 for each $i \in I$. But the system $\{C, C_i : i \in I\}$ does not have the strong CHIP at x_0 because $C \cap (\cap_{i \in I} C_i) = \{(t_1, 0) : t_1 \in [-1, 1]\}$, $N_{C \cap (\cap_{i \in I} C_i)}(x_0) = \{(t_1, t_2) : t_1 \leq 0\}$, and $N_C(x_0) + \sum_{i \in I} N_{C_i}(x_0) = \{(t_1, t_2) : t_1 = 0\}$. □

Remark 4.4. In general, $\{\operatorname{aff} C, C_1\}$ may not have the strong CHIP even if the C-interior-point condition is satisfied by the system $\{C, C_1\}$. For example, let $I = \{1\}$, $X = \mathbb{R}^2$, and C be the line $t_2 = 0$ while C_1 is bounded by the lines $t_2 = 0$, $t_2 = 2$, $t_1 = 1$, and the half-circle $t_1^2 + (t_2 - 1)^2 = 1$, $t_1 \in [-1, 0]$. Clearly, $\{C, C_1\}$ satisfies the C-interior-point condition but $\{\operatorname{aff} C, C_1\}$ does not have the strong CHIP at $x_0 = (0, 0)$. This example also shows that (c) cannot be dropped in the first part of Theorem 4.1.

Our next two theorems address the case when some C_i might have an empty interior. We will use the notion that a subset in X is finite codimensional.

Definition 4.1. *Let A and B be two nonempty convex subsets of X. We say that A is*

(i) *finite codimensional in B if the closed subspace $\overline{\operatorname{span} B} \cap \overline{\operatorname{span} A}$ is a finite codimensional subspace of $\overline{\operatorname{span} B}$;*

(ii) *finite codimensional if $\overline{\operatorname{span} A}$ is finite codimensional in X.*

Obviously, if B is finite dimensional, any nonempty convex subset A of X is finite codimensional in B.

Theorem 4.2. *Let $x_0 \in C \cap (\cap_{i \in I} C_i)$. The system $\{C, C_i : i \in I\}$ has the strong CHIP at x_0 if the following conditions are satisfied.*

(a) *The system $\{C, C_i : i \in I\}$ satisfies the weak-strong C-interior-point condition with (I_1, I_2).*

(b) *The set-valued mapping $i \mapsto (\operatorname{aff} C) \cap C_i$ is lower semicontinuous on I.*

(c) *The pair $\{\operatorname{aff} C, C_i\}$ has the strong CHIP at x_0 for each $i \in I \setminus (I_1 \cup I_2)$.*

(d) *C is finite dimensional or $I_C^{\mathrm{rb}}(x_0)$ is finite.*

(e) *C_i is finite codimensional for each $i \in I_1$.*

Moreover, the same conclusion also holds if (a), (b), *and* (c) *in the above assumptions are replaced simultaneously by* (a*) *and* (b*).

(a*) *The same as* (a) *but delete the word "relative."*

(b*) *The set-valued mapping $i \mapsto C_i$ is lower semicontinuous on I.*

Proof. By (a), there exist $\bar{x} \in X$ and two disjoint finite subsets I_1, I_2 of I such that $\bar{x}$ is a weak-strong C-interior point with (I_1, I_2) of the CCS-system $\{C, C_i : i \in I\}$; that is, C_i is a polyhedron for each $i \in I_2$ and

$$\bar{x} \in \operatorname{ri} C \bigcap \left(\operatorname{rint}_C \bigcap_{i \in I_0} C_i \right) \bigcap \left(\bigcap_{i \in I_1} \operatorname{ri} C_i \right) \bigcap \left(\bigcap_{i \in I_2} C_i \right), \tag{4.11}$$

where $I_0 = I \setminus (I_1 \cup I_2)$. Let J denote the closure of I_0; namely, J equals the union of I_0 with I^l, where the subset I^l of $I_1 \cup I_2$ is defined by

$$I^l := \{i \in I_1 \cup I_2 : \ i \text{ is a limit point of } I\}. \tag{4.12}$$

For each $i \in I^l$, we define $\vec{C}_i$ by

(4.13)
$$\begin{aligned} \vec{C}_i : \quad &= \{x \in X : \exists x_j \in (\operatorname{aff} C) \cap C_j \text{ for all } j \in I \setminus \{i\} \text{ such that } \lim_{j \to i} x_j = x\} \\ &= \{x \in X : \exists x_j \in (\operatorname{aff} C) \cap C_j \text{ for all } j \in I_0 \text{ such that } \lim_{j \to i} x_j = x\}, \end{aligned}$$

thanks to the fact that $I_1 \cup I_2$ is a finite set. By assumption (b) and Proposition 3.1, we have

$$(\operatorname{aff} C) \cap C_i \subseteq \vec{C}_i \quad \text{for each } i \in I^l. \tag{4.14}$$

Moreover,

$$\bar{x} \in \operatorname{ri} C \bigcap \left(\operatorname{rint}_C \bigcap_{i \in I_0} C_i \right) \bigcap \left(\operatorname{rint}_C \bigcap_{i \in I^l} \vec{C}_i \right). \tag{4.15}$$

In fact, since $\bar{x} \in \operatorname{rint}_C \bigcap_{i \in I_0} C_i$, there exists $\delta > 0$ such that $(\operatorname{aff} C) \cap \mathbf{B}(\bar{x}, \delta) \subseteq C_j$ for each $j \in I_0$. From (4.13) it follows that $(\operatorname{aff} C) \cap \mathbf{B}(\bar{x}, \delta) \subseteq \vec{C}_i$ for each $i \in I^l$. Therefore $\bar{x} \in \operatorname{rint}_C(\bigcap_{i \in I^l} \vec{C}_i)$ and (4.15) holds. Note further that each $\vec{C}_i$ is convex and closed. In fact, let $i \in I^l$ and let $\{x_n\} \subseteq \vec{C}_i$ be such that $x_n \to z$. Then $\lim_{j \to i} d_{C_j \cap \operatorname{aff} C}(x_n) = 0$ for each $n = 1, 2, \ldots$. Since

$$d_{C_j \cap \operatorname{aff} C}(z) \leq \|x_n - z\| + d_{C_j \cap \operatorname{aff} C}(x_n),$$

we have that $\lim_{j \to i} d_{C_j \cap \operatorname{aff} C}(z) = 0$. This implies that $z \in \vec{C}_i$ and so $\vec{C}_i$ is closed. The convexity of $\vec{C}_i$ follows from the convexity of the sets $(\operatorname{aff} C) \cap C_j$ $(j \in I)$ and the definition of $\vec{C}_i$. Recall that $J = I_0 \cup I^l$ and define $\vec{C}_i = C_i$ for each $i \in I_0$. Then J is compact and $\{C, \vec{C}_i : i \in J\}$ is a CCS-system with the following properties:

(i) $\{C, \vec{C}_i : i \in J\}$ satisfies the strong C-interior-point condition; in fact,

$$\bar{x} \in \operatorname{ri} C \bigcap \operatorname{rint}_C \left(\bigcap_{j \in J} \vec{C}_j \right) \tag{4.16}$$

(see (4.15)).

(ii) The set-valued function $j \mapsto (\operatorname{aff} C) \cap \vec{C}_j$ is lower semicontinuous on J.

(iii) Either C is finite dimensional or $\vec{I}_C^{\mathrm{rb}}(x_0)$ is a finite set, where

$$\vec{I}_C^{\mathrm{rb}}(x_0) := \{i \in J :\ x_0 \in \mathrm{bd}_C\, \vec{C}_i\}. \tag{4.17}$$

In fact, (ii) follows from assumption (b) and the definition of $\vec{C}_j$. Moreover, (iii) follows from (d) and the fact that $\vec{I}_C^{\mathrm{rb}}(x_0) \subseteq I_C^{\mathrm{rb}}(x_0)$ (because of (4.14) and $x_0 \in (\mathrm{aff}\, C) \cap C_i)$. Thus, by Remark 4.2, we have that

$$N_{C\cap(\cap_{j\in J}\vec{C}_j)}(x_0) \subseteq N_C(x_0) + \sum_{j\in J} N_{(\mathrm{aff}\, C)\cap\vec{C}_j}(x_0). \tag{4.18}$$

We will show that

$$N_{(\mathrm{aff}\, C)\cap\vec{C}_j}(x_0) \subseteq N_C(x_0) + N_{C_j}(x_0) \quad \text{for each } j \in J. \tag{4.19}$$

This inclusion is simply from assumption (c) if $j \in I_0$. Next consider the case when $j \in J \cap I_2$. In this case one has $j \in I^l \cap I_2$ and it follows from (4.14) that $(\mathrm{aff}\, C) \cap C_j \subseteq \vec{C}_j$. Consequently, one has

$$N_{(\mathrm{aff}\, C)\cap\vec{C}_j}(x_0) \subseteq N_{(\mathrm{aff}\, C)\cap C_j}(x_0) \subseteq N_{C\cap C_j}(x_0) = N_C(x_0) + N_{C_j}(x_0),$$

where the last equality holds by Theorem 2.2 (which is applicable here because C_j is a polyhedron and $\bar{x} \in \mathrm{ri}\, C \cap C_j$). It remains to consider the case when $j \in I_1 \cap I^l$ for (4.19). To do this and for a later application, let us consider a general $j \in I_1$ (for the time being regardless of whether $j \in I^l$ or not). Then by (4.11) and definitions, $\bar{x} \in \mathrm{ri}\, C_j = \mathrm{rint}_{C_j} C_j$. Hence, by Lemma 3.1, $C_j - \bar{x}$ admits a (C_j)-Minkowski extension $\widetilde{C}_j - \bar{x}$: $\widetilde{C}_j$ is a closed convex set such that

$$\bar{x} \in \mathrm{int}\, \widetilde{C}_j \text{ and } \overline{\mathrm{aff}\, C_j} \cap \widetilde{C}_j = \overline{\mathrm{aff}\, C_j} \cap C_j = C_j = (\mathrm{aff}\, C_j) \cap \widetilde{C}_j \quad \text{for all } j \in I_1. \tag{4.20}$$

Combining this with (4.14),

$$(\mathrm{aff}\, C) \cap \overline{\mathrm{aff}\, C_j} \cap \widetilde{C}_j \subseteq (\mathrm{aff}\, C) \cap \vec{C}_j \quad \text{for each } j \in I_1 \cap I^l. \tag{4.21}$$

Thus if $j \in I_1 \cap I^l$, then it follows from (4.21) and Theorem 2.2 that

$$\begin{aligned} N_{(\mathrm{aff}\, C)\cap\vec{C}_j}(x_0) &\subseteq N_{\mathrm{aff}\, C\cap\overline{\mathrm{aff}\, C_j}\cap\widetilde{C}_j}(x_0) \\ &\subseteq N_{(\mathrm{aff}\, C)\cap\overline{\mathrm{aff}\, C_j}}(x_0) + N_{\widetilde{C}_j}(x_0) \\ &\subseteq N_{\mathrm{aff}\, C}(x_0) + N_{\overline{\mathrm{aff}\, C_j}}(x_0) + N_{\widetilde{C}_j}(x_0) \\ &\subseteq N_{\mathrm{aff}\, C}(x_0) + N_{\overline{\mathrm{aff}\, C_j}\cap\widetilde{C}_j}(x_0) \\ &\subseteq N_C(x_0) + N_{C_j}(x_0) \end{aligned}$$

thanks to (4.20), and thus (4.19) is verified. Here we have used Theorem 2.2 (twice) which is applicable as $\bar{x} \in \mathrm{int}\, \widetilde{C}_j \cap \overline{\mathrm{aff}\, C_j} \cap \mathrm{ri}\,(\mathrm{aff}\, C)$ and $\overline{\mathrm{aff}\, C_j}$ is a polyhedron (being an affine subspace of finite codimension) by assumption (e). Therefore (4.19) is established in all possible cases. Combining (4.19) and (4.18), we have

$$N_{C\cap(\cap_{j\in J}\vec{C}_i)}(x_0) \subseteq N_C(x_0) + \sum_{j\in J} N_{C_j}(x_0). \tag{4.22}$$

Recalling $\vec{C}_i = C_i$ for each $i \in I_0$, $J = I_0 \cup I^l$ and $I = J \cup I_1 \cup I_2$, it is a routine matter to verify from (4.14) and (4.20) that

$$C \bigcap \left(\bigcap_{i \in I} C_i \right) = \left\{ C \bigcap \left(\bigcap_{i \in J} \vec{C}_i \right) \bigcap \left(\bigcap_{i \in I_1} \overline{\operatorname{aff} C_i} \right) \bigcap \left(\bigcap_{i \in I_2} C_i \right) \right\} \bigcap \left(\bigcap_{i \in I_1} \widetilde{C}_i \right). \tag{4.23}$$

(For example, if x is a member of the set on the right-hand side of (4.23) and if $i \in I_1$, then $x \in C_i$ by (4.20) and so it is not difficult to verify that x belongs to the set on the left-hand side of (4.23).) Moreover, by virtue of the general inclusion property, $\operatorname{ri} C \bigcap (\operatorname{rint}_C \bigcap_{i \in J} \vec{C}_i) \subseteq \operatorname{ri}(C \bigcap (\bigcap_{i \in J} \vec{C}_i))$ (which can be verified by definition). This with (4.16) implies that $\bar{x} \in \operatorname{ri}(C \bigcap (\bigcap_{i \in J} \vec{C}_i))$ and hence it follows from (4.11) and (4.20) that

$$\bar{x} \in \operatorname{ri} \left(C \bigcap \left(\bigcap_{i \in J} \vec{C}_i \right) \right) \bigcap \left(\bigcap_{i \in I_1} \overline{\operatorname{aff} C_i} \right) \bigcap \left(\bigcap_{i \in I_2} C_i \right) \bigcap \left(\operatorname{int} \bigcap_{i \in I_1} \widetilde{C}_i \right). \tag{4.24}$$

Thus Theorem 2.2 is applicable to computing the normal cone of the set on the left-hand side of (4.23) at x_0 (noting that each C_i with $i \in I_2$ is a polyhedron and that each $\overline{\operatorname{aff} C_i}$ with $i \in I_1$ is also a polyhedron as noted before):

$$\begin{aligned}
&N(x_0; C \cap (\cap_{i \in I} C_i)) \\
&\quad = N(x_0; (C \cap (\cap_{i \in J} \vec{C}_i)) \cap (\cap_{i \in I_1} \overline{\operatorname{aff} C_i}) \cap (\cap_{i \in I_2} C_i)) + \sum_{i \in I_1} N(x_0; \widetilde{C}_i) \\
&\quad = N(x_0; C \cap (\cap_{i \in J} \vec{C}_i)) + \sum_{i \in I_1} N(x_0; \overline{\operatorname{aff} C_i}) + \sum_{i \in I_2} N(x_0; C_i) + \sum_{i \in I_1} N(x_0; \widetilde{C}_i) \\
&\quad \subseteq N(x_0; C) + \sum_{i \in J} N(x_0; C_i) + \sum_{i \in I_1} N(x_0; \overline{\operatorname{aff} C_i} \cap \widetilde{C}_i) + \sum_{i \in I_2} N(x_0; C_i),
\end{aligned}$$

thanks to (4.22). Since $I = J \cup I_1 \cup I_2$ and in view of (4.20), this implies that $\{C, C_i : i \in I\}$ has the strong CHIP at x_0. This completes the proof for the first part of Theorem 4.2.

For the proof of the second part, by (a*) there exist $\bar{x} \in X$ and two disjoint finite subsets I_1, I_2 of I such that

$$\bar{x} \in \operatorname{ri} C \bigcap \left(\operatorname{int} \bigcap_{i \in I_0} C_i \right) \bigcap \left(\bigcap_{i \in I_1} \operatorname{ri} C_i \right) \bigcap \left(\bigcap_{i \in I_2} C_i \right). \tag{4.25}$$

Now the proof is completed almost the same as for the first part, with the only modifications as follows. We use (4.25) in place of (4.11). We use "int" to replace "rint_C" in (4.15); in (4.13) and (4.14) we use "C_i" to replace "$(\operatorname{aff} C) \cap C_i$." □

Below we provide a sufficient condition ensuring the strong CHIP for a CCS-system with a closed subspace as a base-set.

LEMMA 4.1. *Let $\{C_i : i \in I\}$ be a family of nonempty closed convex subsets of X, $Y \subseteq X$ be a vector subspace, and $x_0 \in Y \bigcap (\bigcap_{i \in I} C_i)$. Then the system $\{Y, C_i : i \in I\}$ has the strong CHIP at x_0 provided that*

(a) *for each $i \in I$, either $C_i \subseteq Y$ or C_i is a polyhedron;*

(b) *$\{Y \cap C_i : i \in I\}$ has the strong CHIP at x_0 in Y.*

Proof. By (b) and recalling the notation of N^Y given in (2.17), we have

$$N^Y_{Y\cap(\cap_{i\in I} C_i)}(x_0) = \sum_{i\in I} N^Y_{Y\cap C_i}(x_0).$$

Now let $x^* \in N_{Y\cap(\cap_{i\in I} C_i)}(x_0)$. Then $x^*|_Y \in N^Y_{Y\cap(\cap_{i\in I} C_i)}(x_0)$ and, hence, there exist a finite subset I_0 of I and $\widetilde{x}^*_i \in N^Y_{Y\cap C_i}(x_0)$, $i \in I_0$, such that

$$x^*|_Y = \sum_{i\in I_0} \widetilde{x}^*_i \quad \text{on } Y. \tag{4.26}$$

For each $i \in I_0$, let $x^*_i \in X^*$ be an extension of $\widetilde{x}^*_i$. Then $x^*_i \in N_{Y\cap C_i}(x_0)$. Note that in the case when $C_i \subseteq Y$, $N_{Y\cap C_i}(x_0) = N_{C_i}(x_0)$, while in the case when C_i is a polyhedron, $N_{Y\cap C_i}(x_0) = N_Y(x_0) + N_{C_i}(x_0)$ by Theorem 2.2. Thus (a) implies that $\sum_{i\in I_0} x^*_i \in N_Y(x_0) + \sum_{i\in I} N_{C_i}(x_0)$. Denote $y^* := x^* - \sum_{i\in I_0} x^*_i$. Then $y^* \in N_Y(x_0)$ and so $x^* = y^* + \sum_{i\in I_0} x^*_i \in N_Y(x_0) + \sum_{i\in I} N_{C_i}(x_0)$. This shows that $N_{Y\cap(\cap_{i\in I} C_i)}(x_0) \subseteq N_Y(x_0) + \sum_{i\in I} N_{C_i}(x_0)$ and the proof is complete. □

The following consequence of Lemma 4.1 and Theorem 4.2 will be further extended in Corollary 4.4.

Corollary 4.1. *Let C, C_1 be a pair of closed convex subsets of X such that* $\operatorname{ri} C \cap \operatorname{ri} C_1 \neq \emptyset$. *Suppose that (at least) one of the sets is finite dimensional or finite codimensional. Then $\{C, C_1\}$ has the strong CHIP.*

Proof. By symmetry, we need only to consider two cases: (i) C_1 is finite codimensional; (ii) C is finite dimensional. The case (i) is clear by Theorem 4.2 (with $I = I_1 = \{1\}$ and $I_2 = \emptyset$). For the case (ii), let $x_0 \in C \cap C_1$. Let $\overline{\operatorname{span} C} + \overline{\operatorname{span} C_1}$ be denoted by Y. Then C_1 is finite codimensional in Y and hence, by what we just proved, the system $\{C, C_1\}$ in Y has the strong CHIP in Y. Since C and C_1 are subsets of Y, it follows from Lemma 4.1 that the system $\{Y, C, C_1\}$ has the strong CHIP. This implies that $\{C, C_1\}$ has the strong CHIP in X since $N_Y(x_0) \subseteq N_C(x_0)$ for each $x_0 \in C \cap C_1$. □

If assumption (d) in Theorem 4.2 is replaced by the stronger assumption that C is finite dimensional, then (e) can be dropped. This will be proved in Theorem 4.3 below. For preparing its proof and also for a later use, we first give a lemma.

Lemma 4.2. *Let $x_0 \in C \cap (\cap_{i\in I} C_i)$. The CCS-system $\{C, C_i : i \in I\}$ has the strong CHIP at x_0 if it satisfies* (a*) *and* (b*) *of Theorem* 4.2 *as well as the following conditions.*

($\bar{\text{c}}$) *$\{C, C_i : i \in I_1 \cup I_2\}$ has the strong CHIP at x_0.*

(d) *The same as* (d) *in Theorem* 4.2.

Proof. As in the beginning of the proof of Theorem 4.2, let I^l be defined by (4.12) and let $\vec{C}_i$, for each $i \in I^l$, be defined by (4.13) with X in place of C. Let $J := I_0 \cup I^l$. Define $\vec{C}_i = C_i$ for each $i \in I_0$. Then J is compact and $\{C, \vec{C}_i : i \in J\}$ is a CCS-system with the following properties:

(i) $\{C, \vec{C}_i : i \in J\}$ satisfies the strong interior-point condition; in fact,

$$\bar{x} \in \operatorname{ri} C \bigcap \left(\operatorname{int} \bigcap_{j\in J} \vec{C}_j \right). \tag{4.27}$$

(ii) The set-valued function $j \mapsto \vec{C}_j$ is lower semicontinuous on J.

(iii) $C_j \subseteq \vec{C}_j$ for each $j \in J$.

(iv) C is finite dimensional or $\vec{J}(x_0)$ is finite, where

$$\vec{J}(x_0) = \{i \in J : \ x_0 \in \operatorname{bd} \vec{C}_i\}.$$

Thanks to (i), (ii), and (iv), it is easy to see that the system $\{C, \vec{C}_i : \ i \in J\}$ satisfies the conditions (a*), (b*), and (d) of Theorem 4.1. Hence, by Theorems 2.2 and 4.1 and the above (iii), we obtain that

$$\begin{aligned} N(x_0; C) + N(x_0; (\cap_{i\in J} \vec{C}_i)) &= N(x_0; C \cap (\cap_{i\in J} \vec{C}_i)) \\ &= N(x_0; C) + \textstyle\sum_{i\in J} N(x_0; \vec{C}_i) \\ &\subseteq N(x_0; C) + \textstyle\sum_{i\in J} N(x_0; C_i). \end{aligned} \tag{4.28}$$

Noting by (iii) that

$$C \bigcap \left(\bigcap_{i\in I} C_i \right) = C \bigcap \left(\bigcap_{i\in J} \vec{C}_i \right) \bigcap \left(\bigcap_{i\in I_1 \cup I_2} C_i \right) \tag{4.29}$$

and that

$$\bar{x} \in \operatorname{ri} C \bigcap \left(\operatorname{int} \bigcap_{j\in J} \vec{C}_j \right) \bigcap \left(\bigcap_{i\in I_1 \cup I_2} C_i \right), \tag{4.30}$$

(a) of Theorem 2.2 can be applied to conclude that

$$\begin{aligned} N(x_0; C \cap (\cap_{i\in I} C_i)) &= N(x_0; \cap_{i\in J} \vec{C}_i) + N(x_0; C \cap (\cap_{i\in I_1 \cup I_2} C_i)) \\ &= N(x_0; C) + N(x_0; (\cap_{i\in J} \vec{C}_i)) \\ &\quad + \textstyle\sum_{i\in I_1} N(x_0; C_i) + \textstyle\sum_{i\in I_2} N(x_0; C_i), \end{aligned} \tag{4.31}$$

thanks to assumption ($\bar{\text{c}}$). Combining (4.28) and (4.31) gives the desired conclusion and the proof is complete. □

THEOREM 4.3. *Let $x_0 \in C \cap (\cap_{i\in I} C_i)$. The system $\{C, C_i : \ i \in I\}$ has the strong CHIP at x_0 if it satisfies* (a), (b), *and* (c) *of Theorem* 4.2 *and the following condition.*

($\bar{\text{d}}$) *C is finite dimensional.*

Moreover, the same conclusion also holds if (a) + (b) + (c) + ($\bar{\text{d}}$) *in the above assumptions is replaced by* (a*) + (b*) + ($\bar{\text{d}}$), *where* (a*) *and* (b*) *are as in Theorem* 4.2.

Proof. Denote

$$\widehat{C} = C - x_0, \quad Z := \operatorname{span} \widehat{C}, \quad \widehat{C}_i = C_i - x_0 \quad \text{for each } i \in I. \tag{4.32}$$

Then, by assumptions, Z is finite dimensional and

$$\operatorname{ri} \widehat{C} \bigcap \left(\operatorname{rint}_{\widehat{C}} \bigcap_{i\in I_0} \widehat{C}_i \right) \bigcap \left(\bigcap_{i\in I_1} \operatorname{ri} \widehat{C}_i \right) \bigcap \left(\bigcap_{i\in I_2} \widehat{C}_i \right) \neq \emptyset. \tag{4.33}$$

Letting $C_i^\sharp$ denote the intersection $Z \cap \widehat{C}_i$, it follows that

$$\operatorname{ri} \widehat{C} \bigcap \left(\operatorname{rint}_{\widehat{C}} \bigcap_{i\in I_0} C_i^\sharp \right) \bigcap \left(\bigcap_{i\in I_1} \operatorname{ri} C_i^\sharp \right) \bigcap \left(\bigcap_{i\in I_2} C_i^\sharp \right) \neq \emptyset. \tag{4.34}$$

Noting aff $\widehat{C} = Z$, this implies that, as a system in the Banach space Z, $\{\widehat{C}, C_i^\sharp :\ i \in I\}$ satisfies the weak-strong interior-point condition. In fact, with respect to the relative topology in Z, one has

$$\operatorname{rint}_Z \widehat{C} \bigcap \left(\operatorname{rint}_Z \bigcap_{i \in I_0} C_i^\sharp \right) \bigcap \left(\bigcap_{i \in I_1} \operatorname{ri} C_i^\sharp \right) \bigcap \left(\bigcap_{i \in I_2} C_i^\sharp \right) \neq \emptyset. \tag{4.35}$$

By Theorem DLW (applied to the system $\{\widehat{C}, \bigcap_{i \in I} C_i^\sharp\}$ in Z), we have

$$N^Z_{\widehat{C} \cap (\cap_{i \in I} C_i^\sharp)}(0) = N^Z_{\widehat{C}}(0) + N^Z_{\cap_{i \in I} C_i^\sharp}(0);$$

namely, in the normed linear space X,

$$N_{\widehat{C} \cap (\cap_{i \in I} \widehat{C}_i)}(0) = N_{\widehat{C}}(0) + N_{\cap_{i \in I} C_i^\sharp}(0), \tag{4.36}$$

because $\widehat{C} \cap (\cap_{i \in I} C_i^\sharp) = \widehat{C} \cap (\cap_{i \in I} \widehat{C}_i)$, and $\widehat{C}$ as well as $C_i^\sharp$ are subsets of Z. We will show that

$$N_{\cap_{i \in I} C_i^\sharp}(0) \subseteq N_{\widehat{C}}(0) + \sum_{i \in I} N_{\widehat{C}_i}(0). \tag{4.37}$$

Granting this, (4.36) and an easy translation argument imply that

$$N_{C \cap (\cap_{i \in I} C_i)}(x_0) \subseteq N_C(x_0) + \sum_{i \in I} N_{C_i}(x_0),$$

which shows that $\{C, C_i :\ i \in I\}$ has the strong CHIP at x_0. It remains to prove (4.37). To do this, we shall apply Lemma 4.1 to $Y := Z$ and the system $\{D_i\}$, where

$$D_i = \begin{cases} C_i^\sharp = \widehat{C}_i \cap Z, & i \in I \setminus I_2, \\ \widehat{C}_i, & i \in I_2. \end{cases}$$

We suppose, without loss of generality, that $I \setminus I_2 \neq \emptyset$ (otherwise (4.37) holds by Theorem 2.2). Then $\cap_{i \in I} C_i^\sharp = \cap_{i \in I} D_i$. Moreover, assumption (c) (for $i \in I \setminus (I_1 \cup I_2)$) and Corollary 4.1 (for $i \in I_1$) imply that for each $i \in I \setminus I_2$, $\{Z, \widehat{C}_i\}$ has the strong CHIP at 0 and hence that

$$N_{D_i}(0) = N_{Z \cap \widehat{C}_i}(0) = N_Z(0) + N_{\widehat{C}_i}(0) \subseteq N_{\widehat{C}}(0) + N_{\widehat{C}_i}(0). \tag{4.38}$$

We claim that $\{D_i :\ i \in I\}$ has the strong CHIP at 0. Granting this, it follows from (4.38) that

$$\begin{aligned} N_{\cap_{i \in I} C_i^\sharp}(0) = N_{\cap_{i \in I} D_i}(0) &= \sum_{i \in I} N_{D_i}(0) \\ &= \sum_{i \in I \setminus I_2} N_{D_i}(0) + \sum_{i \in I_2} N_{\widehat{C}_i}(0) \subseteq N_{\widehat{C}} + \sum_{i \in I} N_{\widehat{C}_i}(0), \end{aligned}$$

which shows (4.37). Thus it remains to prove the above claim. In view of Lemma 4.1, it suffices to show that, as a system in the subspace Z, $\{Z \cap D_i :\ i \in I\}$ has the strong CHIP at 0 (note that for each $i \in I_2$, $\widehat{C}_i$ is a polyhedron because C_i is a polyhedron). By (4.35), this system in Z satisfies the weak-strong interior-point condition on Z

with (I_1, I_2). Assumption (b) tells us that $i \mapsto D_i \cap Z$ is lower semicontinuous on I. It is trivial that for each $i \in I_1$, $D_i \cap Z$ is finite codimensional in Z as Z is finite dimensional. Hence it is easy to see that the second part of Theorem 4.2 is applicable to the system $\{Z, Z \cap D_i : i \in I\}$ in place of $\{C, C_i : i \in I\}$ if assumptions (a), (b), (c), and ($\bar{\text{d}}$) are assumed. Thus, this system (in Z) does have the strong CHIP at 0. Our claim is therefore established and this completes the proof of the first part of the theorem.

For the second part, suppose that the system $\{C, C_i : i \in I\}$ satisfies (a*) + (b*) + ($\bar{\text{d}}$) (in place of (a) + (b) + (c) + ($\bar{\text{d}}$)). Then, by ($\bar{\text{d}}$) and by the result of the first part (applied to the finite subsystem $\{C, C_i : i \in I_1 \cup I_2\}$), one concludes that $\{C, C_i : i \in I_1 \cup I_2\}$ has the strong CHIP at x_0. Hence, by Lemma 4.2, $\{C, C_i : i \in I\}$ also has the strong CHIP at x_0. □

Corollary 4.2. *Let $x_0 \in C \cap (\cap_{i \in I} C_i)$. The system $\{C, C_i : i \in I\}$ has the strong CHIP at x_0 if it satisfies* (a*) *and* (b*) *of Theorem* 4.2 *as well as the following condition.*

(d*) *At least one of $\{C, C_i : i \in I_1\}$ is finite dimensional.*

Proof. By assumption (a*), take $\bar{x}$ satisfying (4.25). By assumption (d*) and in view of the second part of Theorem 4.3, it suffices to consider the case when C_{i_0} is finite dimensional for some $i_0 \in I_1$. Let

$$I_1' = I_1 \cup \{i_\infty\},$$

where i_∞ is a new index such that $i_\infty \notin I$. Let $I_0 = I \setminus (I_1 \cup I_2)$ and define $D := C_{i_0} \cap (\cap_{i \in I_0} C_i)$. Then $\operatorname{ri} C_{i_0} \cap \operatorname{int}(\cap_{i \in I_0} C_i) \subseteq \operatorname{ri} D$, and thus $\bar{x} \in \operatorname{ri} D$ by (4.25). Thus

$$\bar{x} \in \operatorname{ri} D \bigcap \left(\bigcap_{i \in I_1} \operatorname{ri} C_i \right) \bigcap \operatorname{ri} C \bigcap \left(\bigcap_{i \in I_2} C_i \right). \tag{4.39}$$

Letting $J = I_1' \cup I_2$, $D_{i_\infty} = C$, and $D_i = C_i$ for each $i \in I_1 \cup I_2$, (4.39) implies that the new CCS-system $\{D, D_j : j \in J\}$ satisfies the weak-strong interior-point condition with (I_1', I_2), that is, the condition (a*) of Theorem 4.3 stated for $\{C, C_i : i \in I\}$. It also satisfies (b*) of Theorem 4.3 as J is finite. Therefore by applying Theorem 4.3 to the new system we have that

$$\begin{aligned} N_{D \cap (\cap_{j \in J} D_i)}(x_0) &= N_D(x_0) + \textstyle\sum_{j \in J} N_{D_j}(x_0) \\ &= N_{C_{i_0} \cap (\cap_{i \in I_0} C_i)}(x_0) + N_C(x_0) + \textstyle\sum_{i \in I_1 \cup I_2} N_{C_i}(x_0) \\ &\subseteq N_C(x_0) + N_{C_{i_0} \cap (\cap_{i \in I} C_i)}(x_0). \end{aligned} \tag{4.40}$$

Applying Theorem 4.3 to the system $\{C_{i_0}, C_i : i \in I\}$ and noting that $D \cap (\cap_{j \in J} D_i) = C \cap (\cap_{i \in I} C_i)$, it follows from (4.40) that

$$N_{C \cap (\cap_{i \in I} C_i)}(x_0) = N_{D \cap (\cap_{j \in J} D_i)}(x_0) = N_C(x_0) + \sum_{i \in I} N_{C_i}(x_0).$$

The proof is complete. □

Corollary 4.3. *Let $x_0 \in C \cap (\cap_{i \in I} C_i)$. The system $\{C, C_i : i \in I\}$ has the strong CHIP at x_0 if it satisfies* (a*) *and* (b*) *of Theorem* 4.2 *as well as the following conditions.*

(c*) *For any $i, j \in I_1$, C_i is finite codimensional in C as well as in C_j.*

($\widetilde{\text{d}}$) *$I(x_0) = \{i \in I : x_0 \in \operatorname{bd} C_i\}$ is finite.*

Proof. By Lemma 4.2, it suffices to show that ($\bar{\text{c}}$) of Lemma 4.2 holds; that is, the system $\{C, C_i : i \in I_1 \cup I_2\}$ has the strong CHIP at x_0. For this purpose, take a new index $0 \notin I$ and, for each $i \in I_1$, let Y_0 and Y_i denote the closed subspaces spanned by C and C_i, respectively, and let

$$Y = Y_0 + \sum_{i \in I_1} Y_i. \tag{4.41}$$

We claim that each C_i for $i \in I_1$ is finite codimensional in Y. To verify this claim, let $i \in I_1$ and set $J = \{0\} \cup I_1 \setminus \{i\}$. By assumption (c*), C_i is finite codimensional in Y_j, for each $j \in J$, and it follows from Definition 4.1 that there exists a finite dimensional subspace Y_j' of Y_j such that Y_j is the direct sum of Y_j' and $Y_j \cap Y_i$, that is,

$$Y_j = Y_j' + (Y_j \cap Y_i) \quad \text{and} \quad Y_j' \cap (Y_j \cap Y_i) = \{0\}.$$

Hence, for each $j \in J$,

$$Y_j + Y_i = Y_j' + Y_i \quad \text{and} \quad Y_j' \cap Y_i = \{0\}. \tag{4.42}$$

Then, by (4.41) and (4.42),

$$Y = \sum_{j \in J}(Y_j + Y_i) = \sum_{j \in J} Y_j' + Y_i \quad \text{and} \quad \left(\sum_{j \in J} Y_j'\right) \cap Y_i = \{0\}. \tag{4.43}$$

This implies that Y_i is finite codimensional in Y since $\sum_{j \in J} Y_j'$ is finite dimensional. The claim is proved. Noting that $C_i \cap Y = C_i$ for each $i \in I_1$, this implies that, as a CCS-system in Y, $\{C, C_i \cap Y : i \in I_1 \cup I_2\}$ has property (e) of Theorem 4.2 stated for $\{C, C_i : i \in I\}$. Moreover, it also satisfies (d) thanks to assumption ($\widetilde{\text{d}}$) and (4.1). Therefore one can apply the second part of Theorem 4.2 (with $I_0 = \emptyset$) to conclude that this finite system in the subspace Y has the strong CHIP at x_0. Noting that $C, C_i \subseteq Y$ for each $i \in I_1$ and C_j is a polyhedron for each $j \in I_2$, it follows from Lemma 4.1 that $\{Y, C, C_i : i \in I_1 \cup I_2\}$ has the strong CHIP at x_0 in X. Therefore $\{C, C_i : i \in I_1 \cup I_2\}$ has the strong CHIP at x_0 (because $C \subseteq Y$). □

We obtain below an extension of Rockafellar's result [27, Corollary 23.8.1, p. 223] in the setting of general normed linear spaces.

COROLLARY 4.4. *Let $I = J \cup K$ be finite with J, K disjoint such that C_k is a polyhedron for each $k \in K$, and suppose that*

$$\operatorname{ri} C \bigcap \left(\bigcap_{j \in J} \operatorname{ri} C_j\right) \bigcap \left(\bigcap_{k \in K} C_i\right) \neq \emptyset. \tag{4.44}$$

Then the system $\{C, C_i : i \in I\}$ has the strong CHIP if at least one of the following conditions is satisfied.

(a) *At least one of $\{C, C_j : j \in J\}$ is finite dimensional.*

(b) *For each $j \in J$, C_j is finite codimensional in C and C_i, respectively, for each $i \in J$ (e.g., C_j is finite codimensional for each $j \in J$).*

Proof. Since I is finite and thanks to (4.44), the system $\{C, C_i : i \in I\}$ satisfies (a*) and (b*) of Theorem 4.2 (with $(I_1, I_2) = (J, K)$). Now apply Corollary 4.2 and Corollary 4.3 to conclude the proof. □

5. Subsystems and the strong CHIP. Recall that I and $\{C, C_i : i \in I\}$ are as explained in the beginning of the preceding section. Our main result of this section is the following.

THEOREM 5.1. *Suppose that the finite subsystem $\{C, C_i : i \in J\}$ has the strong CHIP for each finite subset J of I. Then the system $\{C, C_i : i \in I\}$ has the strong CHIP provided the following conditions are satisfied.*

(a) *C is finite dimensional (say, $\dim C := l < +\infty$).*

(b) *The set-valued function $i \mapsto (\operatorname{aff} C) \cap C_i$ is Kuratowski continuous on I.*

(c) *For any finite subset J of I (with the number of elements $|J| \leq l$ if X is real and $|J| \leq 2l$ if X is complex), the subsystem $\{C, C_i : i \in J\}$ satisfies the following C-interior-point condition:*

$$C \bigcap \left(\bigcap_{i \in J} \operatorname{rint}_C C_i \right) \neq \emptyset. \tag{5.1}$$

Proof. Since I is compact and metrizable, there exists a sequence $\{I_k\}$ of subsets of I such that

(i) each I_k is finite;

(ii) $I_k \subseteq I_{k+1}$ for $k = 1, 2, \dots$;

(iii) I equals the closure of $\bigcup_{k=1}^{\infty} I_k$.

Let $S = \bigcap_{i \in I} C_i$, $K = C \cap S$, and $S_k = \bigcap_{i \in I_k} C_i$ for each k. Then

$$K = C \bigcap \left(\bigcap_{i \in I} C_i \right) \subseteq C \cap S_k. \tag{5.2}$$

Let $x_0 \in C \bigcap (\bigcap_{i \in I} C_i)$ and let $x^* \in N_K(x_0)$. We have to show that $x^* \in N_C(x_0) + \sum_{i \in I} N_{C_i}(x_0)$. We will first show that there exist $\{x_k\} \subseteq C$ with $x_k \to x_0$ and $x_k^* \in N_{C \cap S_k}(x_k)$ such that $\{x_k^*\}$ is bounded and

$$\lim_{k \to \infty} \langle x_k^*, y \rangle = \langle x^*, y \rangle \quad \text{for each } y \in \operatorname{span}(C - x_0). \tag{5.3}$$

In fact, since $Z := \operatorname{span}(C - x_0)$ is finite dimensional, we may assume, without loss of generality, that the norm restricted to Z is both strictly convex and smooth. Clearly, we may assume that $x^*|_Z \neq 0$. Take $z_0 \in Z$ such that $\langle x^*, z_0 \rangle = \|x^*|_Z\| \cdot \|z_0\| = \|z_0\|^2$. Write $x := x_0 + z_0$. Then, $x^*|_Z = J(x - x_0)|_Z$. By the Hahn–Banach theorem, let $\bar{x}^* \in X^*$ be a norm-preserving extension of $x^*|_Z$. Then $\bar{x}^* \in N_K(x_0) \cap J(x - x_0)$. Hence by Proposition 2.1, $x_0 = P_K(x)$. Let $x_k = P_{C \cap S_k}(x)$. Then $x - x_k \in Z$ because $x_0 - x_k$, $x - x_0 \in Z$. Moreover,

$$\|x_k\| \leq \|x_k - x\| + \|x\| \leq \|x - x_0\| + \|x\|;$$

hence $\{x_k\} \subset C$ is bounded. Without loss of generality, assume that $x_k \to \bar{x}$ for some $\bar{x} \in C$. Let $i \in I$. By (ii) and (iii), there exists $\{i_k\} \subseteq I$ with $i_k \in I_k$ for each k such that $i_k \to i$. Noting that $x_k \in (\operatorname{aff} C) \cap C_{i_k}$ and that $x_k \to \bar{x}$, we have that $\bar{x} \in (\operatorname{aff} C) \cap C_i$ by the upper Kuratowski semicontinuity assumed in (b). This shows that $\bar{x} \in K$. Because

$$\|x - \bar{x}\| = \lim_{k \to \infty} \|x - x_k\| \leq \|x - y\| \quad \text{for each } y \in K,$$

$\bar{x} = P_K(x) = x_0$ and hence $x_k \to x_0$. On the other hand, by Proposition 2.1, there exists $x_k^* \in N_{C \cap S_k}(x_k) \cap J(x - x_k)$. Consequently, $\{x_k^*\}$ is bounded since

$\|x_k^*\| = \|x - x_k\|$. Moreover, by the smoothness of the norm in Z, the mapping $z \mapsto J(z)|_Z$ is norm-weak* continuous. Hence $x_k^*|_Z \to x^*|_Z$ as $x - x_k \to x - x_0$ in Z. Thus (5.3) holds.

By assumption, the finite subsystem $\{C, C_i : i \in I_k\}$ has the strong CHIP at x_k, and hence

$$x_k^* \in N_{C\cap S_k}(x_k) = N_C(x_k) + \sum_{i\in I_k} N_{C_i}(x_k). \tag{5.4}$$

If there exists a subsequence $\{k_j\}$ of $\{k\}$ such that $x_{k_j}^* \in N_C(x_{k_j})$ for each j, then $x^* \in N_C(x_0)$ by (5.3) (and so $x^* \in N_C(x_0) + \sum_{i\in I} N_{C_i}(x_0)$). Therefore we may assume that for each k, $x_k^* \notin N_C(x_k)$. Recalling that the dimension of Z is l, it follows from (5.4) and [27, Corollary 17.1.2] that there exist $z_k^* \in N_C(x_0)$, $i_j^k \in I_k$, and $\widehat{y}_{i_j^k}^* \in N_{C_{i_j^k}}(x_0)$ such that

$$x_k^* = z_k^* + \sum_{j=1}^{s} \widehat{y}_{i_j^k}^* \quad \text{on } Z \quad \text{for all } k = 1, 2, \ldots, \tag{5.5}$$

where $s \le l$ if X is real and $s \le 2l$ if X is complex. Without loss of generality, assume that $\widehat{y}_{i_j^k}^*|_Z \ne 0$ for each $j = 1, 2, \ldots, s$. Set $\lambda_j^k = \|\widehat{y}_{i_j^k}^*|_Z\|$, $y_{i_j^k}^* = \frac{\widehat{y}_{i_j^k}^*}{\lambda_j^k}$. Then $\lambda_j^k > 0$ for $j = 1, 2, \ldots, s$ and

$$x_k^* = z_k^* + \sum_{j=1}^{s} \lambda_j^k y_{i_j^k}^* \quad \text{on } Z \quad \text{for all } k = 1, 2, \ldots. \tag{5.6}$$

Let $\lambda^k := \sum_{j=1}^{s} \lambda_j^k$. Then $\{\lambda^k\}$ is bounded. Indeed, if not, by considering a subsequence if necessary, we have that $\lim_{k\to\infty} \lambda^k = +\infty$. Thus $\frac{x_k^*}{\lambda^k} \to 0$ as $k \to \infty$. Furthermore, without loss of generality, we may assume that as $k \to \infty$,

$$i_j^k \to i_j \quad \text{and} \quad \frac{\lambda_j^k}{\lambda^k} \to \mu_j, \quad j = 1, 2, \ldots, s. \tag{5.7}$$

Then $\sum_{j=1}^{s} \mu_j = 1$. Since $\{y_{i_j^k}^*\}$ is bounded, by (5.6), $\{\frac{z_k^*}{\lambda^k}|_Z\}$ is bounded too. Thus, we may also assume that there exist $\widetilde{z}_0^*, \widetilde{y}_{i_j}^* \in Z^*$ such that

$$\frac{z_k^*}{\lambda^k} \to \widetilde{z}_0^* \quad \text{and} \quad y_{i_j^k}^* \to \widetilde{y}_{i_j}^* \quad \text{on } Z \tag{5.8}$$

as $k \to \infty$. Then, since $z_k^* \in N_C(x_k)$, by (5.8) and the fact that $x_k \to x_0$,

$$\langle \widetilde{z}_0^*, z - x_0 \rangle \le 0 \quad \text{for each } z \in C. \tag{5.9}$$

Let $z \in (Z + x_0) \cap C_{i_j}$. Since $i_j^k \to i_j$ and thanks to assumption (b), there exists $\{z_k\}$ with each $z_k \in (\operatorname{aff} C) \cap C_{i_j^k}$ such that $z_k \to z$ as $k \to \infty$. Then $\langle y_{i_j^k}^*, z_k - x_k \rangle \le 0$. Since $x_k \to x_0$, it follows from (5.8) that for each j,

$$\langle \widetilde{y}_{i_j}^*, z - x_0 \rangle \le 0 \quad \text{for each } z \in (Z + x_0) \cap C_{i_j}. \tag{5.10}$$

Consequently, by the Hahn–Banach theorem, $\widetilde{z}_0^*$ and $\widetilde{y}_{i_j}^*$ can be extended to $z_0^* \in N_C(x_0)$ and $y_{i_j}^* \in N_{C_{i_j} \cap (Z+x_0)}(x_0)$. Clearly, by (5.6), (5.7), and (5.8),

$$0 = z_0^* + \sum_{j=1}^{s} \mu_j y_{i_j}^* \quad \text{on } Z. \tag{5.11}$$

By assumption (c), there exists $\bar{y} \in C \cap \operatorname{rint}_C C_{i_j}$ for each $j = 1, 2, \ldots, s$. Then, by (5.10), $\langle y_{i_j}^*, \bar{y} - x_0 \rangle \leq 0$ for each $j = 1, 2, \ldots, s$. We claim that each of the above inequalities must be strict. Indeed, suppose otherwise that $\langle y_{i_j}^*, \bar{y} - x_0 \rangle = 0$ for some j. Let $z \in Z$ and let $z_t := t(z + x_0) + (1-t)\bar{y}$. Then for any t with $|t|$ small enough, $z_t \in (\operatorname{aff} C) \cap C_{i_j}$ and thus, by (5.10),

$$t\langle y_{i_j}^*, z \rangle = \langle y_{i_j}^*, z_t - x_0 \rangle - (1-t)\langle y_{i_j}^*, \bar{y} - x_0 \rangle = \langle y_{i_j}^*, z_t - x_0 \rangle \leq 0.$$

This implies that $\langle y_{i_j}^*, z \rangle = 0$, that is, $y_{i_j}^*|_Z = 0$, which contradicts that $\|y_{i_j}^*|_Z\| = \|\widetilde{y}_{i_j}^*\| = 1$. Hence,

$$\left\langle z_0^* + \sum_{j=1}^{s} \mu_j y_{i_j}^*, \bar{y} - x_0 \right\rangle < 0,$$

which contradicts (5.11). This shows that $\{\lambda^k\}$ is bounded. Note that $N_{(Z+x_0)\cap C_{i_j}}(x_0) \subseteq N_C(x_0) + N_{C_{i_j}}(x_0)$. Thus, taking the limits on the two sides of (5.6) and using the similar arguments as above (if necessary, using subsequences), we get that

$$x^* = z_0^* + \sum_{j=1}^{s} \lambda_j y_{i_j}^* \quad \text{on } Z \tag{5.12}$$

for some $\lambda_j \geq 0$, $z_0^* \in N_C(x_0)$, and $y_{i_j}^* \in N_{C_{i_j}}(x_0)$ $(j = 1, 2, \ldots, s)$. Let $y^* = x^* - z_0^* - \sum_{j=1}^{s} \lambda_j y_{i_j}^*$. Then $y^* \in N_C(x_0)$ by (5.12) and thus $x^* \in N_C(x_0) + \sum_{i \in I} N_{C_i}(x_0)$. The proof is complete. □

Corollary 5.1. *Suppose that the CCS-system $\{C, C_i : i \in I\}$ satisfies the interior-point condition, $\dim C < +\infty$, and the set-valued function $i \mapsto (\operatorname{aff} C) \cap C_i$ is Kuratowski continuous. Then the system $\{C, C_i : i \in I\}$ has the strong CHIP.*

Proof. The assumed interior-point condition clearly implies (c) of Theorem 5.1; it also implies that each of the finite subsystems of $\{C, C_i : i \in I\}$ has the strong CHIP by Theorem DLW. Hence the conclusion holds by Theorem 5.1. □

Remark 5.1. Examples 5.1, 5.2, and 5.3 will show that none of the conditions (a), (b), and (c) in Theorem 5.1 can be dropped. Each of these examples will be a CCS-system without the strong CHIP, but each of the finite subsystems of each of these CCS-systems does have the strong CHIP (each C_i being a polyhedron, and the base-set being the whole space). In each of these examples, I is the compact subset of $\mathbb{R}$ defined by $I = \{0, 1, \frac{1}{2}, \ldots, \frac{1}{i}, \ldots\}$.

Example 5.1. Let $C = X = \{x = (x_1, x_2, \ldots, x_k, \ldots) : x_k \in \mathbb{R}, \lim_k x_k \text{ exists}\}$ with the norm defined by

$$\|x\| = \sup_k |x_k|, \quad x = (x_k) \in X.$$

Define

$$C_i = \begin{cases} \{x = (x_k) \in X : \lim_k x_k \leq 0\}, & i = 0, \\ \{x = (x_k) \in X : x_{\frac{1}{i}} \leq 0\}, & i \in I \setminus \{0\}. \end{cases}$$

Then the set-valued function $i \mapsto C_i$ is Kuratowski continuous on I. In fact, let $\{i_n\} \subseteq I$ be a sequence satisfying $i_n \to 0$. To show the Kuratowski continuity at 0, let $\{x^{i_n}\}$ be a sequence satisfying $x^{i_n} \in C_{i_n}$ and $x^{i_n} \to x^0$. Noting

$$x^0_{\frac{1}{i_n}} = x^0_{\frac{1}{i_n}} - x^{i_n}_{\frac{1}{i_n}} + x^{i_n}_{\frac{1}{i_n}} \le x^0_{\frac{1}{i_n}} - x^{i_n}_{\frac{1}{i_n}} \le \|x^0 - x^{i_n}\| \to 0,$$

one has that $\lim_k x^0_k \le 0$ and thus $x^0 \in C_0$. This proves that the set-valued function $i \mapsto C_i$ is upper Kuratowski continuous at 0. To show the lower Kuratowski continuity at 0, let $x^0 \in C_0$. Then $a = \lim_k x^0_k \le 0$. Define $x^{i_n} = (x^{i_n}_k)$ with $x^{i_n}_k = x^0_k$ if $k \ne \frac{1}{i_n}$ and $x^{i_n}_k = a$ if $k = \frac{1}{i_n}$. Then $x^{i_n} \in C_{i_n}$ and $\lim_n x^{i_n} = x^0$. This shows that the set-valued function $i \mapsto C_i$ is lower Kuratowski continuous at 0. Note that $\bar{x} \in \operatorname{int}(\bigcap_{i\in I} C_i)$ for $\bar{x} = (-1, -1, \dots) \in X$. Hence the conditions (b) and (c) in Theorem 5.1 are satisfied. Let $x_0 = 0$. Then $x_0 \in \bigcap_{i\in I} C_i$. It is easy to see that $\sum_{i\in I} N_{C_i}(x_0)$ is not closed; hence this system does not have the strong CHIP at x_0. □

Example 5.2. Let $C = X = \mathbb{R}^2$. Define

$$C_i = \begin{cases} \{x = (x_1, x_2) \in X : \ x_1 + x_2 \le 0\}, & i = 0, \\ \{x = (x_1, x_2) \in X : \ x_1 + i x_2 \le 0\}, & i \in I \setminus \{0\}. \end{cases}$$

Then $\bigcap_{i\in I} C_i = \{(x_1, x_2) : \ x_1 \le 0, \, x_1 + x_2 \le 0\}$. Let $x_0 = 0$. Then $x_0 \in \operatorname{bd} \bigcap_{i\in I} C_i$. Clearly, (a) in Theorem 5.1 is satisfied. Since $\bar{x} = (-1, \frac{1}{2}) \in \operatorname{int}(\bigcap_{i\in I} C_i)$, condition (c) in Theorem 5.1 is satisfied too. However, $N_{\bigcap_{i\in I} C_i}(x_0) = \{(t_1, t_2) : \ 0 \le t_2 \le t_1\}$ and $\sum_{i\in I} N_{C_i}(x_0) = \{(t_1, t_2) : \ 0 < t_2 \le t_1\} \cup \{(0, 0)\}$. Therefore this system does not have the strong CHIP at x_0. Note that condition (b) is not satisfied. □

Example 5.3. Let $C = X = \mathbb{R}^2$ and define

$$C_i = \begin{cases} \{x = (x_1, x_2) : \ x_2 \le 0\}, & i = 1, \\ \{x = (x_1, x_2) : \ i x_1 - x_2 - i^2 \le 0\}, & i \in I \setminus \{1\}. \end{cases}$$

Then $\bigcap_{i\in I} C_i = \{(x_1, 0) \in \mathbb{R}^2 : \ x_1 \le 0\}$. Let $x_0 = 0$. Hence

$$N_{\bigcap_{i\in I} C_i}(x_0) = \{(t_1, t_2) \in \mathbb{R}^2 : \ t_1 \ge 0\}$$

and

$$\sum_{i\in I} N_{C_i}(x_0) = \operatorname{cone}\{(0, -1), \, (0, 1)\} = \{(t_1, t_2) \in \mathbb{R}^2 : \ t_1 = 0\}.$$

Consequently, this system does not have the strong CHIP at x_0. Note that conditions (a) and (b) in Theorem 5.1 are satisfied but condition (c) is not. □

Acknowledgment. We thank the referee for his helpful comments.

REFERENCES

[1] A. Baken, F. Deutsch, and W. Li, *Strong CHIP, normality, and linear regularity of convex sets*, Trans. Amer. Math. Soc., 357 (2005), pp. 3831–3863.

[2] V. Barbu and T. Precupanu, *Convexity and Optimization in Banach Spaces*, Sijthoff & Noordhoff, Groningen, The Netherlands, 1978.

[3] H. Bauschke, J. Borwein, and W. Li, *Strong conical hull intersection property, bounded linear regularity, Jameson's property (G), and error bounds in convex optimization*, Math. Program., Ser. A, 86 (1999), pp. 135–160.

[4] D. Braess, *Nonlinear Approximation Theory*, Springer-Verlag, New York, 1986.

[5] C. Chui, F. Deutsch, and J. Ward, *Constrained best approximation in Hilbert space*, Constr. Approx., 6 (1990), pp. 35–64.

[6] C. Chui, F. Deutsch, and J. Ward, *Constrained best approximation in Hilbert space* II, J. Approx. Theory, 71 (1992), pp. 231–238.

[7] F. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley and Sons, New York, 1983.

[8] F. Deutsch, *The role of the strong conical hull intersection property in convex optimization and approximation*, in Approximation Theory IX, Vol. I: Theoretical Aspects, C. Chui and L. Schumaker, eds., Vanderbilt University Press, Nashville, TN, 1998, pp. 105–112.

[9] F. Deutsch, *Some Applications of Functional Analysis to Approximation Theory*, doctoral dissertation, Brown University, Providence, RI, 1965.

[10] F. Deutsch, *Best Approximation in Inner Product Spaces*, Springer-Verlag, New York, 2001.

[11] F. Deutsch, W. Li, and J. Swetits, *Fenchel duality and the strong conical hull intersection property*, J. Optim. Theory Appl., 102 (1997), pp. 681–695.

[12] F. Deutsch, W. Li, and J. Ward, *A dual approach to constrained interpolation from a convex subset of Hilbert space*, J. Approx. Theory, 90 (1997), pp. 385–414.

[13] F. Deutsch, W. Li, and J. D. Ward, *Best approximation from the intersection of a closed convex set and a polyhedron in Hilbert space, weak Slater conditions, and the strong conical hull intersection property*, SIAM J. Optim., 10 (1999), pp. 252–268.

[14] F. Deutsch, V. Ubhaya, J. Ward, and Y. Xu, *Constrained best approximation in Hilbert space* III: *Application to n-convex functions*, Constr. Approx., 12 (1996), pp. 361–384.

[15] M. A. Goberna and M. A. Lopez, *Linear Semi-infinite Optimization*, John Wiley and Sons, Chichester, 1998.

[16] J. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms* I, Grundlehren Math. Wiss. 305, Springer-Verlag, Berlin, 1993.

[17] C. Li, *On best uniform restricted range approximation in complex-valued continuous function spaces*, J. Approx. Theory, 120 (2003), pp. 71–84.

[18] C. Li and X.-Q. Jin, *Nonlinearly constrained best approximation in Hilbert spaces: The strong CHIP and the basic constraint qualification*, SIAM J. Optim., 13 (2002), pp. 228–239.

[19] C. Li and K. F. Ng, *On best approximation by nonconvex sets and perturbation of nonconvex inequality systems in Hilbert spaces*, SIAM J. Optim., 13 (2002), pp. 726–744.

[20] C. Li and K. F. Ng, *Constraint qualification, the strong CHIP, and best approximation with convex constraints in Banach spaces*, SIAM J. Optim., 14 (2003), pp. 584–607.

[21] C. Li and K. F. Ng, *On constraint qualification for an infinite system of convex inequalities in a Banach space*, SIAM J. Optim., 15 (2005), pp. 488–512.

[22] C. Li and K. F. Ng, *On best restricted range approximation in continuous complex-valued function spaces*, J. Approx. Theory, in press.

[23] W. Li, C. Nahak, and I. Singer, *Constraint qualifications for semi-infinite systems of convex inequalities*, SIAM J. Optim., 11 (2000), pp. 31–52.

[24] C. Micchelli, P. Smith, J. Swetits, and J. Ward, *Constrained L_p-approximation*, Constr. Approx., 1 (1985), pp. 93–102.

[25] C. A. Micchelli and F. I. Utreras, *Smoothing and interpolation in a convex subset of a Hilbert space*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 728–746.

[26] K. F. Ng and W. Song, *Fenchel duality in infinite-dimensional setting and its applications*, Nonlinear Anal., 55 (2003), pp. 845–858.

[27] R. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[28] G. Sh. Rubenstein, *On an extremal problem in a normed linear space*, Sibirsk. Mat. Zh., 6 (1965), pp. 711–714 (in Russian).

[29] I. Singer, *The Theory of Best Approximation and Functional Analysis*, CBMS-NSF Regional Conf. Ser. in Appl. Math. 13, SIAM, Philadelphia, 1974.

[30] I. Singer, *Duality for optimization and best approximation over finite intersection*, Numer. Funct. Anal. Optim., 19 (1998), pp. 903–915.

[31] G. S. Smirnov and R. G. Smirnov, *Best uniform restricted range approximation of complex-valued functions*, C. R. Math. Rep. Acad. Sci. Canada, 19 (1997), pp. 58–63.

[32] G. S. Smirnov and R. G. Smirnov, *Best uniform approximation of complex-valued functions by generalized polynomials having restricted range*, J. Approx. Theory, 100 (1999), pp. 284–303.

[33] G. S. Smirnov and R. G. Smirnov, *Kolmogorov-type theory of best restricted approximation*, East J. Approx., 6 (2000), pp. 309–329.

[34] G. S. Smirnov and R. G. Smirnov, *Best restricted approximation of complex-valued functions* II, C. R. Acad. Sci. Paris Sér. I Math., 330 (2000), pp. 1059–1064.

[35] G. S. Smirnov and R. G. Smirnov, *Theory of best restricted ranges approximation revisited: A characterization theorem*, in Approximation Theory and Its Applications, Pr. Inst. Mat. Nats. Akad. Nauk Ukr. Mat. Zastos. 31, Institute of Mathematics, Kiev, Ukraine, 2000, pp. 436–445.

SIAM J. OPTIM.
Vol. 16, No. 2, pp. 341–357

A FILTER-TRUST-REGION METHOD FOR UNCONSTRAINED OPTIMIZATION*

NICK I. M. GOULD[†], CAROLINE SAINVITU[‡], AND PHILIPPE L. TOINT[‡]

Abstract. A new filter-trust-region algorithm for solving unconstrained nonlinear optimization problems is introduced. Based on the filter technique introduced by Fletcher and Leyffer, it extends an existing technique of Gould, Leyffer, and Toint [*SIAM J. Optim.*, 15 (2004), pp. 17–38] for nonlinear equations and nonlinear least-squares to the fully general unconstrained optimization problem. The new algorithm is shown to be globally convergent to at least one second-order critical point, and numerical experiments indicate that it is very competitive with more classical trust-region algorithms.

Key words. unconstrained optimization, filter methods, trust-region algorithms, convergence theory, numerical experiments

AMS subject classifications. 90C30, 65K05, 90C26, 90C06

DOI. 10.1137/040603851

1. Introduction.

Since filter methods were first introduced for constrained nonlinear optimization by Fletcher and Leyffer [5], they have enjoyed considerable interest in their original domain of application [1, 4, 6, 7, 16, 17]. More recently, they have been extended by Gould, Leyffer, and Toint [8] and Gould and Toint [12] to the nonlinear feasibility problem (including nonlinear equations and nonlinear least-squares), which is to minimize the norm of the violations of a set of (possibly nonlinear and/or nonconvex) constraints. It is the purpose of the present paper to consider the further extension of the filter techniques to general unconstrained optimization problems.

The presentation is organized as follows. Section 2 introduces the problem and the new algorithm, whose global convergence to points satisfying second-order optimality conditions is shown in section 3.1. The results of numerical experience with the new method are discussed in section 4, and some conclusions and perspectives are finally presented in section 5.

2. The problem and the new algorithm.

We consider the unconstrained minimization problem

$$\min_{x \in \mathbb{R}^n} \; f(x), \tag{2.1}$$

where f is a twice continuously differentiable function of the variables $x \in \mathbb{R}^n$. An efficient technique for solving this problem is to use Newton's method, which, from a current iterate x_k, computes a trial step s_k by minimizing a model of the objective

[*] Received by the editors November 18, 2004; accepted for publication (in revised form) February 9, 2005; published electronically October 7, 2005.
http://www.siam.org/journals/siopt/16-2/60385.html

[†] Computational Science and Engineering Department, Rutherford Appleton Laboratory, Chilton, Oxfordshire, UK (gould@rl.ac.uk). The work of this author was supported by EPSRC grant GR/S42170.

[‡] Department of Mathematics, University of Namur, 61, rue de Bruxelles, B-5000 Namur, Belgium (caroline.sainvitu@fundp.ac.be, philippe.toint@fundp.ac.be). The work of the third author was conducted in the framework of the Interuniversity Attraction Poles Programme of the Belgian Science Policy Agency.

function consisting of the first three terms of its Taylor's expansion around x_k, yielding a *trial point*

$$x_k^+ = x_k + s_k.$$

Unfortunately, it is well known that such an algorithm may not always be well defined (when the Taylor's model is nonconvex), or convergent from any initial point x_0. These difficulties can be circumvented by restricting the model minimization to a *trust region* containing x_k, in a manner that is now well established (see Conn, Gould, and Toint [2] for an extensive description of trust-region methods and their properties). We propose to further extend such methods by introducing a multidimensional filter technique, whose aim is to encourage convergence to first-order critical points by driving every component of the objective's gradient

$$\nabla_x f(x) \overset{\text{def}}{=} g(x) = (g_1(x), \ldots, g_n(x))^T$$

to zero.

2.1. Computing a trial point. Before indicating how to apply our filter technique, we start by describing how to compute the trial point $x_k^+ = x_k + s_k$ from a current iterate x_k. At each iteration, we define the model of the objective function to be

$$m_k(x_k + s) = f(x_k) + g_k^T s + \frac{1}{2} s^T H_k s,$$

where $g_k = \nabla_x f(x_k)$ and H_k is a symmetric approximation to $\nabla_{xx} f(x_k)$, and consider a *trust region* centered at x_k:

$$\mathcal{B}_k = \{x_k + s \mid \|s\| \le \Delta_k\},$$

where we believe this model to be adequate. (In this definition and below, $\|\cdot\|$ stands for the Euclidean ℓ_2 norm). A trial step s_k is then computed by minimizing the model (possibly only approximately). At variance with classical trust-region methods, we do not require here that

$$\|s_k\| \le \Delta_k \tag{2.2}$$

at every iteration of our algorithm. The convergence analysis that follows requires, as is common in trust-region methods [2, Chapter 6], that this step provides, at iteration k, a *sufficient decrease on the model*, which is to say that

$$m_k(x_k) - m_k(x_k + s_k) \ge \kappa_{\text{mdc}} \max\left[\|g_k\| \min\left[\frac{\|g_k\|}{\beta_k}, \Delta_k\right], |\tau_k| \min[\tau_k^2, \Delta_k^2]\right], \tag{2.3}$$

where κ_{mdc} is a constant in $(0, 1)$, β_k is a positive upper bound on the norm of the Hessian of the model m_k, i.e.,

$$\beta_k \overset{\text{def}}{=} 1 + \|H_k\|,$$

and $\tau_k = \min[0, \lambda_{\min}[H_k]]$. Although this condition seems technical, there are efficient numerical methods to compute s_k that guarantee that it holds (see [9, 13], or, more generally, [2, Chapter 7]). Typical trust-region algorithms then evaluate the objective function at the trial point and accept x_k^+ as the new iterate if the reduction achieved in the objective function is at least a fraction of that predicted by the model. The trust-region radius Δ_k is also possibly enlarged if this is the case, or it is reduced if the achieved reduction is too small.

2.2. The multidimensional filter. We now consider using a filter mechanism to potentially accept x_k^+ as the new iterate more often. The notion of filter is based on that of *dominance*: for our problem, we say that a point x_1 *dominates* a point x_2 whenever

$$|g_i(x_1)| \le |g_i(x_2)| \quad \text{for all} \quad i = 1, \dots, n.$$

Thus, if iterate x_1 dominates iterate x_2 and if we focus our attention on convergence to first-order critical points only, the latter is of no real interest to us since x_1 is at least as good as x_2 for each of the components of the gradient. All we need to do is remember iterates that are not dominated by other iterates by using a structure called a *filter*. We define a *multidimensional filter* $\mathcal{F}$ as a list of n-tuples of the form $(g_{k,1}, \dots, g_{k,n})$, where $g_{k,i} \overset{\text{def}}{=} g_i(x_k)$, such that if g_k and g_ℓ belong to $\mathcal{F}$, then

$$|g_{k,j}| < |g_{\ell,j}| \quad \text{for at least one} \quad j \in \{1, \dots, n\}. \tag{2.4}$$

Filter methods propose to accept a new trial iterate x_k^+ if it is not dominated by any other iterate in the filter.

However, we do not wish to accept a new point x_k^+ if one of the components of $g(x_k^+)$ is arbitrarily close to being dominated by another point already in the filter. In order to avoid this situation, we slightly strengthen our acceptability test and say that a new trial point x_k^+ is *acceptable for the filter* $\mathcal{F}$ if and only if

$$\text{for all} \;\; g_\ell \in \mathcal{F} \qquad \exists\, j \in \{1, \dots, n\} \;:\; |g_j(x_k^+)| \le |g_{\ell,j}| - \gamma_g \|g_\ell\|, \tag{2.5}$$

where $\gamma_g \in (0, 1/\sqrt{n})$ is a small positive constant. If an iterate x_k is acceptable in the sense of (2.5), we may wish to add it to the filter and remove from it every $g_\ell \in \mathcal{F}$ such that $|g_{\ell,j}| > |g_{k,j}|$ for all $j \in \{1, \dots, n\}$.

While the mechanism described so far is adequate for convex problems (where a zero gradient is both necessary and sufficient for second-order criticality), it may be unsuitable for nonconvex ones. Indeed it might prevent progress away from a saddle point, in which case an increase in the gradient components is acceptable. We therefore modify the filter mechanism to ensure that the filter is reset to the empty set after each iteration, giving sufficient descent on the objective function at which the model m_k was detected to be nonconvex, and set an upper bound on the acceptable objective function values to ensure that the obtained decrease is permanent.

We are now able to combine these ideas into an algorithm, whose main objective is to let the filter play the major role in ensuring global convergence within "convex basins," falling back on the usual trust-region method only if things do not go well or if negative curvature is encountered.

Algorithm 2.1. *Filter-Trust-Region Algorithm.*

Step 0 : Initialization.

An initial point x_0 and an initial trust-region radius $\Delta_0 > 0$ are given. The constants $\gamma_g \in (0, 1/\sqrt{n})$, η_1, η_2, γ_1, γ_2, and γ_3 are also given and satisfy

$$0 < \eta_1 \le \eta_2 < 1 \quad \text{and} \quad 0 < \gamma_1 \le \gamma_2 < 1 \le \gamma_3. \tag{2.6}$$

Compute $f(x_0)$ and $g(x_0)$, set $k = 0$. Initialize the filter $\mathcal{F}$ to the empty set and choose $f_{\text{sup}} \ge f(x_0)$. Define two flags `RESTRICT` and `NONCONVEX`, the former to be unset.

Step 1: Determine a trial step.
Compute a finite step s_k that "sufficiently reduces" the model m_k, i.e., that satisfies (2.3) and that also satisfies $\|s_k\| \le \Delta_k$ if `RESTRICT` is set or if m_k is nonconvex. In the latter case, set `NONCONVEX`; otherwise unset it. Compute the trial point $x_k^+ = x_k + s_k$.

Step 2: Compute $f(x_k^+)$ and define the following ratio:

$$\rho_k = \frac{f(x_k) - f(x_k^+)}{m_k(x_k) - m_k(x_k^+)}.$$

If $f(x_k^+) > f_{\rm sup}$, set $x_{k+1} = x_k$, set `RESTRICT` and go to Step 4.

Step 3: Test to accept the trial step.

- Compute $g_k^+ = g(x_k^+)$.
- If x_k^+ is acceptable for the filter $\mathcal{F}$ and `NONCONVEX` is unset:
  Set $x_{k+1} = x_k^+$, unset `RESTRICT` and add g_k^+ to the filter $\mathcal{F}$ if either $\rho_k < \eta_1$ or $\|s_k\| > \Delta_k$.
- If x_k^+ is not acceptable for the filter $\mathcal{F}$ or `NONCONVEX` is set:
  If $\rho_k \ge \eta_1$ and $\|s_k\| \le \Delta_k$, then
    set $x_{k+1} = x_k^+$, unset `RESTRICT` and if `NONCONVEX` is set, set $f_{\rm sup} = f(x_{k+1})$ and reinitialize the filter $\mathcal{F}$ to the empty set;
  else set $x_{k+1} = x_k$ and set `RESTRICT`.

Step 4: Update the trust-region radius.
If $\|s_k\| \le \Delta_k$, update the trust-region radius by choosing

$$\Delta_{k+1} \in \begin{cases} [\gamma_1\Delta_k, \gamma_2\Delta_k] & \text{if} \quad \rho_k < \eta_1, \\ [\gamma_2\Delta_k, \Delta_k] & \text{if} \quad \rho_k \in [\eta_1, \eta_2), \\ [\Delta_k, \gamma_3\Delta_k] & \text{if} \quad \rho_k \ge \eta_2; \end{cases} \tag{2.7}$$

otherwise, set $\Delta_{k+1} = \Delta_k$. Increment k by one and go to Step 1.

Note that, as stated, our algorithm lacks a formal stopping criterion. In practice, one would obviously stop the calculation if $\|g_k\|$ falls below some user-defined tolerance and `NONCONVEX` is unset, or if some fixed maximum number of iterations is exceeded. Also note that our conditions on the step might require us to recompute s_k within the trust region if negative curvature were discovered for the model only after computing a step beyond the trust-region boundary. Fortunately, this is typically a very cheap calculation and can be achieved by backtracking [14] or by other suitable restriction techniques [9].

3. Global convergence.

Global convergence properties of Algorithm 2.1 will be proved under the following assumptions.

$\boxed{\textbf{A1}}$ f is twice continuously differentiable on $\mathbb{R}^n$.

$\boxed{\textbf{A2}}$ The iterates x_k remain in a closed, bounded domain of $\mathbb{R}^n$.

$\boxed{\textbf{A3}}$ For all k, the model m_k is twice differentiable on $\mathbb{R}^n$ and has a uniformly bounded Hessian.

Note that A1, A2, and A3 together imply that there exist constants $\kappa_{\rm l}$, $\kappa_{\rm u} \ge \kappa_{\rm l}$, $\kappa_{\rm ufh} \ge 1$, and $\kappa_{\rm umh} \ge 1$ such that

$$f(x_k) \in [\kappa_{\rm l}, \kappa_{\rm u}], \quad \|\nabla_{xx} f(x_k)\| \le \kappa_{\rm ufh}, \quad \text{and} \quad \|H_k\| \le \kappa_{\rm umh} - 1 \tag{3.1}$$

for all k. Combining this with the definition of β_k, we have that

$$\beta_k \le \kappa_{\rm umh} \tag{3.2}$$

for all k and all x in the convex hull of $\{x_k\}$. For the purpose of our analysis, we shall consider

$$\mathcal{S} = \{k \mid x_{k+1} = x_k + s_k\},$$

the set of *successful iterations*;

$$\mathcal{A} = \{k \mid g_k^+ \text{ is added to the filter}\},$$

the set of *filter iterations*;

$$\mathcal{D} = \{k \mid \rho_k \geq \eta_1\},$$

the set of *sufficient descent iterations*; and

$$\mathcal{N} = \{k \mid \texttt{NONCONVEX} \text{ is set}\},$$

the set of *nonconvex iterations*. Observe that $\mathcal{A} \subseteq \mathcal{S}$ and

$$\mathcal{S} \cap \mathcal{N} = \mathcal{D} \cap \mathcal{N}. \tag{3.3}$$

We conclude this section by stating a crucial property of the algorithm.

LEMMA 3.1. *We have that, for all $k \geq 0$,*

$$f(x_0) - f(x_{k+1}) \geq \sum_{\substack{j=0\\ j \in \mathcal{S}\cap\mathcal{N}}}^{k} [f(x_j) - f(x_{j+1})]. \tag{3.4}$$

Proof. Denoting $\mathcal{S} \cap \mathcal{N} = \{k_i\}$, we observe that the definition of f_{sup} in the algorithm ensures that

$$f(x_{k_{i+1}}) \leq f(x_\ell) < f(x_{k_i})$$

for all i and all $k_i + 1 \leq \ell \leq k_{i+1}$. This directly implies the desired inequality. □

3.1. Convergence to critical points. We first prove the convergence of our algorithm to first-order critical points.

Our first step is to prove that, as long as a first-order critical point is not approached, we do not have infinitely many successful nonconvex iterations in the course of the algorithm. We start by recalling two results from [2] in order to show that the trust-region radius is bounded away from zero in this case.

LEMMA 3.2. *Suppose that* A1–A3 *hold and that $\|s_k\| \leq \Delta_k$. Then we have that*

$$|f(x_k + s_k) - m_k(x_k + s_k)| \leq \kappa_{\text{ubh}} \Delta_k^2, \tag{3.5}$$

where $x_k + s_k \in \mathcal{B}_k$ and

$$\kappa_{\text{ubh}} \overset{\text{def}}{=} \max[\kappa_{\text{ufh}}, \kappa_{\text{umh}}]. \tag{3.6}$$

The proof is identical to that of Theorem 6.4.1 in [2], but we now need to make the additional assumption that $\|s_k\| \leq \Delta_k$ explicit (instead of being implicit, in this reference, in the definition of a trust-region step).

We now show that the trust-region radius must increase if the current iterate is not first-order critical and the trust-region radius is small enough.

LEMMA 3.3. *Suppose that* A1–A3 *hold and that* $\|s_k\| \leq \Delta_k$. *Suppose furthermore that* $g_k \neq 0$ *and that*

$$\Delta_k \leq \frac{\kappa_{\text{mdc}} \|g_k\|(1-\eta_2)}{\kappa_{\text{ubh}}}. \tag{3.7}$$

Then iteration $\rho_k \geq \eta_2$ *and*

$$\Delta_{k+1} \geq \Delta_k. \tag{3.8}$$

The proof is the same as Theorem 6.4.2 in [2] when $\|s_k\| \leq \Delta_k$. As a consequence, we obtain that the radius cannot become too small as long as a first-order critical point is not approached.

LEMMA 3.4. *Suppose that* A1–A3 *hold and that there exists a constant* $\kappa_{\text{lbg}} > 0$ *such that* $\|g_k\| \geq \kappa_{\text{lbg}}$ *for all* k. *Then there is a constant* $\kappa_{\text{lbd}} > 0$ *such that*

$$\Delta_k \geq \kappa_{\text{lbd}} \tag{3.9}$$

for all k.

Proof. Assume that iteration k is the first such that

$$\Delta_{k+1} \leq \gamma_1 \min\left[\Delta_0, \frac{\kappa_{\text{mdc}}\, \kappa_{\text{lbg}}(1-\eta_2)}{\kappa_{\text{ubh}}}\right] \overset{\text{def}}{=} \gamma_1 \delta_0. \tag{3.10}$$

This means that the trust-region radius has been decreased at iteration k, which in turn implies, from the condition in Step 4 of the algorithm, that $\|s_k\| \leq \Delta_k$. We also have that $\gamma_1 \Delta_k \leq \Delta_{k+1}$ and hence that

$$\Delta_k \leq \delta_0 \leq \frac{\kappa_{\text{mdc}}\, \kappa_{\text{lbg}}(1-\eta_2)}{\kappa_{\text{ubh}}}.$$

Our assumption on the norm of the gradient then implies that (3.7) holds. This and the fact that $\|s_k\| \leq \Delta_k$ thus give that (3.8) is satisfied. But this contradicts the fact that iteration k is the first such that (3.10) holds, and our initial assumption is therefore impossible. This yields the desired conclusion with $\kappa_{\text{lbd}} = \gamma_1 \delta_0$. □

We now prove the crucial result that the number of successful nonconvex iterations must be finite unless a first-order critical point is approached.

THEOREM 3.5. *Suppose that* A1–A3 *hold and that there exists a constant* $\kappa_{\text{lbg}} > 0$ *such that* $\|g_k\| \geq \kappa_{\text{lbg}}$ *for all* k. *Then there can be only finitely many successful nonconvex iterations in the course of the algorithm, i.e.,* $|\mathcal{S} \cap \mathcal{N}| < +\infty$.

Proof. Suppose, for the purpose of obtaining a contradiction, that there are infinitely many successful nonconvex iterations, which we index by $\mathcal{S} \cap \mathcal{N} = \{k_i\}$. It follows from (3.3) that the algorithm also guarantees that $\rho_k \geq \eta_1$ for all iterations in $\mathcal{S} \cap \mathcal{N}$, which in turn implies, with (2.3), that for $k \in \mathcal{S} \cap \mathcal{N}$,

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq \eta_1 [m_k(x_k) - m_k(x_k + s_k)] \\ &\geq \eta_1\, \kappa_{\text{mdc}} \|g_k\| \min\left[\frac{\|g_k\|}{\beta_k}, \Delta_k\right] \\ &\geq \eta_1\, \kappa_{\text{mdc}}\, \kappa_{\text{lbg}} \min\left[\frac{\kappa_{\text{lbg}}}{\kappa_{\text{umh}}}, \kappa_{\text{lbd}}\right], \end{aligned}$$

where we have used Lemma 3.4, (3.2), and our lower bound on the gradient norm to obtain the last inequality. Combining now this bound with (3.4), we deduce that

$$f(x_0) - f(x_{k+1}) \geq \sum_{\substack{j=0 \\ j \in \mathcal{S} \cap \mathcal{N}}}^{k} [f(x_j) - f(x_{j+1})] \geq \varsigma_k \, \eta_1 \, \kappa_{\rm mdc} \, \kappa_{\rm lbg} \min \left[\frac{\kappa_{\rm lbg}}{\kappa_{\rm umh}}, \kappa_{\rm lbd} \right],$$

where $\varsigma_k = |\{1, \ldots, k\} \cap \mathcal{S} \cap \mathcal{N}|$. As we have supposed that there are infinitely many successful nonconvex iterations, we have that

$$\lim_{k \to \infty} \varsigma_k = +\infty,$$

and $[f(x_0) - f(x_{k+1})]$ is unbounded above, which contradicts the fact that the objective function is bounded below, as stated in (3.1). Our initial assumption must then be false, and the set $\mathcal{S} \cap \mathcal{N}$ of successful nonconvex iterations must be finite. □

We now establish the criticality of the limit point of the sequence of iterates when there are only finitely many successful iterations.

THEOREM 3.6. *Suppose that* A1–A3 *and* (2.3) *hold and that there are only finitely many successful iterations, i.e.,* $|\mathcal{S}| < +\infty$. *Then* $x_k = x^*$ *for all sufficiently large* k, *and* x^* *is first-order critical.*

Proof. Let k_0 be the index of the last successful iterate. Then $x^* = x_{k_0+1} = x_{k_0+j}$ and

$$\rho_{k_0+j} < \eta_1 \quad \text{for all} \quad j > 1. \tag{3.11}$$

Now observe that `RESTRICT` is set by the algorithm in the course of every unsuccessful iteration. This flag must thus be set at the beginning of every iteration of index $k_0 + j + 1$ for $j > 0$. As a consequence, $\|s_{k_0+j+2}\| \leq \Delta_{k_0+j+2}$ for all $j > 0$. This, (3.11), and the mechanism of Step 4 of the algorithm then imply that

$$\lim_{k \to \infty} \Delta_k = 0. \tag{3.12}$$

Assume now, for the purpose of establishing a contradiction, that $\|g_{k_0+1}\| \geq \varepsilon$ for some $\varepsilon > 0$. Then Lemma 3.4 implies that (3.12) is impossible and we deduce that

$$\|g_{k_0+j}\| = 0$$

for all $j > 0$. □

Having proved the desired convergence property for the case where $\mathcal{S}$ is finite, we restrict our attention, for the rest of this section, to the case where there are infinitely many successful iterations, i.e., $|\mathcal{S}| = +\infty$. We first investigate what happens if infinitely many values are added to the filter in the course of the algorithm.

THEOREM 3.7. *Suppose that* A1–A3 *hold and that* $|\mathcal{A}| = |\mathcal{S}| = +\infty$. *Then*

$$\liminf_{k \to \infty} \|g_k\| = 0. \tag{3.13}$$

Proof. Assume, for the purpose of obtaining a contradiction, that for all k large enough,

$$\|g_k\| \geq \kappa_{\rm lbg} \tag{3.14}$$

for some $\kappa_{\rm lbg} > 0$ and define $\{k_i\} = \mathcal{A}$. The bound (3.14) and Theorem 3.5 then imply that $|\mathcal{S} \cap \mathcal{N}|$ is finite and therefore that the filter is no longer reset to the empty set for k sufficiently large. Moreover, since our assumptions imply that $\{\|g_{k_{i+1}}\|\}$ is bounded above and away from zero, there must exist a subsequence $\{k_\ell\} \subseteq \{k_{i+1}\}$ such that

$$\lim_{\ell\to\infty} g_{k_\ell} = g_\infty \qquad \text{with} \quad \|g_\infty\| \geq \kappa_{\rm lbg}. \tag{3.15}$$

By definition of $\{k_\ell\}$, x_{k_ℓ} is acceptable for the filter in each iteration $\ell - 1$. This implies, since the filter is not reset for ℓ large enough, that, for each ℓ sufficiently large, there exists an index $j_\ell \in \{1, \dots, n\}$ such that

$$|g_{k_\ell, j_\ell}| - |g_{k_{\ell-1}, j_\ell}| < -\gamma_g \|g_{k_{\ell-1}}\|. \tag{3.16}$$

But (3.14) implies that $\|g_{k_{\ell-1}}\| \geq \kappa_{\rm lbg}$ for all ℓ sufficiently large. Hence we deduce from (3.16) that

$$|g_{k_\ell, j_\ell}| - |g_{k_{\ell-1}, j_\ell}| < -\gamma_g \kappa_{\rm lbg}$$

for all ℓ sufficiently large. But the left-hand side of this inequality tends to zero when ℓ tends to infinity because of (3.15), yielding the desired contradiction. Hence (3.13) holds. □

Consider now the case where the number of iterates added to the filter in the course of the algorithm is finite.

THEOREM 3.8. *Suppose that* A1–A3 *hold and that* $|\mathcal{S}| = +\infty$ *but* $|\mathcal{A}| < +\infty$. *Then* (3.13) *holds.*

Proof. Assume, again for the purpose of obtaining a contradiction, that (3.14) holds for all k large enough and for some $\kappa_{\rm lbg} > 0$. The finiteness of $|\mathcal{A}|$ then implies that $\rho_k \geq \eta_1$ and that $\|s_k\| \leq \Delta_k$ for all $k \in \mathcal{S}$ sufficiently large. If we define $\bar{\varsigma}_{p,k} = |\{p, \dots, k\} \cap \mathcal{S}|$, we then obtain that

$$f(x_p) - f(x_{k+1}) = \sum_{\substack{j=p\\ j\in\mathcal{S}}}^{k} [f(x_j) - f(x_{j+1})] \geq \bar{\varsigma}_{p,k}\, \eta_1\, \kappa_{\rm mdc}\, \kappa_{\rm lbg} \min\left[\frac{\kappa_{\rm lbg}}{\kappa_{\rm umh}}, \kappa_{\rm lbd}\right],$$

for p and k sufficiently large, where, as above, we used (2.3), (3.2), (3.9), and (3.14) to derive the inequality. But $\bar{\varsigma}_{p,k}$ tends to infinity with k for a fixed p sufficiently large since $|\mathcal{S}|$ is infinite, and we again derive a contradiction from the fact that $f(x_{k+1})$ then becomes unbounded below. The limit (3.13) then follows. □

By the two last theorems, we have that at least one of the limit points of the sequence of iterates generated by the algorithm satisfies the first-order necessary condition. As the following example shows, this cannot be improved without modifying the algorithm.

Example 3.1. Consider the objective function

$$f(x) = x^3(3x - 4),$$

which has a (degenerate[1]) first-order critical point at $x = 0$, which is not a minimizer, and its global minimizer at $x = 1$. We will show that it is possible for Algorithm 2.1

[1] In other words, both its first and second derivatives vanish.

to construct iterates for which $x_{2k} = -(\frac{1}{2})^k$ and $x_{2k+1} = \frac{5}{4}$ for $k = 0, 1, 2, \ldots$; clearly there are two limit points, $x_*^{\mathrm{L}} = 0$ and $x_*^{\mathrm{R}} = \frac{5}{4}$, but only the first is critical.

Let $\Delta_0 > 2$, and suppose that $\gamma_g < \frac{1}{2}$ and that the trust-region updating scheme (2.7) is specifically

$$\Delta_{k+1} = \begin{cases} \frac{1}{2}\Delta_k & \text{if } \rho_k < \eta_1, \\ \Delta_k & \text{if } \eta_1 \leq \rho_k < \eta_2, \\ 2\Delta_k & \text{if } \eta_2 \leq \rho_k. \end{cases} \tag{3.17}$$

Now suppose that

$$\mathcal{F} = \{f'(x_{2k})\} \equiv \{-12(1 + (\tfrac{1}{2})^k)(\tfrac{1}{2})^{2k}\} \text{ and } \Delta_{2k} > 2. \tag{3.18}$$

We then show that the above iteration is possible for Algorithm 2.1 and that (3.18) will persist.

Consider first $x_{2k} = -(\frac{1}{2})^k$ and the convex model

$$m_{2k}(x_{2k} + s) = f(x_{2k}) + sf'(x_{2k}) + \tfrac{1}{2}s^2 h_{2k}, \text{ where } h_{2k} = -\frac{f'(x_{2k})}{\frac{5}{4} - x_{2k}} > 0.$$

Then the unconstrained global minimizer of m_{2k} is $s_{2k} = \frac{5}{4} - x_{2k}$, and s_{2k} will sufficiently reduce the model within the trust region since $\Delta_{2k} > 2 > \frac{5}{4} + (\frac{1}{2})^k$. Moreover,

$$m_{2k}(x_{2k}) - m_{2k}(x_{2k} + s_{2k}) = \frac{1}{2}\frac{(f'(x_{2k}))^2}{h_{2k}} = \frac{1}{2}\left(\frac{5}{4} - x_{2k}\right) f'(x_{2k}) \to 0$$

while

$$f(x_{2k}) - f(x_{2k} + s_{2k}) = f(x_{2k}) - f\left(\frac{5}{4}\right) > f(0) - f\left(\frac{5}{4}\right) = \frac{125}{256} > 0,$$

and thus

$$\rho_{2k} \geq \eta_2 \tag{3.19}$$

for large enough k. The trial point $x_{2k} + s_{2k}$ is not acceptable for the filter since its gradient is $f'(\frac{5}{4}) = \frac{75}{16} \gg f'(x_{2k})$, but it is an acceptable point because the trust-region bound is inactive and because of (3.19). Thus $x_{2k+1} = x_{2k} + s_{2k} = \frac{5}{4}$, while (3.17) and (3.19) ensure that $\Delta_{2k+1} = 2\Delta_{2k}$.

Now consider $x_{2k+1} = \frac{5}{4}$ and the convex model

$$m_{2k+1}(x_{2k+1} + s) = f(x_{2k+1}) + sf'(x_{2k+1}) + \tfrac{1}{2}s^2 h_{2k+1},$$

where

$$h_{2k+1} = \frac{f'(x_{2k+1})}{x_{2k+1} + (\frac{1}{2})^{k+1}} > 0.$$

As before, the unconstrained global minimizer of m_{2k+1} is $s_{2k+1} = -x_{2k+1} - (\frac{1}{2})^{k+1}$, and s_{2k+1} will sufficiently reduce the model within the trust region since $\Delta_{2k+1} > 4 > \frac{5}{4} + (\frac{1}{2})^k$. Although $f(x_{2k+1}) - f(x_{2k+1} + s_{2k+1}) < 0$ and hence

$$\rho_{2k+1} < 0, \tag{3.20}$$

$x_{2k+1} + s_{2k+1} = -(\frac{1}{2})^{k+1}$ is acceptable for the filter since it is easy to check that

$$|f'(x_{2k+1} + s_{2k+1})| = |f'(-(\tfrac{1}{2})^{k+1})| < \tfrac{1}{2}|f'(x_{2k})|.$$

Hence $x_{2k+2} = x_{2k+1} + s_{2k+1} = -(\frac{1}{2})^{k+1}$. Moreover, (3.17) and (3.20) imply that $f'(x_{2k+2})$ replaces $f'(x_{2k})$ in the filter and that $\Delta_{2k+2} = \frac{1}{2}\Delta_{2k+1} = \Delta_{2k}$, and thus that (3.18) persists.

It is unclear how to enforce the property that all limit points are first-order critical without adversely affecting the algorithm's numerical behavior. We have considered not allowing filter iterations when the ratio between the current gradient norm and the smallest gradient norm found so far exceeds some prescribed (large) constant. While such a modification does not appear to affect the results of our numerical experiments, to date we have been unable to show that the modification yields the desired conclusion. Since we believe that the likelihood of the algorithm converging to more than a single limit point is very small (as with every trust-region method we are aware of), the issue really is of mostly theoretical interest.

We thus pursue our analysis by examining convergence to second-order critical points under the assumption that there is only one limit point. As in [2], we also assume the following:

$\boxed{\textbf{A4}}$ The matrix H_k is arbitrarily close to $\nabla_{xx} f(x_k)$ whenever a first-order critical point is approached; i.e.,

$$\lim_{k\to\infty} \|\nabla_{xx} f(x_k) - H_k\| = 0 \text{ whenever } \lim_{k\to\infty} \|g_k\| = 0.$$

(Notice that $h_{2k} \to 0$ and thus that A4 holds in the above example.)

We are then able to derive the following theorem.

THEOREM 3.9. *Suppose that* A1–A4 *hold and that the complete sequence of iterates* $\{x_k\}$ *converge to the unique limit point* x^*. *Then* x^* *is a second-order critical point.*

Proof. Our proof is strongly inspired by Theorem 6.6.4 of [2]. Observe that our previous results imply that

$$g(x^*) = 0. \tag{3.21}$$

For the purpose of deriving a contradiction, assume now that

$$\tau_* \overset{\text{def}}{=} \lambda_{\min}[\nabla_{xx} f(x^*)] < 0. \tag{3.22}$$

Then, using A4 and (3.21), we deduce that there exists a k_0 such that for $k \geq k_0$,

$$\lambda_{\min}[H_k] < \tfrac{1}{2}\tau_* < 0$$

and, consequently, that $k \in \mathcal{N}$ and

$$\|s_k\| \leq \Delta_k \tag{3.23}$$

for $k \geq k_0$. Our sufficient decrease condition (2.3) then ensures that for $k \geq k_0$,

$$m_k(x_k) - m_k(x_k + s_k) \geq \tfrac{1}{2}\kappa_{\text{mdc}}|\tau_*| \min[\tfrac{1}{4}\tau_*^2, \Delta_k^2]. \tag{3.24}$$

Consider now the ratio of achieved versus predicted reduction ρ_k in the case where $\Delta_k \leq \frac{1}{2}|\tau_*|$. Thus, (3.24) and (3.23) imply that

$$m_k(x_k) - m_k(x_k + s_k) \geq \tfrac{1}{2}\kappa_{\text{mdc}}|\tau_*|\Delta_k^2 \geq \tfrac{1}{2}\kappa_{\text{mdc}}|\tau_*|\|s_k\|^2 \tag{3.25}$$

for $k \geq k_0$. Using the mean value theorem and the Cauchy–Schwarz inequality successively, we obtain that for some ξ_k in the segment $[x_k, x_k + s_k]$,

$$\begin{aligned} |\rho_k - 1| &= \left| \frac{f(x_k + s_k) - m_k(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)} \right| \\ &\leq \frac{|s_k^T \nabla_{xx} f(\xi_k) s_k - s_k^T H_k s_k|}{\kappa_{\mathrm{mdc}} |\tau_*| \|s_k\|^2} \qquad (3.26) \\ &\leq \frac{1}{\kappa_{\mathrm{mdc}} |\tau_*|} \, \|\nabla_{xx} f(\xi_k) - H_k\| \end{aligned}$$

for $k \geq k_0$ and $\Delta_k \leq \frac{1}{2}|\tau_*|$. Since $\|\xi_k - x_k\| \leq \|s_k\| \leq \Delta_k$ for $k \geq k_0$, A1, (3.21), and A4 imply that the rightmost term of (3.26) must be arbitrarily small for Δ_k sufficiently small and k sufficiently large. Thus, there must exist a $k_1 \geq k_0$ and a $\delta_1 \in (0, \frac{1}{2}|\tau_*|]$ such that

$$\rho_k \geq \eta_2 \text{ for all } k \geq k_1 \text{ such that } \Delta_k \leq \delta_1.$$

As a consequence, each iteration where these two conditions hold must be very successful and the algorithm then guarantees that $\Delta_{k+1} \geq \Delta_k$. This and the inequality $\gamma_1 \delta_1 < \delta_1 \leq \frac{1}{2}|\tau_*|$ in turn imply that

$$\Delta_k \geq \min[\gamma_1 \delta_1, \Delta_{k_0}] \overset{\text{def}}{=} \delta_2 \qquad (3.27)$$

for all $k \geq k_1$. For every successful iteration $k \geq k_1$, we then obtain from (3.24) that

$$f(x_k) - f(x_{k+1}) \geq \tfrac{1}{2} \eta_1 \kappa_{\mathrm{mdc}} |\tau_*| \min[\tfrac{1}{4}\tau_*^2, \delta_2^2] > 0.$$

Remembering now that $k \in \mathcal{N}$ for $k \geq k_1$ (and thus that $|\mathcal{N}| = \infty$), we obtain from (3.4) that $|\mathcal{S} \cap \mathcal{N}|$, and hence $|\mathcal{S}|$, must be finite, which in turn implies that the trust-region radius tends to zero. But this contradicts (3.27). Hence our initial assumption (3.22) must be false and the proof is complete. □

We finally note that, at least in theory, nothing prevents the filter size from growing, possibly to infinity. Practically, a very large number of points might therefore be required, and this could, again in principle, be a serious drawback, especially for large-scale instances where each filter point has itself a large number of components. Fortunately, this problem can be fixed without sacrificing our convergence guarantee. Should the problem arise in that, at some iteration, the total storage for filter points reaches a user-defined upper limit, two different techniques can be used to continue the calculation. The first is simply to revert to a pure trust-region scheme from that iteration on. Admittedly, we would then lose some of the potential benefits of using a filter technique, but convergence is not put at risk. The second strategy is a progressive form of the first. As indicated in [8], the components of the gradient can be grouped in progressively larger sets (the filter entries being then defined as the Euclidean norm of the subvector of components belonging to the set). This results in a progressive decrease of the amount of storage required to store the entire filter. In the limit where a single component set is considered and assuming dominated filter points are removed, the filter reduces to a single number (an upper bound on the Euclidean norm of the gradient), thus eliminating all storage problems.

4. Numerical experiments. We now report the results obtained by running our algorithm on the set of 159 unconstrained[2] problems from the CUTEr collection [10]. The names of the problems with their dimensions[3] are detailed in Table 4.1.

In each case, the starting point supplied with the problem was used. All tests were performed in double precision on a Dell Latitude C840 portable computer (1.6 Mhz, 1 Gbyte of RAM) under Red Hat 9.0 Linux and the Lahey Fortran compiler (version L6.10a) with default options. All attempts to solve the test problems were limited to a maximum of 1000 iterations or 1 hour of CPU time. The values $\gamma_1 = 0.0625$, $\gamma_2 = 0.25$, $\gamma_3 = 2$, $\eta_1 = 0.01$, $\eta_2 = 0.9$, $\Delta_0 = 1$, and

$$\gamma_g = \min\left[0.001, \frac{1}{2\sqrt{n}}\right]$$

were used.

Two particular variants were tested. The first (called default) is the algorithm as described above, where exact first and second derivatives are used and where, at each iteration, the trial point is computed by approximately minimizing $m_k(x_k + s)$ using the generalized Lanczos trust-region algorithm of [9] (without preconditioning) as implemented in the GALAHAD library [11]. This procedure is terminated at the first s for which

$$\|\nabla m_k(x_k + s)\| \leq \min\left[0.1, \sqrt{\max(\epsilon_M, \|\nabla m_k(x_k)\|)}\right] \|\nabla m_k(x_k)\|, \tag{4.1}$$

where ϵ_M is the machine precision. In addition, we choose

$$f_{\text{sup}} = \min(10^6 |f(x_0)|, f(x_0) + 1000)$$

at Step 0 of the algorithm. Based on practical experience [12], we also impose that $\|s_k\| \leq 1000\Delta_k$ at all iterations following the first one at which a restricted step was taken. The algorithm stops if

$$\|\nabla f(x_k)\| \leq 10^{-6}\sqrt{n}. \tag{4.2}$$

Finally, dominated filter points are always removed from the filter. The second algorithmic variant is the pure trust-region version, which is the same algorithm with the exception that no trial point is ever accepted for the filter and RESTRICT is always set.

On the 159 problems, both the default and the pure trust-region versions successfully solve 143. For the problems where both variants succeed, they report the same final objective function value. Failure occurs because the maximal iteration count is reached before convergence is declared, except for problems ARGLINB and ARGLINC that are judged to be too ill-conditioned by the default version, and for problems MEYER3, SCURLY20, and SCURLY30, where the pure trust-region variant stops for the same reason. The filter variant is thus just as reliable[4] as the trust-region version.

Figures 4.1, 4.2, and 4.3 give the performance profiles for the two variants for iterations, CPU time, and the total number of conjugate-gradient iterations, respectively. Performance profiles give, for every $\sigma \geq 1$, the proportion $p(\sigma)$ of test problems on which each considered algorithmic variant has a performance within a factor σ of

[2] We excluded problem BROYDN7D because of its multiple local minima.

[3] The number of free variables.

[4] The two variants consistently fail on CHAINWOO, HYDC20LS, LMINSURF, LOGHAIRY, MEYER3, NLMSURF, NONCVXU2, NONCVXUN, SBRYBND, SCOSINE, and SCURLY10.

Table 4.1
The test problems and their dimensions.

Problem	n	Problem	n	Problem	n
AIRCRFTB	5	DQRTIC	5000	OSBORNEA	5
ALLINITU	4	EDENSCH	10000	OSBORNEB	11
ARGLINA	200	EG2	1000	PALMER1C	8
ARGLINB	200	EIGENALS	2550	PALMER1D	7
ARGLINC	200	EIGENBLS	2550	PALMER2C	8
ARWHEAD	5000	EIGENCLS	2652	PALMER3C	8
BARD	3	ENGVAL1	10000	PALMER4C	8
BDQRTIC	5000	ENGVAL2	2	PALMER5C	6
BEALE	2	ERRINROS	50	PALMER6C	8
BIGGS3	3	EXPFIT	2	PALMER7C	8
BIGGS5	5	EXTROSNB	1000	PALMER8C	8
BIGGS6	6	FMINSRF2	5625	PARKCH	15
BOX2	2	FMINSURF	49	PENALTY1	1000
BOX3	3	FREUROTH	5000	PENALTY2	200
BRKMCC	2	GENROSE	500	PENALTY3	200
BROWNAL	200	GROWTHLS	3	POWELLSG	5000
BROWNBS	2	GULF	3	POWER	100
BROWNDEN	4	HAIRY	2	QUARTC	5000
BRYBND	5000	HATFLDD	3	RAYBENDL	2046
CHAINWOO	4000	HATFLDE	3	RAYBENDS	2046
CHNROSNB	50	HEART6LS	6	ROSENBR	2
CLIFF	2	HEART8LS	8	S308	2
CLPLATEA	10100	HELIX	3	SBRYBND	500
CLPLATEB	4970	HIELOW	3	SCHMVETT	5000
CLPLATEC	4970	HILBERTA	2	SCOSINE	5000
COSINE	10000	HILBERTB	10	SCURLY10	100
CRAGGLVY	5000	HIMMELBB	2	SCURLY20	100
CUBE	2	HIMMELBF	4	SCURLY30	100
CURLY10	10000	HIMMELBG	2	SENSORS	100
CURLY20	10000	HIMMELBH	2	SINEVAL	2
CURLY30	1000	HYDC20LS	99	SINQUAD	10000
DECONVU	61	JENSMP	2	SISSER	2
DENSCHNA	2	KOWOSB	4	SNAIL	2
DENSCHNB	2	LIARWHD	5000	SPARSINE	5000
DENSCHNC	2	LMINSURF	5329	SPARSQUR	10000
DENSCHND	3	LOGHAIRY	2	SPMSRTLS	4900
DENSCHNE	3	MANCINO	100	SROSENBR	5000
DENSCHNF	2	MARATOSB	2	SSC	4900
DIXMAANA	9000	MEXHAT	2	STRATEC	10
DIXMAANB	9000	MEYER3	3	TESTQUAD	5000
DIXMAANC	9000	MINSURF	36	TOINTGOR	50
DIXMAAND	9000	MOREBV	5000	TOINTGSS	5000
DIXMAANE	9000	MSQRTALS	1024	TOINTPSP	50
DIXMAANF	9000	MSQRTBLS	1024	TOINTQOR	50
DIXMAANG	9000	NCB20	5010	TQUARTIC	5000
DIXMAANH	9000	NCB20B	5000	TRIDIA	5000
DIXMAANI	9000	NLMSURF	5329	VARDIM	200
DIXMAANJ	9000	NONCVXU2	5000	VAREIGVL	50
DIXMAANK	9000	NONCVXUN	5000	VIBRBEAM	8
DIXMAANL	9000	NONDIA	5000	WATSON	12
DIXON3DQ	10000	NONDQUAR	5000	WOODS	10000
DJTL	2	NONMSQRT	100	YFITU	3
DQDRTIC	5000	ODC	4900	ZANGWIL2	2

the best (see [3] for a more complete discussion). When comparing CPU times, we must take into account the variability of reported CPU times for identical runs on the same machine. We have chosen to round all reported times to the nearest multiple

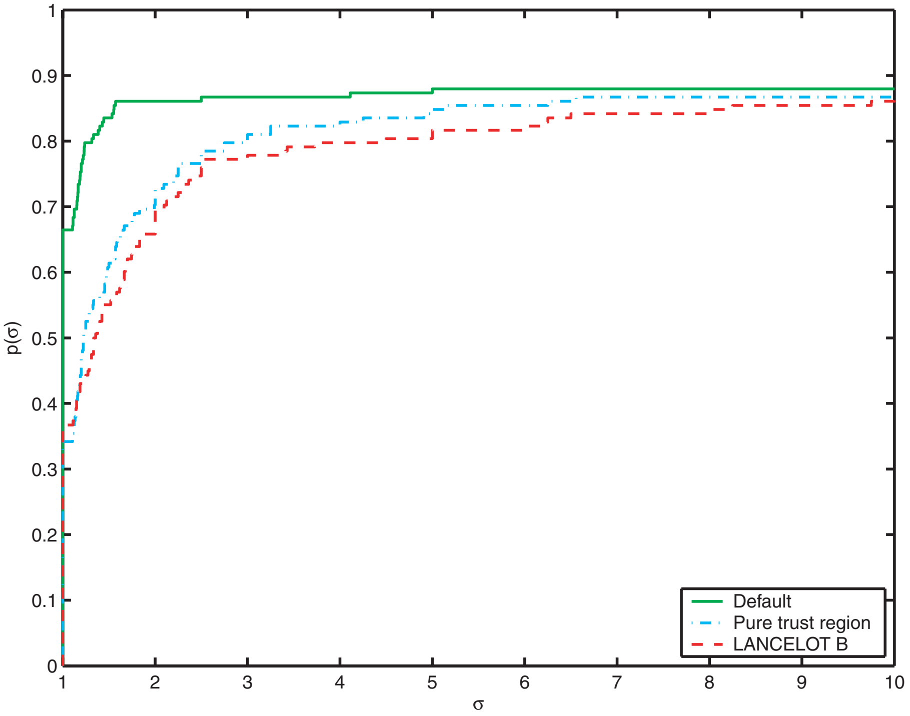


Fig. 4.1. *Iteration performance profiles for the two variants and* LANCELOT B.

of 0.1 second. Problems for which all variants required zero seconds (after rounding) are not included in the comparison since the ranking of algorithms with CPU times less than clock accuracy is, in our opinion, of doubtful relevance, both because it is very unreliable and also because its practical impact is negligible (all algorithms are extremely quick in this case). We have also chosen to replace all remaining zero times by the average of the class of times that are rounded to zero (assuming uniform distribution), that is, in our case, by 0.025 (the middle of the interval $[0, 0.05]$).

It is not difficult to see in these figures that the filter variant is significantly more efficient than the pure trust-region method in terms of the number of iterations (which is identical to the number of function evaluations minus one). Its advantage is smaller but significant in terms of CPU time and conjugate-gradient iterations. Interestingly, the cost of managing the filter does not appear to dominate the calculation, despite the potentially large number of entries. A closer look at the results shows that the maximum number of filter entries does not exceed 5 for 119 problems, lies between 6 and 10 for 11 problems, lies between 11 and 50 for 11 problems, and exceeds 50 for 4 problems only: EIGENBLS (85 entries), RAYBENDS (340 entries), SCURLY20 (176 entries), and SCURLY30 (233 entries). None of the three last problems could be solved by the pure trust-region method. Moreover, we did not observe any obvious correlation between filter size and number of variables.

The profiles also include a comparison with LANCELOT-B, one of the GALAHAD codes [11]. This is a nonmonotone trust-region algorithm (see [15] or [2, section 10.1]), which we used unpreconditioned with $\Delta_0 = 1$ and with its other settings at their default values. Again this method, which successfully solves 141 out of 159 problems, appears to be consistently inferior to the new filter algorithm. It does not solve RAYBENDS, SCURLY20, or SCURLY30 either. This comparison is interesting in that it suggests not only that the improved performance of the new algorithm might be due

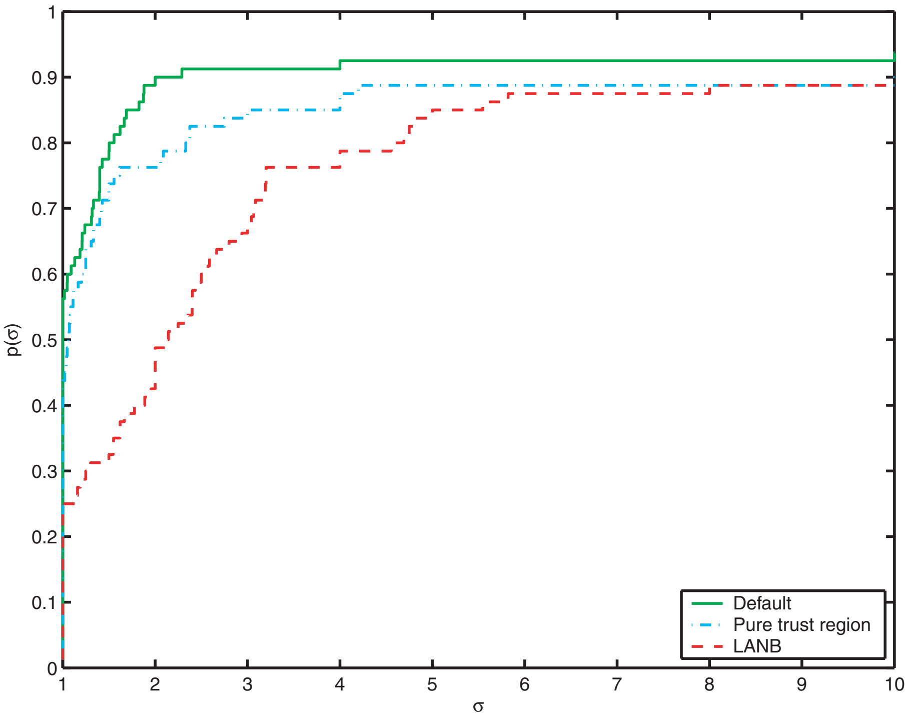


Fig. 4.2. *CPU performance profiles for the two variants and* LANCELOT B.

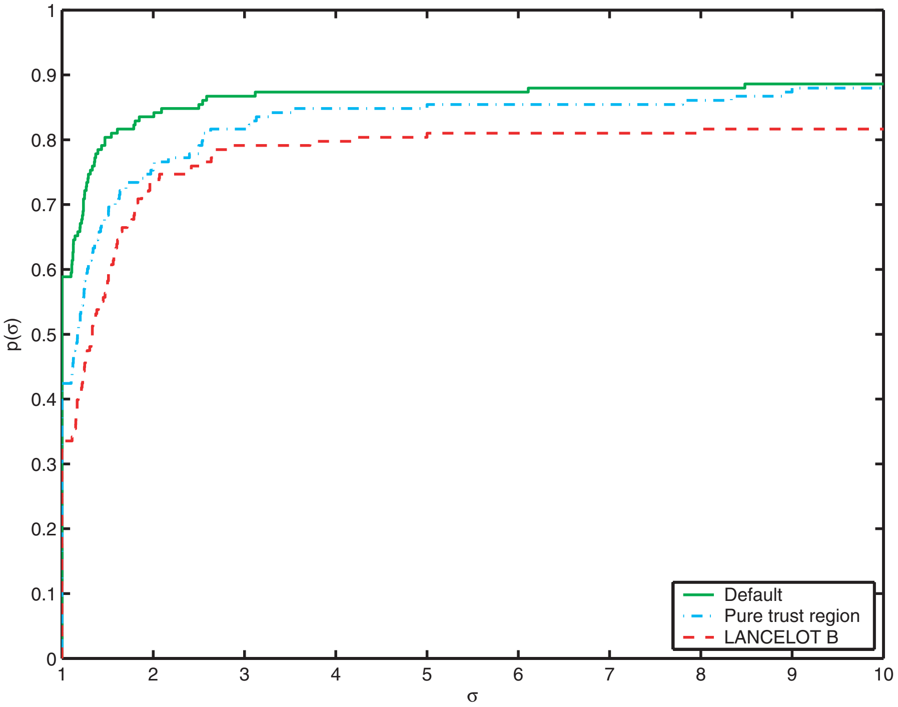


Fig. 4.3. *CG iteration performance profiles for the two variants and* LANCELOT B.

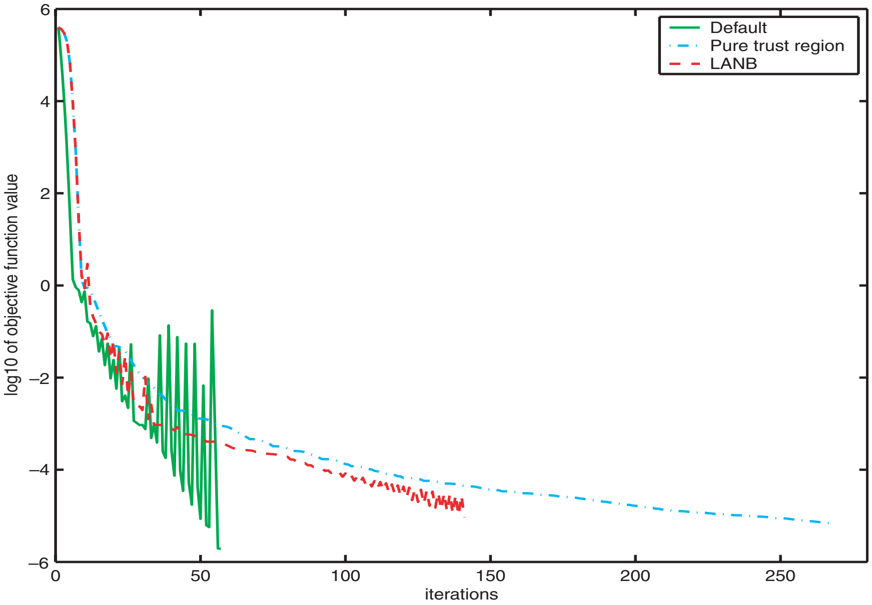


FIG. 4.4. *The objective function value as a function of the iteration progress on the* EXTROSNB *problem for the two variants and* LANCELOT B*. The default variant oscillates the most and converges first, followed by the moderately nonmonotone* LANCELOT B*, itself followed by the monotone pure trust-region variant.*

to the nonmonotone nature of the mechanism to accept new iterates, but also that the capability to use steps that extend beyond the trust-region boundaries is also crucial.

We finally present in Figure 4.4 a plot of the evolution of the objective function value for the default and trust-region variants, as well as for LANCELOT B. This plot is typical of the cases where the new algorithm outperforms the others. For this algorithm, we note the large oscillations in objective value prior to convergence. Looking at this figure, it is remarkable that the algorithm is nevertheless provably convergent.

5. Conclusion. We have presented a filter algorithm for unconstrained optimization and have shown, under standard assumptions, that it produces at least a first-order critical point, irrespective of the chosen starting point. Under mild additional conditions, we also proved that convergence of the complete sequence of iterates can occur only to a second-order critical point. Preliminary numerical experience on the set of unconstrained test problems from the CUTEr collection indicates that significant gains in CPU time and in the number of iterations and function/gradient evaluations can be achieved.

Acknowledgment. The authors are indebted to two anonymous referees for their constructive comments.

REFERENCES

[1] C. M. Chin and R. Fletcher, *On the global convergence of an SLP-filter algorithm that takes EQP steps*, Math. Program., 96 (2003), pp. 161–177.
[2] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust-Region Methods*, MPS-SIAM Ser. Optim. 1, SIAM, Philadelphia, 2000.
[3] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program., 91 (2002), pp. 201–213.

[4] R. Fletcher, N. I. M. Gould, S. Leyffer, Ph. L. Toint, and A. Wächter, *Global convergence of a trust-region SQP-filter algorithm for nonlinear programming*, SIAM J. Optim., 13 (2002), pp. 635–659.

[5] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Math. Program., 91 (2002), pp. 239–269.

[6] R. Fletcher, S. Leyffer, and Ph. L. Toint, *On the global convergence of a filter–SQP algorithm*, SIAM J. Optim., 13 (2002), pp. 44–59.

[7] C. C. Gonzaga, E. Karas, and M. Vanti, *A globally convergent filter method for nonlinear programming*, SIAM J. Optim., 14 (2003), pp. 646–669.

[8] N. I. M. Gould, S. Leyffer, and Ph. L. Toint, *A multidimensional filter algorithm for nonlinear equations and nonlinear least-squares*, SIAM J. Optim., 15 (2004), pp. 17–38.

[9] N. I. M. Gould, S. Lucidi, M. Roma, and Ph. L. Toint, *Solving the trust-region subproblem using the Lanczos method*, SIAM J. Optim., 9 (1999), pp. 504–525.

[10] N. I. M. Gould, D. Orban, and Ph. L. Toint, CUTEr*, a constrained and unconstrained testing environment, revisited*, ACM Trans. Math. Software, 29 (2003), pp. 373–394.

[11] N. I. M. Gould, D. Orban, and Ph. L. Toint, GALAHAD*—a library of thread-safe Fortran* 90 *packages for large-scale nonlinear optimization*, ACM Trans. Math. Software, 29 (2003), pp. 353–372.

[12] N. I. M. Gould and Ph. L. Toint, FILTRANE*, a Fortran* 95 *Filter-Trust-Region Package for Solving Systems of Nonlinear Equalities, Nonlinear Inequalities and Nonlinear Least-Squares Problems*, Technical report 03/15, Rutherford Appleton Laboratory, Chilton, Oxfordshire, UK, 2003.

[13] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[14] J. Nocedal and Y. Yuan, *Combining trust region and line search techniques*, in Advances in Nonlinear Programming, Appl. Optim. 14, Y. Yuan, ed., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998, pp. 153–175.

[15] Ph. L. Toint, *A non-monotone trust-region algorithm for nonlinear optimization subject to convex constraints*, Math. Programming, 77 (1997), pp. 69–94.

[16] M. Ulbrich, S. Ulbrich, and L. N. Vicente, *A globally convergent primal-dual interior-point filter method for nonlinear programming*, Math. Program., 100 (2004), pp. 379–410.

[17] A. Wächter and L. T. Biegler, *Global and Local Convergence of Line Search Filter Methods for Nonlinear Programming*, Technical report CAPD B-01-09, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA, 2001.

SIAM J. Optim.
Vol. 16, No. 2, pp. 358–367

ADJUSTED SUBLEVEL SETS, NORMAL OPERATOR, AND QUASI-CONVEX PROGRAMMING*

D. AUSSEL[†] AND N. HADJISAVVAS[‡]

Abstract. A new notion of "adjusted sublevel set" of a function is introduced and studied. These sets lie between the sublevel and strict sublevel sets of the function. In contrast to the normal operators to sublevel or strict sublevel sets that were studied in the literature so far, the normal operator to the adjusted sublevel sets is both quasi-monotone and, in the case of quasi-convex functions, cone upper-semicontinuous. This makes this new notion appropriate for all kinds of quasi-convex functions and, in particular, for quasi-convex functions whose graph presents a "flat part." Application is given to quasi-convex optimization through the study of an associated variational inequality problem.

Key words. quasi-convexity, normal operator, quasi-convex programming, sublevel set

AMS subject classifications. Primary, 49J52; Secondary, 49J40, 26A51, 26B25, 90C26

DOI. 10.1137/040606958

1. Introduction.

Let X be a Banach space and $f : X \to \mathbb{R} \cup \{+\infty\}$ be a quasi-convex function. The aim of this paper is to propose and study a new concept of sublevel set and its associated normal operator to sublevel sets of a quasi-convex function, and then provide an application to quasi-convex optimization. The idea of using normal cones to the sublevel sets $S_{f(x)} = \{y \in X : f(y) \le f(x)\}$ or strict sublevel sets $S^<_{f(x)} = \{y \in X : f(y) < f(x)\}$ is due to the fact that, in contrast to convexity that can be described through the convexity of the epigraph, quasi-convexity is related to convexity of the sublevel sets. The idea was exploited by Borde and Crouzeix [5], who mainly established continuity properties; by Aussel and Daniilidis [3], who characterized some classes of quasi-convex functions; and by Eberhard and Crouzeix [8], who studied the integration of these operators as a means to obtain the quasi-convex function.

However, in those papers the most meaningful results were found in the case where, roughly speaking, f admits no "flat parts." If $S_{f(x)}\backslash\overline{S^<_{f(x)}} \neq \emptyset$ for some $x \in \operatorname{dom} f$ (i.e., there exists a flat part), then none of the normal operators defined in the literature is able to satisfy at the same time quasi-monotonicity and upper semicontinuity in a sense appropriate for cone-valued operators, even if the considered function is lower semicontinuous and quasi-convex (see [5, Example 2.2] and Example 2.1 below). This has induced the authors of previous studies on the subject to restrict their attention to the class of quasi-convex functions such that each local minimum is a global minimum (or, equivalently, $\overline{S^<_\lambda} = S_\lambda$ for all $\lambda > \inf f$).

In section 2 we propose a concept of "adjusted sublevel set" $S^a(x)$ which allows us to deal with all kinds of quasi-convex functions. Based on these adjusted sublevel sets we then define the normal operator. In section 3 we study the properties of the normal

*Received by the editors April 2, 2004; accepted for publication (in revised form) February 10, 2005; published electronically October 7, 2005.
http://www.siam.org/journals/siopt/16-2/60695.html

[†] Lab. MANO, University of Perpignan, 52 av. P. Alduy, 66860 Perpignan Cedex, France (aussel@univ-perp.fr).

[‡] Department of Product and Systems Design Engineering, University of the Aegean, Hermoupolis, 84100 Syros, Greece (nhad@aegean.gr).

operator and, in particular, some nonemptiness properties, quasi-monotonicity, and continuity results.

Finally, using this normal operator and our recent study of quasi-monotone variational inequalities [4] we prove an existence result for the minimization of a quasi-convex function over a convex set.

2. Definitions and basic properties.

Let X be a real Banach space, X^* its topological dual, and $\langle\cdot,\cdot\rangle$ the duality pairing. The topological closure of a set A will be denoted by $\overline{A}$ for the norm topology and $\overline{A}^*$ for the w* topology, whereas $x_i^* \overset{w^*}{\rightharpoonup} x^*$ means that $x_i^* \to x^*$ in the w* topology. As usual, $\operatorname{co} A$ will denote the convex hull of A. We denote by $B(x,\varepsilon)$ and $\overline{B}(x,\varepsilon)$ the open ball $\{y \in X : \|y-x\| < \varepsilon\}$ and the closed ball $\{y \in X : \|y-x\| \leq \varepsilon\}$. Also, given any nonempty $A \subseteq X$ we denote by $B(A,\varepsilon)$ and $\overline{B}(A,\varepsilon)$ the sets $\{x \in X : dist(x,A) < \varepsilon\}$ and $\{x \in X : dist(x,A) \leq \varepsilon\}$, respectively, where $dist(x,A) = \inf\{\|x-y\| : y \in A\}$ is the distance of x from A. Given $x, y \in X$, we set $[x,y] = \{tx + (1-t)y : t \in [0,1]\}$. The domain and the graph of a multivalued operator $T : X \to 2^{X^*}$ will be denoted, respectively, by $\operatorname{dom}(T)$ and $\operatorname{gr} T$. We will mainly deal with operators whose values are convex cones; in this case, since the values are unbounded, we have to consider a modified definition of upper semicontinuity. We first recall that a convex subset C of a convex cone L in X^* is called a base if $0 \notin \overline{C}^*$ and $L = \bigcup_{t\geq 0} tC$.

DEFINITION 2.1. *An operator $T : X \to 2^{X^*}$ whose values are convex cones is called norm-to-w* cone upper-semicontinuous at $x \in \operatorname{dom}(T)$ if there exist a neighborhood U of x and a base $C(u)$ of $T(u)$ for each $u \in U$, such that $u \to C(u)$ is norm-to-w* upper semicontinuous at x.*

It turns out that we may always suppose that, locally, the base $C(u)$ is the intersection of $T(u)$ with a fixed hyperplane. To see this, we first define a conic w*-neighborhood of a cone L in X^* to be a w*-open cone M (i.e., a w*-open set such that $tM \subseteq M$ for all $t > 0$) such that $L \subseteq M \cup \{0\}$.

PROPOSITION 2.2. *Let $T : X \to 2^{X^*}$ be a multivalued operator whose values are convex cones different from $\{0\}$. Given $x \in \operatorname{dom}(T)$, the following are equivalent:*

(i) *T is norm-to-w* cone upper-semicontinuous at x.*

(ii) *$T(x)$ has a base, and for every conic w*-neighborhood M of $T(x)$ there exists a neighborhood U of x such that $T(u) \subseteq M \cup \{0\}$ for all $u \in U$.*

(iii) *There exists a w*-closed hyperplane A of X^* and a neighborhood U of x such that for all $u \in U$, $D(u) = T(u) \cap A$ is a base of $T(u)$ and the operator D is norm-to-w* upper semicontinuous at x.*

Proof. If (i) holds and M is a conic w*-neighborhood of $T(x)$, then M is a w*-neighborhood of $C(x)$. Hence there exists a neighborhood U of x such that $C(u) \subseteq M$ for every $u \in U$. Then obviously $T(u) \subseteq M \cup \{0\}$.

Suppose that (ii) holds. Then $T(x)$ has a base $C(x)$. Since $0 \notin \overline{C(x)}^*$, by convex separation we deduce the existence of some $x_1 \in X$ such that $\langle x^*, x_1\rangle > 0$ for all $x^* \in C(x)$. The set $B = \{x^* \in X^* : \langle x^*, x_1\rangle > 0\}$ is a conic neighborhood of $T(x)$; hence there exists a neighborhood U of x such that for every $u \in U$ one has $T(u) \subseteq B \cup \{0\}$. Set $A = \{x^* \in X^* : \langle x^*, x_1\rangle = 1\}$. Since $T(u) \neq \{0\}$ it follows that $D(u) := T(u) \cap A$ is a base of $T(u)$. To show the semicontinuity of D let us consider a w*-open neighborhood V of $D(x)$. Then $V \cap A$ is w*-open in A. The function $f : B \to A$ defined by $f(x^*) = \frac{x^*}{\langle x^*, x_1\rangle}$ is w*-continuous; thus, the set $\bigcup_{t>0} t(V \cap A) = f^{-1}(V \cap A)$ is w*-open. From $D(x) \subseteq (V \cap A)$ we deduce that $\bigcup_{t>0} t(V \cap A)$ is a conic w*-neighborhood of $T(x)$. Since (ii) holds, there exists a neighborhood U_1 of x, $U_1 \subseteq U$, such that $T(u) \subseteq \bigcup_{t>0} t\,(V \cap A) \cup \{0\}$. It follows immediately that $D(u) \subseteq V$, i.e., (iii) holds.

Finally, (iii) obviously implies (i). □

A definition of upper semicontinuity suitable for cone-valued operators, similar to property (ii) in the proposition above, was given in [13] (where continuity was taken with respect to the norm topology) and in [5] (where the definition was given in a finite-dimensional setting), the main difference being that in these papers no reference to bases was made.

Given a set $A \subseteq X$, the negative polar cone of A will be denoted by A^-. Let $f : X \to \mathbb{R} \cup \{+\infty\}$ be a function. For any $\lambda \in \mathbb{R}$, define $S_\lambda = \{y \in X \;:\; f(y) \leq \lambda\}$, $S^<_\lambda = \{y \in X \;:\; f(y) < \lambda\}$, $S^=_\lambda = \{y \in X : f\,(y) = \lambda\}$, and, for any $x \in \operatorname{dom} f \setminus \arg\min f$, $\rho_x = dist(x, S^<_{f(x)})$.

Definition 2.3. *Let $f : X \to \mathbb{R} \cup \{+\infty\}$ be any function. To any element $x \in \operatorname{dom} f$ we associate the* adjusted sublevel set *$S^a_f(x)$ defined by*

$$S^a_f(x) = S_{f(x)} \cap \overline{B}\left(S^<_{f(x)}, \rho_x\right)$$

if $x \notin \arg\min f$, and $S^a_f(x) = S_{f(x)}$ otherwise.

Clearly x is always an element of $S^a_f(x)$. If $x \in \operatorname{dom} f \setminus \arg\min f$ is such that $\rho_x = 0$, then $S^a_f(x) = S_{f(x)} \cap \overline{S^<_{f(x)}}$; if, moreover, f is lower semicontinuous on $\operatorname{dom} f$, then $S^a_f(x) = \overline{S^<_{f(x)}}$.

The convexity of the sublevel sets (resp., strict sublevel sets) characterizes the quasi-convexity of the function. This still holds true for the adjusted sublevel sets.

Proposition 2.4. *Let $f : X \to \mathbb{R} \cup \{+\infty\}$ be any function, with domain $\operatorname{dom} f$. Then*

$$f \textit{ is quasi-convex} \iff S^a_f(x) \textit{ is convex}, \;\forall\, x \in \operatorname{dom} f.$$

Proof. Let us suppose that $S^a_f\,(u)$ is convex for every $u \in \operatorname{dom} f$. We will show that for any $x \in \operatorname{dom} f$, $S_{f(x)}$ is convex. If $x \in \arg\min f$, then $S_{f(x)} = S^a_f\,(x)$ is convex by assumption. Assume now that $x \notin \arg\min f$ and take $y, z \in S_{f(x)}$.

If both y and z belong to $\overline{B}(S^<_{f(x)}, \rho_x)$, then $y, z \in S^a_f\,(x)$; thus $[y, z] \subseteq S^a_f\,(x) \subseteq S_{f(x)}$.

If both y and z do not belong to $\overline{B}(S^<_{f(x)}, \rho_x)$, then $f(x) = f(y) = f(z)$, $\overline{S^<_{f(z)}} = \overline{S^<_{f(y)}} = \overline{S^<_{f(x)}}$, and ρ_y, ρ_z are positive. If, say, $\rho_y \geq \rho_z$, then $y, z \in \overline{B}(\overline{S^<_{f(y)}}, \rho_y)$; thus $y, z \in S^a_f\,(y)$ and $[y, z] \subseteq S^a_f\,(y) \subseteq S_{f(y)} = S_{f(x)}$.

Finally, suppose that only one of y, z, say z, belongs to $\overline{B}(S^<_{f(x)}, \rho_x)$ while $y \notin \overline{B}(S^<_{f(x)}, \rho_x)$. Then $f(x) = f(y)$, $\overline{S^<_{f(y)}} = \overline{S^<_{f(x)}}$, and $\rho_y > \rho_x$; thus we have $z \in \overline{B}(S^<_{f(x)}, \rho_x) \subseteq \overline{B}(\overline{S^<_{f(y)}}, \rho_y)$ and we deduce as before that $[y, z] \subseteq S^a_f\,(y) \subseteq S_{f(y)} = S_{f(x)}$.

The other implication is straightforward. □

An operator T is called

quasi-monotone if for every (x, x^*), $(y, y^*) \in \operatorname{gr} T$ the following implication holds:

$$\langle x^*, y - x\rangle > 0 \Rightarrow \langle y^*, y - x\rangle \geq 0;$$

cyclically quasi-monotone if for every $(x_i, x_i^*) \in \operatorname{gr} T$, $i = 1, 2, \ldots, n$, the following implication holds:

$$\langle x_i^*, x_{i+1} - x_i\rangle > 0 \;\forall i = 1, 2, \ldots, n-1 \Rightarrow \langle x_n^*, x_{n+1} - x_n\rangle \leq 0,$$

where $x_{n+1} = x_1$;

cyclically monotone if for every $(x_i, x_i^*) \in \operatorname{gr} T$, $i = 1, 2, \ldots, n$,

$$\sum_{i=1}^{n} \langle x_i^*, x_{i+1} - x_i \rangle \leq 0.$$

By analogy to convex functions, it is known that a lower semicontinuous function is quasi-convex if and only if its Clarke–Rockafellar subdifferential is quasi-monotone [2], [12], or cyclically quasi-monotone [6].

Let $f : X \to \mathbb{R} \cup \{+\infty\}$ be a function. Set

$$N(x) = \left\{x^* \in X^* : \langle x^*, y - x\rangle \leq 0 \ \forall y \in S_{f(x)}\right\},$$
$$N^<(x) = \left\{x^* \in X^* : \langle x^*, y - x\rangle \leq 0 \ \forall y \in S^<_{f(x)}\right\}$$

for every $x \in \operatorname{dom} f$, while we set $N(x) = N^<(x) = \emptyset$ for $x \notin \operatorname{dom} f$. Equivalently, $x^* \in N(x)$ if and only if the following implication holds:

$$\langle x^*, y - x\rangle > 0 \Rightarrow f(y) > f(x);$$

also, $x^* \in N^<(x)$ if and only if

$$\langle x^*, y - x\rangle > 0 \Rightarrow f(y) \geq f(x).$$

These "normal operators" were studied in [5] for functions defined on $\mathbb{R}^n$. They have interesting properties: N is always cyclically quasi-monotone. Also, it can be shown that $N^<$ is cone upper-semicontinuous at every point x where f is lower semicontinuous, provided that there exists $\lambda < f(x)$ such that $\operatorname{int} S_\lambda \neq \emptyset$ (see Proposition 2.2 of [5] for an equivalent statement). However, these two operators are essentially adapted to the class of quasi-convex functions such that any local minimum is a global minimum (in particular, semistrictly quasi-convex functions). In this case, for each $x \in \operatorname{dom} f \setminus \arg\min f$, the sets $S_{f(x)}$ and $S^<_{f(x)}$ have the same closure and $N(x) = N^<(x)$. For quasi-convex functions outside of this class, in general N is not cone upper-semicontinuous (see Example 2.2 in [5]) while $N^<$ is not, in general, quasi-monotone.

Example 2.1. *Define* $f : \mathbb{R}^2 \to \mathbb{R}$ *by*

$$f(a, b) = \begin{cases} |a| + |b| & \text{if } |a| + |b| \leq 1, \\ 1 & \text{if } |a| + |b| > 1. \end{cases}$$

Then f is quasi-convex. Consider $x = (10, 0)$, $x^* = (1, 2)$, $y = (0, 10)$, and $y^* = (2, 1)$. We see that $x^* \in N^<(x)$ and $y^* \in N^<(y)$ (since $|a| + |b| < 1$ implies $(1, 2) \cdot (a - 10, b) \leq 0$ and $(2, 1) \cdot (a, b - 10) \leq 0$) while $\langle x^*, y - x\rangle > 0$ and $\langle y^*, y - x\rangle < 0$. Hence $N^<$ is not quasi-monotone.

In what follows, we will define an operator that has both these properties (cone upper-semicontinuous and quasi-monotonicity) and, consequently, is suitable for relating the minimization of a quasi-convex, lower semicontinuous function f to the variational inequality problem.

Definition 2.5. *To any function $f : X \to \mathbb{R} \cup \{+\infty\}$ we associate the set-valued map $N^a : \operatorname{dom} f \to 2^{X^*}$ defined for any $x \in \operatorname{dom} f$ as the normal cone to the adjusted sublevel set $S^a_f(x)$ at x; i.e.,*

$$N^a(x) = \left\{x^* \in X^* : \langle x^*, y - x\rangle \leq 0 \ \forall y \in S^a_f(x)\right\}.$$

Note that $S^<_{f(x)} \subseteq S^a_f(x) \subseteq S_{f(x)}$ implies that $N(x) \subseteq N^a(x) \subseteq N^<(x)$ for all $x \in \operatorname{dom} f$.

3. Properties of the normal operator. In this section we investigate properties of the normal operator N^a for quasi-convex functions: equivalent definition, nonemptiness, quasi-monotonicity and cone upper-semicontinuity are considered.

We will give for quasi-convex functions an equivalent definition of N^a which clearly suggests that this operator corresponds to a refined version of the operator N. Let us first define for any $x \in \operatorname{dom} f$ the *extended normal cone* of f at x as follows. For every $x \in \operatorname{dom} f \setminus \arg\min f$ we set

$$EN(x) = \{x^* \in X^* \ : \ \langle x^*, y\rangle \le \langle x^*, z\rangle, \ \forall\, y \in S^<_{f(x)}, \ \forall\, z \in \overline{B}(x, \rho_x)\},$$

while for $x \in \arg\min f$ we set $EN(x) = \{0\}$. Note that $EN(x)$ is a closed convex cone. In fact, for $x \in \operatorname{dom} f \setminus \arg\min f$ it is the normal cone at x to the set $S^<_{f(x)} + \overline{B}(0, \rho_x)$ or, equivalently, to its closure $\overline{B}(S^<_{f(x)}, \rho_x)$. In addition, x^* is an element of $EN(x)$ if and only if for all $y \in S^<_{f(x)}$ and all $v \in \overline{B}(0,1)$ one has $\langle x^*, x - y\rangle \ge -\rho_x \langle x^*, v\rangle$. Consequently, for any $x \in \operatorname{dom} f \setminus \arg\min f$, $EN(x)$ admits the following equivalent definition:

$$x^* \in EN(x) \Longleftrightarrow \langle x^*, x - y\rangle \ge \rho_x \|x^*\|, \ \forall\, y \in S^<_{f(x)}. \tag{3.1}$$

Proposition 3.1. *Let f be quasi-convex. Then for each $x \in \operatorname{dom} f$,*

$$N^a(x) = N(x) + EN(x) = \operatorname{co}\left(N(x) \cup EN(x)\right). \tag{3.2}$$

Before proving Proposition 3.1, let us state the following well-known basic lemma.

Lemma 3.2. *Let A, B be convex subsets of X. If $A \cap \operatorname{int} B \neq \emptyset$, then $\overline{A \cap B} = \overline{A} \cap \overline{B}$.*

Proof (of Proposition 3.1). If $x \in \arg\min f$, the equality is obvious. Assume that $x \notin \arg\min f$. We consider two cases. If $\rho_x = 0$, then $S^a_f(x) = \overline{S^<_{f(x)}} \cap S_{f(x)}$, and thus $S^<_{f(x)} \subseteq S^a_f(x) \subseteq \overline{S^<_{f(x)}}$. It follows that $N^a(x) = N^<(x) = EN(x)$. Since $N(x) \subseteq N^<(x)$, we have $N\,(x) + EN\,(x) = EN\,(x) = N^a(x)$.

Now assume that $\rho_x > 0$. Obviously, $N^a(x)$ is the normal cone to the set $\overline{S_{f(x)} \cap \overline{B}(S^<_{f(x)}, \rho_x)}$ at x. However,

$$S_{f(x)} \cap \operatorname{int} \overline{B}\left(S^<_{f(x)}, \rho_x\right) \supseteq S^<_{f(x)} \neq \emptyset; \tag{3.3}$$

hence by Lemma 3.2,

$$\overline{S_{f(x)} \cap \overline{B}\left(S^<_{f(x)}, \rho_x\right)} = \overline{S_{f(x)}} \cap \overline{B}\left(S^<_{f(x)}, \rho_x\right).$$

Therefore, $N^a(x)$ is the normal cone to $\overline{S_{f(x)}} \cap \overline{B}(S^<_{f(x)}, \rho_x)$ at x. From (3.3) and using [1, Thm. 4.1.16] we deduce that $N^a(x) = N(x) + EN\,(x)$. The second equality is obvious. □

Let us set $S^*(0,1) = \{x^* \in X^* : \|x^*\| = 1\}$.

Proposition 3.3. *Let $f : X \to \mathbb{R} \cup \{+\infty\}$ be any function. Then*

(i) *$EN \cap S^*(0,1)$ is cyclically monotone on any nonempty subset*

$$S^=_a = \{x \in X \ : \ f(x) = a\},$$

(ii) *N^a is cyclically quasi-monotone.*

Proof. (i) Let us consider $x_1, x_2, \ldots, x_n \in S_a^=$. We assume that $x_i \notin \arg\min f$ since otherwise $EN\,(x_i) \cap S^*(0,1)$ is empty. Set $x_{n+1} = x_1$ and take $x_i^* \in EN(x_i) \cap S^*(0,1)$, $i = 1, 2, \ldots, n$. According to (3.1), for any $y \in S_{f(x_i)}^<$, $\langle x_i^*, x_i - y\rangle \geq \rho_{x_i}$. This yields

$$\|x_{i+1} - y\| \geq \langle x_i^*, x_{i+1} - x_i\rangle + \langle x_i^*, x_i - y\rangle \geq \rho_{x_i} + \langle x_i^*, x_{i+1} - x_i\rangle,$$

from which, remembering that $f\,(x_i) = a$, we get

$$\forall i = 1, 2, \ldots, n, \quad \rho_{x_{i+1}} = d(x_{i+1}, S_{f(x_i)}^<) \geq \rho_{x_i} + \langle x_i^*, x_{i+1} - x_i\rangle.$$

Adding the inequalities for all i's we obtain

$$\sum_{i=1}^{n} \langle x_i^*, x_{i+1} - x_i\rangle \leq 0;$$

i.e., $EN \cap S^*(0,1)$ is cyclically monotone on $S_a^=$.

(ii) If N^a is not cyclically quasi-monotone, then there exist $x_i \in \operatorname{dom}(f)$, $x_i^* \in N^a(x_i)$, $i = 1, 2, \ldots, n$ such that

$$\langle x_i^*, x_{i+1} - x_i\rangle > 0, \quad i = 1, 2, \ldots, n, \tag{3.4}$$

where $x_{n+1} = x_1$.

Since $N^a(x_i) \subseteq N^<(x_i)$, (3.4) implies that for all $i = 1, 2, \ldots, n$, $f\,(x_i) \leq f\,(x_{i+1})$. Consequently, $f\,(x_1) = f\,(x_2) = \cdots = f\,(x_n)$. This means that $S_{f(x_i)}^<$ is the same for all i. We denote this set by A. From (3.4) and $x_i^* \in N^a(x_i)$ it also follows that $x_{i+1} \notin S_{f(x_i)} \cap \overline{B}\,(A, \rho_{x_i})$. Since $f(x_{i+1}) = f(x_i)$, we have $x_{i+1} \in S_{f(x_i)}$. Hence, $x_{i+1} \notin \overline{B}\,(A, \rho_{x_i})$ for all $i = 1, 2, \ldots, n$. It follows that $\rho_{x_{i+1}} > \rho_{x_i}$ for all $i = 1, 2, \ldots, n$. This easily leads to $\rho_{x_{n+1}} > \rho_{x_1}$, a contradiction. □

According to the preceding proposition, the operator N^a is always quasi-monotone. Just as the so-called *quasi-convex subdifferential* [7], N^a has the property to characterize the quasi-convexity of the associated function not by its quasi-monotonicity, but by its nonemptiness on a dense subset of $\operatorname{dom}(f)$.

PROPOSITION 3.4. *Let $f : X \to \mathbb{R} \cup \{+\infty\}$ be a lower semicontinuous function. Suppose that either f is radially continuous or* $\operatorname{dom}(f)$ *is convex and* $\operatorname{int}(S_a) \neq \emptyset$ *for all $a > \inf_X f$. Then*

(i) *If $N^a(x) \setminus \{0\}$ is nonempty on a dense subset of* $\operatorname{dom}(f) \setminus \arg\min f$*, then f is quasi-convex.*

(ii) *If f is quasi-convex, then $N^a(x) \setminus \{0\} \neq \emptyset$ for all $x \in$* $\operatorname{dom}(f) \setminus \arg\min f$.

(iii) *f is quasi-convex if and only if* $\operatorname{dom}(N^a \setminus \{0\})$ *is dense in* $\operatorname{dom}(f) \setminus \arg\min f$.

Proof. (i) Looking closely into the proof of Proposition 11 of [7] one can observe that it has been shown that, under the assumptions of the present proposition, the function f is quasi-convex provided that the domain of $N^< \setminus \{0\}$ is dense in $\operatorname{dom} f \setminus \arg\min f$. Since $N^a(x) \setminus \{0\} \subseteq N^<(x) \setminus \{0\}$, the assertion follows.

(ii) For every $x \in \operatorname{dom}(f) \setminus \arg\min f$ one has $x \notin S_{f(x)}^<$. It is known that a quasi-convex, lower semicontinuous, and radially continuous function is continuous [7, Prop. 9]. Thus, our assumptions imply that $\operatorname{int}(S_{f(x)}^<) \neq \emptyset$. Hence there exists $x^* \in X^* \setminus \{0\}$ such that

$$\forall\, y \in S_{f(x)}^<,\ \forall\, z \in \overline{B}(x, \rho_x), \quad \langle x^*, y\rangle \leq \langle x^*, z\rangle.$$

Therefore, $x^* \in EN(x)$ and from Proposition 3.1 it follows that $N^a(x) \setminus \{0\} \neq \emptyset$. Finally, assertion (iii) resumes the previous ones. □

PROPOSITION 3.5. *Let f be quasi-convex and such that* $\mathrm{int} S_a \neq \emptyset$ *for all* $a > \inf f$. *If f is lower semicontinuous at $x \in \mathrm{dom}\,(f) \setminus \arg\min f$, then N^a is norm-to-w* cone upper-semicontinuous at x.*

Before proving Proposition 3.5 we establish the following lemma. For any set $U \subseteq X$, $N^<(U)$ denotes as usual the set $\cup_{x \in U} N^<(x)$.

LEMMA 3.6. *Let f be quasi-convex and such that* $\mathrm{int} S_a \neq \emptyset$ *for all* $a > \inf f$. *If f is lower semicontinuous at $x \in \mathrm{dom}\, f \setminus \arg\min f$, then there exists a neighborhood U of x and an element $z \in X \setminus \{0\}$ such that the set $N^<(U) \cap A$, with $A = \{x^* \in X^* : \langle x^*, z\rangle = 1\}$, is a bounded base for the cone $N^<(U)$.*

Proof. Choose $y_0 \in X$ and $\delta > 0$ such that $y_0 \in \mathrm{int}\, S^<_{f(x)-\delta}$. There exists $\varepsilon > 0$ such that

$$\forall z \in B\,(0,1), \quad f\,(y_0 + \varepsilon z) < f(x) - \delta.$$

Since f is lower semicontinuous at x, we can choose $\varepsilon_1 > 0$ such that for every $u \in x + \varepsilon_1 B\,(0,1)$, $f\,(u) > f\,(x) - \delta$. Thus

$$\forall u \in x + \varepsilon_1 B\,(0,1), \quad y_0 + \varepsilon B\,(0,1) \subseteq S^<_{f(u)}. \tag{3.5}$$

Set $\varepsilon_2 = \min\{\varepsilon/2, \varepsilon_1\}$, $U = x + \varepsilon_2 B\,(0,1)$. For every $u \in U$, from (3.5) we deduce that $f\,(y_0 + \varepsilon w) < f\,(u)$ for all $w \in B\,(0,1)$ and thus for every $x^* \in N^<(u)$ we obtain the following:

$$\forall w \in B\,(0,1), \quad \langle x^*, y_0 + \varepsilon w - u\rangle \leq 0.$$

It follows that

$$\begin{aligned} \varepsilon \|x^*\| &= \sup_{w \in B(0,1)} \langle x^*, \varepsilon w\rangle \leq \langle x^*, u - y_0\rangle \\ &= \langle x^*, x - y_0\rangle + \langle x^*, u - x\rangle \leq \langle x^*, x - y_0\rangle + \|x^*\| \frac{\varepsilon}{2} \end{aligned}$$

Thus,

$$\forall u \in U,\ \forall x^* \in N^<(u), \quad \langle x^*, x - y_0\rangle \geq (\varepsilon/2)\,\|x^*\|. \tag{3.6}$$

In particular, $\langle x^*, x - y_0\rangle > 0$ whenever $x^* \in N^<(u) \setminus \{0\}$. Now set $A = \{x^* \in X^* : \langle x^*, x - y_0\rangle = 1\}$. Obviously, for every $u \in U$ and $x^* \in N^<(u) \cap A$, one has $\|x^*\| \leq 2/\varepsilon$; i.e., $N^<(U) \cap A$ is bounded. □

Proof of Proposition 3.5. Let U and A be the neighborhood and hyperplane given by Lemma 3.6. Define $C\,(u) = N^a\,(u) \cap A$, $u \in U$. Obviously, $C\,(u)$ is a convex, w*-compact base of $N^a\,(u)$. We have to show that C is norm-to-w* upper semicontinuous at x. Define $D\,(u) = (N(u) \cup EN(u)) \cap A$, $u \in U$. We first show that D is norm-to-w* upper semicontinuous. According to [10, Prop. 1.2.23] it is sufficient to show that if $(x_i, x_i^*)_{i \in I}$ is a net in $\mathrm{gr}\, D$ such that $x_i \to x$ in norm and $x_i^* \overset{w^*}{\rightharpoonup} x^*$, then $x^* \in D\,(x)$. Since obviously $x^* \in A$, we have to show that $x^* \in EN(x) \cup N\,(x)$. Since $x_i^* \in EN\,(x_i) \cup N\,(x_i)$ we may consider, without loss of generality, that either $x_i^* \in N\,(x_i)$ for all $i \in I$ or $x_i^* \in EN\,(x_i)$ for all $i \in I$.

Suppose first that $x_i^* \in N\,(x_i)$. For every $y \in S^<_{f(x)}$, there exists i_0 such that for all $i > i_0$, $f\,(y) < f\,(x_i)$. Thus, $\langle x_i^*, x_i - y\rangle \geq 0$. Taking into account that x_i^* are bounded as they belong to $N^<(U) \cap A$, we obtain at the limit $\langle x^*, x - y\rangle \geq 0$.

This means that $x^* \in N^<(x)$. If x is not a local minimum, then $\rho_x = 0$; hence $N^<(x) = EN(x)$ so that $x^* \in EN(x) \cup N(x)$ and we are done. If x is a local minimum, then for i sufficiently large $f(x_i) \geq f(x)$. Hence, for every $y \in S_{f(x)}$ we have $y \in S_{f(x_i)}$. Consequently, $\langle x_i^*, x_i - y\rangle \geq 0$ thus implying $\langle x^*, x - y\rangle \geq 0$ for all $y \in S_{f(x)}$. It follows that $x^* \in N(x) \subseteq EN(x) \cup N(x)$.

Now suppose that $x_i^* \in EN(x_i)$. Without loss of generality, we may assume that for all i's we have either $f(x_i) > f(x)$ or $f(x_i) \leq f(x)$. If $f(x_i) > f(x)$ holds, then $S_{f(x)} \subseteq S^<_{f(x_i)}$. Thus,

$$\forall y \in S_{f(x)}, \quad \langle x_i^*, x_i - y\rangle \geq 0$$

and at the limit $\langle x^*, x - y\rangle \geq 0$ for all $y \in S_{f(x)}$, which shows that $x^* \in N(x)$. If on the contrary $f(x_i) \leq f(x)$ holds, then $S^<_{f(x_i)} \subseteq S^<_{f(x)}$; thus

$$\liminf \rho_{x_i} = \liminf \operatorname{dist}\left(x_i, S^<_{f(x_i)}\right) \geq \lim \operatorname{dist}\left(x_i, S^<_{f(x)}\right) = \rho_x. \tag{3.7}$$

Now for each $y \in S^<_{f(x)}$ there exists $i_0 \in I$ such that for all $i > i_0$, $f(x_i) > f(y)$. Thus, $y \in S^<_{f(x_i)}$ and

$$\langle x_i^*, x_i - y\rangle \geq \rho_{x_i} \|x_i^*\|.$$

Using (3.7) and lower semicontinuity of $\|\cdot\|$ at x^*, we find

$$\forall y \in S^<_{f(x)}, \quad \langle x^*, x - y\rangle \geq \rho_x \|x^*\|,$$

which means that $x^* \in EN(x)$. Thus, in all cases $x^* \in EN(x) \cup N(x)$. This shows that D is norm-to-w* upper semicontinuous at x, as desired.

To show that C is norm-to-w* upper semicontinuous at x, it is again sufficient to show that if $(x_i, x_i^*)_{i\in I}$ is a net in $\operatorname{gr} C$ such that $x_i \to x$ in norm and $x_i^* \overset{w^*}{\rightharpoonup} x^*$, then $x^* \in C(x)$. Note that in view of Proposition 3.1,

$$C(x_i) = \operatorname{co}\left((N(x_i) \cap A) \cup (EN(x_i) \cap A)\right);$$

hence, each x_i^* can be written in the form $x_i^* = \lambda_i y_i^* + (1 - \lambda_i) z_i^*$, where $y_i^* \in N(x_i) \cap A$, $z_i^* \in EN(x_i) \cap A$, and $\lambda_i \in [0,1]$. Since y_i^* and z_i^* are bounded (as they belong to $N^<(U) \cap A$), by considering subnets if necessary we may assume that $y_i^* \overset{w^*}{\rightharpoonup} y^*$, $z_i^* \overset{w^*}{\rightharpoonup} z^*$, and $\lambda_i \to \lambda$. By the norm-to-w* upper semicontinuity of D, we know that $y^*, z^* \in D(x)$; hence, $x^* \in C(x)$ and C is norm-to-w* upper semicontinuous at x. ∎

4. Quasi-convex programming. In [4] an existence result for quasi-monotone variational inequality has been proved under weak assumptions, in particular without compactness nor hypothesis on inner points. Taking advantage of the good properties of the normal operator N^a, our aim in this section is to obtain an existence result for the minimization of a quasi-convex function over a convex set through the study of an associated variational inequality.

Given $K \subseteq X$ and an operator $T : K \to 2^{X^*}$ we denote by $S_{str}(T,K)$ the set of *strong solutions* of the Stampacchia variational inequality

$$x_0 \in S_{str}(T,K) \Longleftrightarrow x_0 \in K \text{ and } \exists\, x_0^* \in T(x_0) \ : \ \forall\, x \in K, \ \langle x_0^*, x - x_0\rangle \geq 0.$$

Given $K \subseteq X$ we set $K^\perp = \{x^* \in X^* \ : \ \forall\, x, y \in K,\ \langle x^*, x\rangle = \langle x^*, y\rangle\}$. If we define aff$K$ as the affine hull of K, i.e.,

$$\mathrm{aff}K = \left\{ \sum_{i=1}^n \lambda_i x_i \ : \ \sum_{i=1}^n \lambda_i = 1,\ x_i \in K,\ i = 1, \ldots, n \right\}$$

and $\overline{\mathrm{aff}}K$ the closure of aff$K$, then it is easy to see that $K^\perp = \{0\}$ if and only if $\overline{\mathrm{aff}}K = X$. In optimization problems one can often assume that $K^\perp = \{0\}$ with no loss of generality. It is enough to translate K so that $0 \in K$ and then restrict the problem to the subspace $X_1 = \overline{\mathrm{aff}}K$; then condition $K^\perp = \{0\}$ is fulfilled.

PROPOSITION 4.1. *Let $f : X \to \mathbb{R} \cup \{+\infty\}$ be a quasi-convex function, radially upper semicontinuous on* dom (f), *and $K \subseteq$* dom (f) *be a convex set such that $K^\perp = \{0\}$. Assume that either*

(i) $x_0 \in S_{str}(N^< \setminus \{0\}, K)$, *or*

(ii) $x_0 \in S_{str}(N^a \setminus \{0\}, K)$.

Then for all $x \in K$, $f(x_0) \le f(x)$.

Proof. (i) By assumption, there exists $x_0^* \in N^<(x_0) \setminus \{0\}$ such that for all $x \in K$, $\langle x_0^*, x - x_0\rangle \ge 0$. Since $x_0^* \notin K^\perp$, there exists $y \in K$ such that $\langle x_0^*, y - x_0\rangle \neq 0$; thus $\langle x_0^*, y - x_0\rangle > 0$. Fix such a y and for any $x \in K$ and any $t \in\,]0, 1[$ define $x_t = (1 - t)x + ty$. Then

$$\langle x_0^*, x_t - x_0\rangle = (1 - t)\langle x_0^*, x - x_0\rangle + t\langle x_0^*, y - x_0\rangle > 0.$$

Since $x_0^* \in N^<(x_0)$, this gives $f(x_t) \ge f(x_0)$ and by radial upper semicontinuity $f(x) \ge f(x_0)$.

(ii) This is an immediate consequence of (i) since $N^a(x) \subseteq N^<(x)$ for all x. □

We will use a very weak kind of continuity for multivalued operators (cf. [9]): Given a convex subset $K \subseteq X$ and an operator $T : K \to 2^{X^*} \setminus \{\emptyset\}$, T is called *upper sign-continuous* on K if for any $x, y \in K$,

$$\forall\, t \in\,]0, 1[\,, \qquad \inf_{x_t^* \in T(x_t)} \langle x_t^*, y - x\rangle \ge 0 \Longrightarrow \sup_{x^* \in T(x)} \langle x^*, y - x\rangle \ge 0,$$

where $x_t = (1 - t)x + ty$. If, for example, the restriction of T to every line segment of K is upper semicontinuous with respect to the w*-topology in X^*, then T is upper sign-continuous.

Let us recall the following existence result for the Stampacchia variational inequality [4].

PROPOSITION 4.2. *Let K be a convex subset of X such that $K \cap \overline{B}(0, n)$ is weakly compact for every $n \in \mathbb{N}$. Let further $T : K \to 2^{X^*} \setminus \{\emptyset\}$ be a quasi-monotone operator such that the following coercivity condition holds:*

$$\begin{array}{ll} (4.1) & \exists\, n \in \mathbb{N},\ \forall\, x \in K \setminus \overline{B}(0, n),\ \exists\, y \in K \textit{ with } \|y\| < \|x\| \\ & \textit{such that } \forall\, x^* \in T(x), \langle x^*, x - y\rangle \ge 0. \end{array}$$

Suppose moreover that for every $x \in K$ there exist a neighborhood V_x of x and an upper sign-continuous operator $S_x : V_x \cap K \to 2^{X^*} \setminus \{\emptyset\}$ with convex, w*-compact values satisfying $S_x(y) \subseteq T(y)$ for all $y \in V_x \cap K$. Then $S_{str}(T, K) \neq \emptyset$.

Note that condition (4.1), which has been previously used in Isac [11], is automatically satisfied if K is bounded. We now apply the above results to optimization.

THEOREM 4.3. *Let $f : X \to \mathbb{R} \cup \{+\infty\}$ be a lower semicontinuous quasi-convex function, radially continuous on* dom (f). *Assume that for every $\lambda > \inf_X f$,*

$\mathrm{int}(S_\lambda) \neq \emptyset$. *Let* $K \subseteq \mathrm{dom}\,(f)$ *be convex with* $K^\perp = \{0\}$ *and such that* $K \cap \overline{B}(0,n)$ *is weakly compact for every* $n \in \mathbb{N}$.

If condition (4.1) *holds with* $T = N^a$*, then there exists* $x_0 \in K$ *such that*

$$\forall\, x \in K, \quad f(x) \geq f(x_0).$$

Proof. If $\arg\min f \cap K \neq \emptyset$, we have nothing to prove. Suppose that $\arg\min f \cap K = \emptyset$. According to Proposition 3.3, N^a is quasi-monotone. Further, according to Proposition 3.5, it is norm-to-w* cone upper-semicontinuous on K. Thus, all assumptions of Proposition 4.2 hold for the operator $N^a \setminus \{0\}$, and thus $S_{str}\,(N^a \setminus \{0\}\,, K) \neq \emptyset$. Finally, using Proposition 4.1 we infer that f has a global minimum on K. □

COROLLARY 4.4. *Make the assumptions on* f *and* K *as in Theorem* 4.3. *Assume that there exists* $n \in \mathbb{N}$ *such that for all* $x \in K$, $\|x\| > n$, *there exists* $y \in K$, $\|y\| < \|x\|$, *such that* $f\,(y) < f\,(x)$. *Then there exists* $x_0 \in K$ *such that*

$$\forall\, x \in K, \quad f(x) \geq f(x_0).$$

Proof. If $f\,(y) < f(x)$, then for every $x^* \in N^a(x) \subseteq N^<(x)$, $\langle x^*, y - x\rangle \leq 0$. Hence, coercivity condition (4.1) with $T = N^a$ holds. The corollary follows from Theorem 4.3. □

Acknowledgment. This work was prepared while the second author was visiting the Mathematics Department of the University of Perpignan, France. The author wishes to thank the Department for its hospitality.

REFERENCES

[1] J.-P. AUBIN AND I. EKELAND, *Applied Nonlinear Analysis*, Wiley-Interscience, New York, 1984.

[2] D. AUSSEL, J.-N. CORVELLEC, AND M. LASSONDE, *Subdifferential characterization of quasiconvexity and convexity*, J. Convex. Anal., 1 (1994), pp. 195–201.

[3] D. AUSSEL AND A. DANIILIDIS, *Normal characterization of the main classes of quasiconvex functions*, Set-Valued Anal., 8 (2000), pp. 219–236.

[4] D. AUSSEL AND N. HADJISAVVAS, *On quasimonotone variational inequalities*, J. Optim. Theory Appl., 121 (2004), pp. 445–450.

[5] J. BORDE AND J.-P. CROUZEIX, *Continuity properties of the normal cone to the level sets of a quasiconvex function*, J. Optim. Theory Appl., 66 (1990), pp. 415–429.

[6] A. DANIILIDIS AND N. HADJISAVVAS, *On the subdifferentials of quasiconvex and pseudoconvex functions and cyclic monotonicity*, J. Math. Anal. Appl., 237 (1999), pp. 30–42.

[7] A. DANIILIDIS, N. HADJISAVVAS, AND J.-E. MARTÍNEZ-LEGAZ, *An appropriate subdifferential for quasiconvex functions*, SIAM J. Optim., 12 (2001), pp. 407–420.

[8] A. EBERHARD AND J.-P. CROUZEIX, *Integration of a Normal Cone Relation Generated by the Level Sets of Pseudo-convex Functions*, preprint, 2003.

[9] N. HADJISAVVAS, *Continuity and maximality properties of pseudomonotone operators*, J. Convex Anal., 10 (2003), pp. 465–475.

[10] S. HU AND N. S. PAPAGEORGIOU, *Handbook of Multivalued Analysis, Vol.* I: *Theory*, Kluwer Academic Publishers, Norwell, MA, 1997.

[11] G. ISAC, *A generalization of Karamardian's condition in complementarity theory*, Nonlinear Anal. Forum, 4 (1999), pp. 49–63.

[12] D. T. LUC, *Characterizations of quasiconvex functions*, Bull. Austral. Math. Soc., 48 (1993), pp. 393–405.

[13] D. T. LUC AND J.-P. PENOT, *Convergence of asymptotic directions*, Trans. Amer. Math. Soc., 353 (2001), pp. 4095–4121.

SIAM J. Optim.
Vol. 16, No. 2, pp. 368–379

MINIMIZING WITHIN CONVEX BODIES USING A CONVEX HULL METHOD*

THOMAS LACHAND-ROBERT[†] AND ÉDOUARD OUDET[†]

Abstract. We present numerical methods to solve optimization problems on the space of convex functions or among convex bodies. Hence convexity is a constraint on the admissible objects, whereas the functionals are not required to be convex. To deal with this, our method mixes geometrical and numerical algorithms.

We give several applications arising from classical problems in geometry and analysis: Alexandrov's problem of finding a convex body of prescribed surface function; Cheeger's problem of a subdomain minimizing the ratio surface area on volume; Newton's problem of the body of minimal resistance.

In particular for the latter application, the minimizers are still unknown, except in some particular classes. We give approximate solutions better than the theoretical known ones, hence demonstrating that the minimizers do not belong to these classes.

Key words. optimization, convex functions, numerical schemes, convex bodies, Newton's problem of the body of minimal resistance, Alexandrov, Cheeger

AMS subject classifications. 46N10, 52A40, 52A41

DOI. 10.1137/040608039

1. Introduction.

In this paper, we present numerical methods to solve optimization problems among convex bodies or convex functions. Several problems of this kind appear in geometry, calculus, applied mathematics, etc. As applications, we present some of them, together with our corresponding numerical results.

Dealing with convex bodies or convex functions is usually considered easier in optimization theory. Unfortunately, this is not true when *the optimization space itself* is (a subset of) the set of convex functions or bodies. As an example, consider the following minimization problem, where $M > 0$ is a given parameter, Ω is a regular bounded convex subset of $\mathbb{R}^n$, and g is a continuous function on $\Omega \times \mathbb{R} \times \mathbb{R}^n$:

$$\inf_{u \in C_M} \int_\Omega g(x, u(x), \nabla u(x))\, dx, \tag{1}$$
$$\text{where } C_M = \{u : \Omega \to [-M, 0], \quad u \text{ convex }\}.$$

Without the convexity constraint, this problem is usually handled in a numerical way by considering $g_2'(x, u(x), \nabla u(x)) = \operatorname{div} g_3'(x, u(x), \nabla u(x))$, the associated Euler equation. Such an equation is discretized and solved on a mesh defined on Ω (or, more precisely, a sequence of meshes, in order to achieve a given precision), using, for instance, finite element methods.

1.1. Dealing with the convexity constraint.

The classical numerical methods do not work at all with our problem:

*Received by the editors May 12, 2004; accepted for publication (in revised form) September 7, 2004; published electronically October 7, 2005.

http://www.siam.org/journals/siopt/16-2/60803.html

[†] Laboratoire de mathématiques, Université de Savoie, Campus scientifique, 73376 Le Bourget-du-lac, France (Thomas.Lachand-Robert@univ-savoie.fr, http://www.lama.univ-savoie.fr/∼lachand, Edouard.Oudet@univ-savoie.fr, http://www.lama.univ-savoie.fr/∼oudet).

1. The convexity constraint prevents us from using an Euler equation. In fact, just stating a correct Euler equation for this sort of problem is a difficult task [12, 20, 8]. Discretizing the corresponding equation, then, is rather difficult.
2. The set C_M of admissible functions, considered as a subset of a Sobolev space such as $H^1_{\mathrm{loc}}(\Omega)$, is compact [5]. This makes it easy to prove the existence of a solution of (1) without any other assumption on g. But this also implies that C_M is a very small subset of the function's space, with an empty interior. Therefore most numerical approximations of a candidate function u are not convex. Evaluating the functional on those approximations is likely to yield a value much smaller than the sought minimum.
3. The natural way to evade the previous difficulty is to use only convex approximations. For instance, on a triangular mesh of Ω, it is rather easy to characterize those P1-functions (that is, continuous and affine by parts functions) which are convex. Unfortunately, such an approximation introduces a geometric bias from the mesh. The set of convex functions that are limits of this sort of approximation is much smaller than C_M [13].
4. Penalization processes are other ways to deal with this difficulty. But finding a good penalization is not easy, and this usually yields very slow algorithms, which in this particular case are not very convincing. This yields approximation difficulties similar to those given in 2 above.

A first solution for this kind of numerical problem was presented in [10], and an improved version is given in [9]. However, the algorithms given in these references are not very fast, since they deal with a large number of constraints, and do not apply to those problems where local minimizers exist. The latter are common in the applications since there is no need for the functional itself to be convex to prove the existence of a solution of (1): the mere compacity of C_M, together with the continuity of the functional on an appropriate functions space, is sufficient.

1.2. A mixed-type algorithm. Our main idea to handle numerically (1) is to mix geometrical and numerical algorithms. It is standard that any convex body (or, equivalently, the graph of any convex function) can be described as an intersection of half-spaces or as a convex hull of points. Our discretization consists of considering only a finite number of half-spaces or a finite number of points. (This is not equivalent, and choosing either mode is part of the method.) Reconstructing the convex body is a standard algorithm, and computing the value of the functional, then, is straightforward. Obviously, the convex hull algorithm used implies an additional cost that cannot be neglected. On the other hand, this method makes it easy to deal with additional constraints, such as the fact that functions get values in $[0, M]$. We also show that it is possible to compute the derivative of the functional. Hence we may use gradient methods for minimization.

Note that since this always deals with convex bodies, we are guaranteed that the evaluations of the functional are not smaller than the sought minimum, up to numerical errors. Because the approximation process is valid for any convex body, we can ensure that all minimizers can be approximated arbitrary closely.

The detailed presentation of the method requires us to explain how the half-spaces or points are moved, whether or not their number is increased, and which information on the specific problem is useful for this. We present quite different examples in our applications, in order to pinpoint the corresponding difficulties. Whenever the minimizer of the functional is not unique, gradient methods may get stuck in local minima. We present a “genetic algorithm” to deal with these, too.

In this paper, we concentrate on the three-dimensional settings. The two-dimensional case is much easier, and convex sets in the plane can be parametrized in a number of very simple ways. Even though our methods could be applied to dimensions $n \geq 4$, the convex hull computation may become too expensive.

1.3. Generalized problem. This algorithm's design does not involve any mesh or interpolation process. As an important consequence, we are not limited to convex functions but may also consider convex bodies. This allows us to study problems such as

$$\inf_{A \in \mathcal{A}} \mathcal{F}(A), \quad \text{where } \mathcal{F}(A) := \int_{\partial A} f(x, \nu_A(x), \varphi_A(x))\, d\mathcal{H}^2(x), \tag{2}$$

and $\mathcal{A}$ is a subset of the class of closed convex bodies of $\mathbb{R}^3$. We make use of the following notations:

- ∂A is the boundary of a convex body A;
- ν_A is the almost everywhere defined outer normal vector field on ∂A, with values on the sphere $\mathbf{S}^2$;
- $\varphi_A(x)$ is the signed distance from the supporting plane at x to the origin of coordinates;
- f is a continuous function $\mathbb{R}^3 \times \mathbf{S}^2 \times \mathbb{R} \to \mathbb{R}$.

Since $\varphi_A(x) = x \cdot \nu_A(x)$, the expression of the functional $\mathcal{F}$ is somehow redundant. But the particular case of functions f depending only on ν, φ is important both in applications and in the algorithm used, as we shall see.

As reported in [7], the problem (1) can be reformulated in terms of (2) whenever g depends only on its third variable. In this formulation $\mathcal{A}$ stands for the set of convex subsets of $Q_M := \Omega \times [0, M]$ containing $Q_0 = \Omega \times \{0\}$. Any convex body $A \in \mathcal{A}$ has the form

$$\mathcal{A} = \{(x', x_3) \in \Omega \times \mathbb{R}, 0 \leq x_3 \leq -u(x')\}, \text{ with } u \in C_M.$$

Therefore any $x \in \partial A \setminus Q_0$ has the form $x = (x', -u(x'))$, with $x' \in \Omega$. Then $\nu_A(x) = (\nabla u(x'), 1)/\sqrt{1 + |\nabla u(x')|^2}$, and the function f is deduced from g by the relation $f(\nu) = \nu_3 g\big(\frac{1}{\nu_3}\nu'\big)$, for every $\nu = (\nu', \nu_3) \in \mathbf{S}^2$. Several other problems with a geometrical background may also be formulated in a similar way.

Actually, the formulation (2) allows us to study any problem of the form (1). It is enough to define $f(x, \nu, \varphi) = \nu_3 g(x', -x_3, \frac{1}{\nu_3}\nu')$, taking into account that $x = (x', -u(x'))$.

On the other hand, it is much more practical in the numerical implementation to consider functions f depending only on ν, φ. This avoids numerical surface integration altogether, as explained in section 2, hence reducing greatly the computation time. With such a restriction, only some problems of the form (1) can be considered. Since

$$\varphi_A(x) = \frac{1}{\sqrt{1 + |\nabla u(x')|^2}}\,(x' \cdot \nabla u(x') + u(x')),$$

we can handle functions g depending on $\nabla u(x')$ and the aggregate $x' \cdot \nabla u(x') + u(x')$.

2. Half-spaces and discretization. For every $\nu \in \mathbf{S}^2$ and every $\varphi \geq 0$, let us define the half-space of $\mathbb{R}^3$ using the following notation:

$$[\![\nu, \varphi]\!] := \left\{x \in \mathbb{R}^3,\ x \cdot \nu \leq \varphi\right\}.$$

Lemma 1. *Let A be a convex body of $\mathbb{R}^3$. Then, for all $\varepsilon > 0$, there exists a convex polytope $P \supset A$ such that*

$$|\mathcal{F}(P) - \mathcal{F}(A)| \leq \varepsilon.$$

Proof. Let us define

$$\partial^* A := \{a \in \partial A; \nu_A(a) \text{ exists}\}.$$

Let $(X_j)_{j \in \mathbb{N}}$ be a dense sequence of points in $\partial^* A$, and consider the sequence of convex polytopes $(P_j)_{j \in \mathbb{N}}$ defined by

$$P_j := \bigcup_{k=1}^{j} [\![\nu_A(X_k), \varphi_A(X_k)]\!].$$

Clearly, $P_j \supset A$, and $\lim_{j \to \infty} P_j = A$ for the Hausdorff distance. From a classical theorem of Rockafellar [22], for any $a \in \partial^* A$, and any sequence (p_j), converging to a, with $p_j \in \partial^* P_j$ for all j, we have that $\nu_{P_j}(p_j)$ converges to $\nu_A(a)$. Since $\partial A \setminus \partial^* A$ is $\mathcal{H}^2$-negligible, we get $\mathcal{F}(P_j) \to \mathcal{F}(A)$. □

As every convex polytope is the finite intersection of half-spaces, the natural discretization of (2) is the following finite dimensional problem:

$$\min_{N, \Phi} G(N, \Phi), \tag{3}$$

$$\begin{aligned}
&\text{where } N := (\nu_1, \ldots, \nu_k) \in (\mathbf{S}^2)^k, \quad \Phi := (\varphi_1, \ldots, \varphi_k) \in \mathbb{R}^k,\\
&G(N, \Phi) := \int_{\partial P} f(x, \nu_P(x), \varphi_P(x))\, d\mathcal{H}^2(x),\\
&\text{and } P := P(N, \Phi) := \bigcap_{i=1}^{k} [\![\nu_i, \varphi_i]\!].
\end{aligned}$$

Notice that whenever f does not depend explicitly on x, $G(N, \Phi)$ can be computed as a finite sum, namely

$$G(N, \Phi) = \sum_{i=1}^{k} f(\nu_i, \varphi_i) \mathcal{H}^2(F_i), \text{ where } F_i := [\![\nu_i, \varphi_i]\!] \cap \partial P.$$

This is of primary importance in the numerical algorithms. More general functions f require the computation of integrals such as $\int_{F_i} f(x, \nu_i, \varphi_i)\, d\mathcal{H}^2(x)$, which are computationally expensive.

2.1. Computation of the derivatives. In this paragraph we compute the derivatives of G, in order to use the results in a gradient-like method. We focus on the case where f depends only on ν, φ, since this is the special case used in our actual programs. Straightforward modifications can be done to handle the general case. It suffices to change the term $\frac{\partial f}{\partial \varphi_i}(\nu_i, \varphi_i)\, \mathcal{H}^2(F_i)$ by the integral $\int_{F_i} \frac{\partial f}{\partial \varphi_i}(x, \nu_i, \varphi_i)\, d\mathcal{H}^2(x)$, and similarly with the $\mathcal{H}^1$ term.

Theorem 1. *Let $P := P(N, \Phi)$ be a convex polytope, and let $F_i = [\![\nu_i, \varphi_i]\!] \cap \partial P$. Then for almost every value of φ_i we have*

$$\begin{aligned}
\frac{\partial G}{\partial \varphi_i}(N, \Phi) = {} & \frac{\partial f}{\partial \varphi_i}(\nu_i, \varphi_i)\, \mathcal{H}^2(F_i)\\
& + \sum_{\substack{j \neq i \\ \mathcal{H}^1(F_i \cap F_j) \neq 0}} \mathcal{H}^1(F_i \cap F_j) \left(\frac{f(\nu_j, \varphi_j) - \cos\theta_{ij} f(\nu_i, \varphi_i)}{\sin\theta_{ij}} \right),
\end{aligned} \tag{4}$$

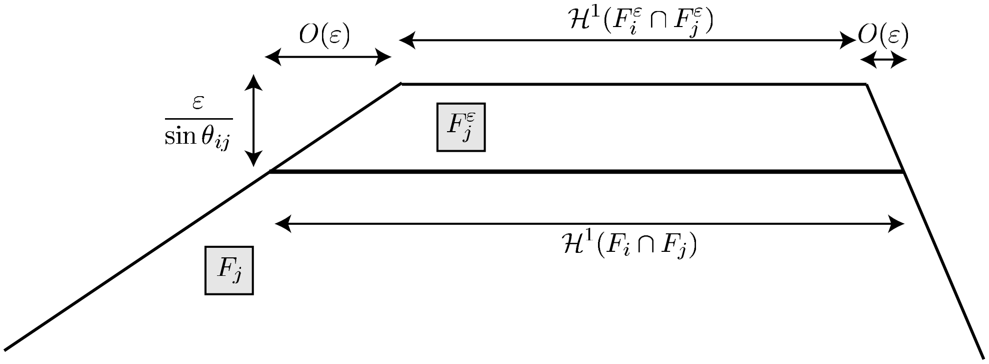

$O(\varepsilon)$
$\mathcal{H}^1(F_i^\varepsilon \cap F_j^\varepsilon)$
$O(\varepsilon)$
$\dfrac{\varepsilon}{\sin\theta_{ij}}$
F_j^ε
F_j
$\mathcal{H}^1(F_i \cap F_j)$

FIG. 1. *Variation of the surface area of F_j (pictured in the plane of F_j) for the variation $\varphi_i \to \varphi_i + \varepsilon$.*

where $\theta_{ij} \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ is defined by $\cos\theta_{ij} = |\nu_i \cdot \nu_j|$ and $\sin\theta_{ij}(\nu_i \cdot \nu_j) \geq 0$.

Proof. For any $\varepsilon \in \mathbb{R}$ consider the difference

$$\begin{aligned} G(\dots,\varphi_i+\varepsilon,\dots) - G(\dots,\varphi_i,\dots) &= f(\nu_i,\varphi_i+\varepsilon)\,\mathcal{H}^2(F_i^\varepsilon) - f(\nu_i,\varphi_i)\,\mathcal{H}^2(F_i^\varepsilon) \\ &\quad + \sum_j f(\nu_j,\varphi_j)\,(\mathcal{H}^2(F_j^\varepsilon) - \mathcal{H}^2(F_j)), \end{aligned}$$

where

$$F_j^\varepsilon = [\![\nu_j,\varphi_j]\!] \cap \partial P(\dots,\varphi_i+\varepsilon,\dots).$$

The first difference $f(\nu_i,\varphi_i+\varepsilon)\,\mathcal{H}^2(F_i^\varepsilon) - f(\nu_i,\varphi_i)\,\mathcal{H}^2(F_i^\varepsilon)$ has the following form: $\varepsilon\frac{\partial f}{\partial\varphi_i}(\nu_i,\varphi_i)\,\mathcal{H}^2(F_i) + o(\varepsilon)$.

To evaluate the remaining sum asymptotically we have to assume that the value of φ_i is such that there is no topological change in the polytope whenever φ_i becomes $\varphi_i + \varepsilon$. This is obviously true for all except a finite number of values of φ_i. We then distinguish two cases:

- $j \neq i$: $\mathcal{H}^2(F_j^\varepsilon) - \mathcal{H}^2(F_j) = \varepsilon\dfrac{\mathcal{H}^1(F_i \cap F_j)}{\sin\theta_{ij}} + o(\varepsilon)$, since the trace of F_i in the plane F_j is offset by $\varepsilon/\sin\theta_{ij}$; see Figure 1.
- $j = i$: $\mathcal{H}^2(F_i^\varepsilon) - \mathcal{H}^2(F_i) = -\varepsilon \displaystyle\sum_{\substack{j\neq i \\ \mathcal{H}^1(F_i\cap F_j)\neq 0}} \mathcal{H}^1(F_i \cap F_j)\cot\theta_{ij} + o(\varepsilon)$, since the trace of F_j in the plane F_i is offset by $\varepsilon\cot\theta_{ij}$; see Figure 2.

This completes the proof of the theorem. □

Remark 2.1. The polyhedral representation, as an intersection of half-planes, yields a technical difficulty that should not be underestimated: some of the boundary planes $\partial[\![\nu_i,\varphi_i]\!]$ are "dormant," meaning the polytope is actually included in the interior of $[\![\nu_i,\varphi_i]\!]$.

In such a situation, formula (4) effectively yields zero, since $\mathcal{H}^2(F_i) = 0 = \mathcal{H}^1(F_i \cap F_j)$.

A similar computation can be achieved for derivatives of G with respect to ν_i, with another algebraic formula as a result. However, numerical evidence proves that using a "full" gradient method has little advantage.

It turns out that it is faster and accurate enough to use only the derivatives with respect to φ_i (as detailed in the next section) and to increase if necessary the number

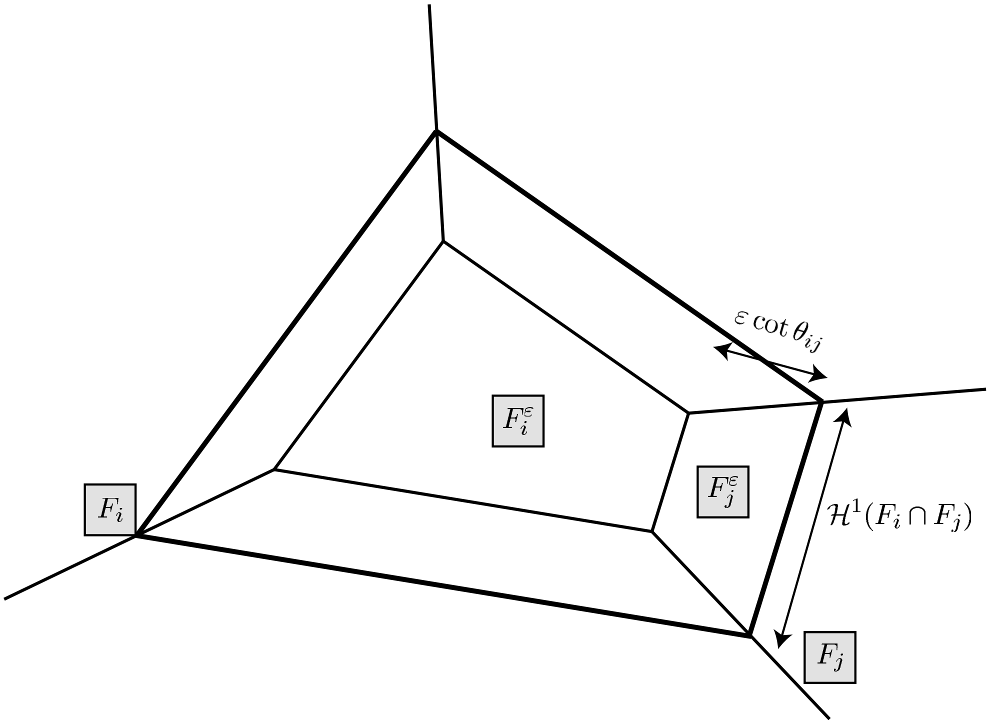


FIG. 2. *Variation of the surface area of F_i (pictured in the plane of F_i) for the variation $\varphi_i \to \varphi_i + \varepsilon$.*

of planes by considering additional half-spaces. We can profit from the "dormant" property by introducing these new half-spaces in a tangent dormant position, letting the minimization method change their position after that. This can be done in different ways, depending on the actual problem considered.

2.2. Summary of the algorithm. Thanks to Theorem 1, it is possible to apply a classical gradient algorithm to the problem (3). Let us summarize the different steps:

0. Choose one admissible polytope $P([\![\nu_1, \varphi_1^0]\!], \dots, [\![\nu_k, \varphi_k^0]\!])$, set $n = 0$.
1. Compute the geometry (vertexes, faces, . . .) of the polytope
$$P([\![\nu_1, \varphi_1^n]\!], \dots, [\![\nu_k, \varphi_k^n]\!]).$$
2. Evaluate the gradient of G with respect to the φ_j using (4). If the Euclidean norm of the gradient is small, then stop here.
3. Project the gradient into the set of admissible directions.
4. Set $\rho_n = \arg\min_{\rho>0} G(\nu_1, \dots, \nu_k, \varphi_1^n - \rho\frac{\partial G}{\partial \varphi_1}, \dots, \varphi_k^n - \rho\frac{\partial G}{\partial \varphi_k})$.
5. Define the new variables $\varphi_1^{n+1} = \varphi_1^n - \rho_n\frac{\partial G}{\partial \varphi_1}, \dots, \varphi_k^{n+1} = \varphi_k^n - \rho_n\frac{\partial G}{\partial \varphi_k}$, $n \leftarrow n+1$ and go to step 1.

Step 3, in particular, depends on the set of admissible bodies. So additional details are given in the examples hereafter. Note that it is possible in step 5 to change the number of planes by adding or removing "dormant" ones. It is also possible to change the value of ν_i whenever the ith plane is "dormant."

2.3. Application to Alexandrov's theorem. It is a classical result from Minkowski [21] that given n different vectors $\nu_1, \dots, \nu_n$ on $\mathbf{S}^2$ such that the dimension of $\operatorname{Span}\{\nu_1,\dots,\nu_n\}$ is equal to 3, and n positive real numbers $a_1,\dots,a_n$ such that $\sum_{i=1}^n a_i\nu_i = 0$, then there exists a three-dimensional convex polytope having n faces $F_1,\dots,F_n$ such that the outward normal vector to F_i equals ν_i and $\mathcal{H}^2(F_i) = a_i$. Moreover, this polytope is unique up to translations.

This result has been extended by Alexandrov [1] to arbitrary convex bodies as follows: given a positive measure μ on $\mathbf{S}^2$ satisfying $\int_{\mathbf{S}^2} y\,d\mu(y) = 0$ and $\operatorname{Span}(\operatorname{supp}\mu) = \mathbb{R}^3$, then there exists a unique convex body A, up to translations, whose surface function measure is equal to μ.

Carlier [6] proved recently that this body is the unique (up to translations) solution of the variational problem

$$\sup_{\varphi\in\Sigma} |A_\varphi|\,, \tag{5}$$

$$\text{with } \Sigma := \left\{\varphi \in C^0(\mathbf{S}^2,\mathbb{R}_+);\ \int_{\mathbf{S}^2}\varphi\,d\mu = 1\right\} \text{ and } A_\varphi := \bigcap_{\nu\in\mathbf{S}^2} [\![\nu,\varphi(\nu)]\!],$$

where $|A_\varphi|$ is the volume of A_φ. Whenever A_φ is optimal, its support function equals φ on the support of μ [6].

Now we recall that the volume of a convex body can be expressed as a boundary integral of its support function, that is,

$$|A| = \frac{1}{3}\int_{\partial A}\varphi_A(x)\,d\mathcal{H}^2(x).$$

Consequently, Alexandrov's problem can be formulated in the form (2) with $f(x,\nu,\varphi) = -\varphi$ and

$$\mathcal{A} = \left\{A \subset \mathbb{R}^3, A \text{ convex }; \varphi_A \geq 0, \int_{\mathbf{S}^2}\varphi_A\,d\mu = 1\right\}.$$

(The sign condition on φ_A is only a normalization expressing the fact that $0 \in A$.)

Whenever μ has a discrete support, namely $\mu = \sum a_i\delta_{\nu_i}$, then (5) solves Minkowski's problem for polytopes. In particular, the value of φ outside the support of μ does not matter for the maximization, and hence only the numbers $\varphi_i := \varphi(\nu_i)$ have to be considered.

Replacing an arbitrary measure μ on $\mathbf{S}^2$ by a sum of Dirac masses is also the more natural discretization of this problem. For polytopes, the set of admissible bodies has the form

$$\mathcal{A} = \Big\{P = P(N,\Phi); \varphi_i \geq 0, \sum_{i=1}^n \varphi_i a_i = 1\Big\}.$$

(Again the conditions $\varphi_i \geq 0$ are here only to limit translations ensuring that $0 \in A$. This is essential in the numerical method.) These are very simple constraints on the admissible values, so step 3 in the algorithm is an elementary projection onto $\mathbb{R}^n_+$ and a hyperplane. Hence the given algorithm can be implemented in a straightforward way.

We present an example result on Figure 3. Here we chose at random 999 vectors ν_i on $\mathbf{S}^2$ and 999 numbers a_i in $[0,1]$ uniformly; ν_{1000} and a_{1000} are determined such that the existence condition $\sum_{i=1}^{1000} a_i\nu_i = 0$ is satisfied.

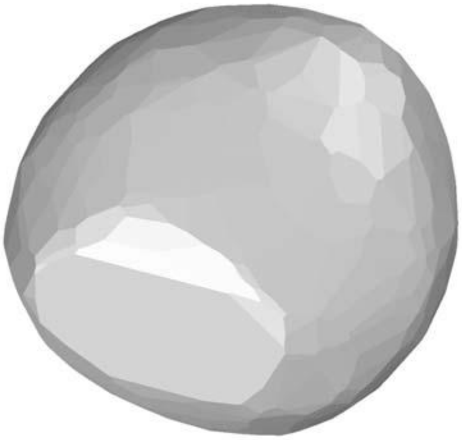


FIG. 3. *A* 1000-*faces convex polyhedron of given face areas and normals reconstructed.*

2.4. Application: Cheeger sets. Let us now present a more involved application. In 1970, Cheeger [11] proposed to study the problem

$$\inf_{X \subset M} \frac{\mathcal{H}^{n-1}(\partial X)}{\mathcal{H}^n(X)} =: h(M), \tag{6}$$

where M is an n-dimensional manifold with boundary. The resulting optimal value, known as the *Cheeger constant*, can be used to give bounds for the first eigenvalue of the Laplace–Beltrami operator on M, and even more general operators [14]. There is a number of variations and applications of this problem; see, for example, [2, 16].

The theoretical results on the problem (6) are rather sparse. It is easy to show that the infimum is usually not attained in this general formulation. On the other hand, it can be proved that minimizers exist whenever $M = \overline{\Omega}$, where $\Omega \subset \mathbb{R}^n$ is a nonempty open set. Moreover, if Ω is convex and $n = 2$, there is a unique convex optimum X which can be computed by algebraic algorithms [18]. On the other hand, if $n \geq 3$, it is not known whether the optimum set is unique or convex, even with Ω convex. However, Ω convex implies that there exists at least one convex optimum [17]. But this optimum is not known for any particular Ω except balls.

Our algorithm allows us to compute an approximation of a convex optimum when $\Omega \subset \mathbb{R}^3$ is convex. Indeed (6) can be reformulated as follows:

$$\min_{A \in \mathcal{A}} \frac{3 \int_{\partial A} d\mathcal{H}^2(x)}{\int_{\partial A} \varphi_A(x)\, d\mathcal{H}^2(x)}, \text{ with } \mathcal{A} = \{A \subset \overline{\Omega}, A \text{ convex and three-dimensional}\}.$$

So the numerator and denominator here have the form $\int_{\partial A} f(\nu_A, \varphi_A)$, and our algorithm can be applied with straightforward modifications.

A key difference with respect to our previous application is the management of the constraint $A \subset \overline{\Omega}$. The set Ω itself is approximated by a polytope (whenever necessary). The corresponding enclosing half-spaces are kept in the algorithm in order to ensure that the approximating polytopes belong to $\mathcal{A}$. For example, if Ω is a unit cube, we fix $\nu_1 = (1, 0, 0), \ldots, \nu_6 = (0, 0, -1)$ and $\varphi_1 = \cdots = \varphi_6 = 1$. Examples are given in Figure 4.

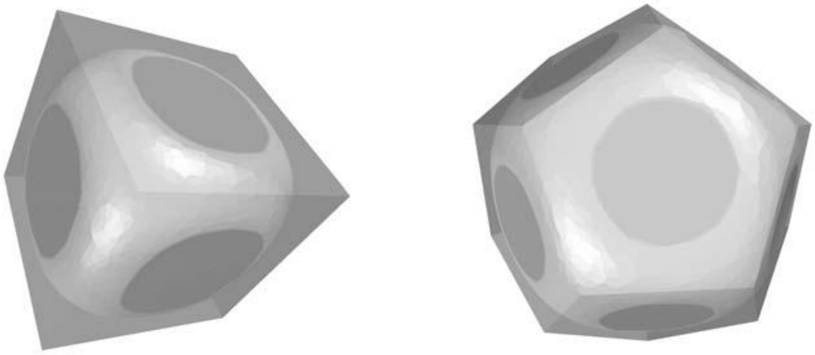


Fig. 4. *Computed solutions for the Cheeger problem in the cube and the dodecahedron.*

Table 1

Upper bounds of the ratio $\frac{h(M)\mathcal{H}^3(M)}{\mathcal{H}^2(\partial M)}$ *for the five regular polyhedrons, as given by our numerical method.*

Regular polyhedron	Ratio
tetrahedron	83%
cube	90%
octahedron	92%
dodecahedron	96%
icosahedron	97%

Computed values for all five regular polyhedron are given in Table 1. Since the Cheeger constant depends on the size of the polyhedron, we give the adimensional ratio $h(M)\mathcal{H}^3(M)/\mathcal{H}^2(\partial M)$ in the table. These are actually upper bounds since the numerical methods give only an approximation of the optimal subset X.

This approach allows us to handle any problem with constraints of the form

$$Q_0 \subset A \subset Q_1, \tag{7}$$

assuming that Q_1 is convex. (For Q_0 it is not a restriction to assume it is convex.) Other examples of problems of this kind come from mathematical economy; see the references in [9] and also in [4].

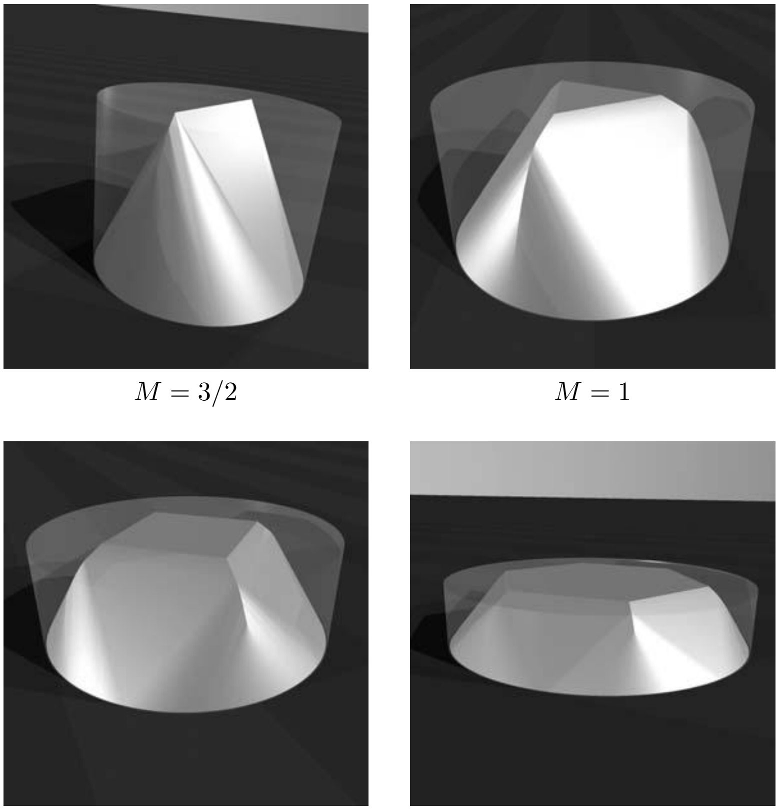
3. Newton's problem of the body of minimal resistance.

The problem of the body of minimal resistance has been settled by Newton in his *Principia*: given a body progressing at constant speed in a fluid, what shape should it be given in order to minimize its resistance? Expressed in its more classical way, this can be formulated as the following optimization problem:

$$\min_{\substack{u:\Omega\to[-M,0]\\ u \text{ convex}}} \int_\Omega \frac{dx}{1+|\nabla u|^2}, \tag{8}$$

where $M > 0$ is a given parameter and Ω is the unit disk of $\mathbb{R}^2$. There are a lot of variants from this formulation and a huge amount of literature on this problem; see [5, 19] and their references.

Newton considered only radial solutions of this problem, and his solution was already considered surprising. But it has been proved in [3] that the solutions of (8) are not radially symmetric. Unfortunately, it has been impossible until now to



$M = 3/2$ $M = 1$

$M = 7/10$ $M = 4/10$

Fig. 5. *Computed solutions of Newton's problem of the body of minimal resistance.*

describe more precisely the minimizers. Some theoretical results suggests that they should be developable in a sense given in [19]: a developable body being the convex hull of the unit circle in the plane $x_3 = 0$ and a convex set in the plane $x_3 = -M$.

So in this application, we are considering a problem of the form (1), with $g(x, u, p) = 1/(1 + |p|^2)$. As explained in section 1.3, this can be reformulated as (2) with $f(x, \nu, \varphi) = (\nu_3)_+^3$, where $t_+ := \max(t, 0)$ for any $t \in \mathbb{R}$. The set $\mathcal{A}$ is the set of convex bodies with a constraint of the kind (7), with $Q_0 := \Omega \times \{0\}$ and $Q_1 := \Omega \times [0, M]$.

In the classical application, Ω is a disk. So we discretize these constraints by replacing the disk by a regular polygon Ω_ℓ, with ℓ sides. (In practice we used $\ell = 300$.) In this particular problem, this yields an overestimated value of the functional. Indeed, if $A \subset \Omega_\ell \times [0, M]$ is convex, then $\tilde{A} := A \cap Q_1$ belongs to $\mathcal{A}$, and $\mathcal{F}(\tilde{A}) \leq \mathcal{F}(A)$ since $f \geq 0$ and vanishes on $\partial\tilde{A} \setminus \partial A$, where the normal vectors belong to $\{e_3\}^\perp$. Obviously for a minimization problem, this is not a predicament to overestimate the functional.

Using our gradient method on this problem yields different results, starting with different initial shapes. This is likely the consequence of the existence of local minima. (Note that no theoretical result is known on the number or on the kind of critical points in this problem.) So our method needs to be preprocessed to start closer from a global minimum.

We use a genetic algorithm for this task. It is inspired from the ideas developed by Holland [15].

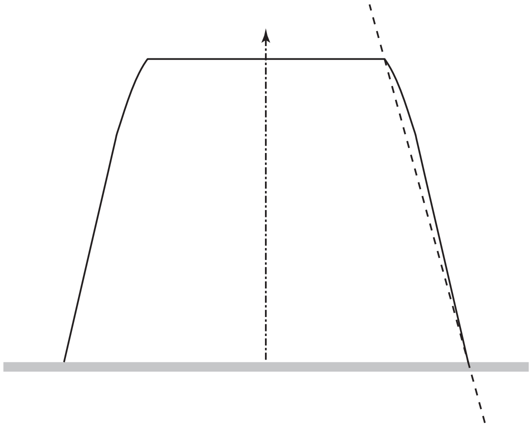


FIG. 6. *Profile of computed optimal shape ($M = 3/2$): the solution is not developable.*

TABLE 2
Minimal values of the Newton's resistance.

M	Newton's radial value	Best developable values	Numerical values
3/2	0.7526	0.7019	0.7012
1	1.1775	1.1561	1.1379
7/10	1.5685	1.5566	1.5457
4/10	2.1074	2.1034	2.1006

Our tests exhibit a behavior corresponding to the theoretical results given in [19]. Even for local minimizers, the image set of ν_A is sparse in $\mathbf{S}^2$. This suggests that optimal sets could be described with a lot fewer parameters as convex hulls of points instead of as an intersection of half-spaces. Therefore, we use the information given in the stochastic step (from the genetic algorithm) in two ways: as an initial set for the gradient method and as an initial guess of the appropriate set of normal vectors to use. But the stochastic step itself represents the convex bodies as convex hull of points in $\Omega_\ell \times [0, M]$, together with the vertices $\Omega_\ell \times \{0\}$. The genetic algorithm optimizes the position of these points.

With these improvements, we get similar shapes for any run of the algorithm. Some of them are pictured in Figure 5 for different values of the parameter M. These solutions are not developable in the sense of [19]. This can be seen more precisely in Figure 6, where only the profile of the body is pictured.

Note that the corresponding values obtained by our method are smaller than the best developable values described in [19], even though they are slightly overestimated, as explained before; see Table 2.

It is a common conjecture on this problem that the solution is smooth except on the top and bottom parts, that is, on $u^{(-1)}(0, M)$. However, C^2-regularity would imply the developability property [19, Conjecture 2]. Our results demonstrate the nonoptimality of the best previously known profiles, and consequently the nonregularity of the minimizers.

REFERENCES

[1] A. D. ALEXANDROV, *Theory of mixed volumes for convex bodies*, Mathem. Sb. USSR, 2 (1937), pp. 947–972.

[2] G. Bellettini, V. Caselles, and M. Novaga, *The total variation flow in* $\mathbb{R}^N$, J. Differential Equations, 184 (2002), pp. 475–525.

[3] F. Brock, V. Ferone, and B. Kawohl, *A symmetry problem in the calculus of variations*, Calc. Var. Partial Differential Equations, 4 (1996), pp. 593–599.

[4] G. Buttazzo and P. Guasoni, *Shape optimization problems over classes of convex domains*, J. Convex Anal., 4 (1997), pp. 343–351.

[5] G. Buttazzo, V. Ferone, and B. Kawohl, *Minimum problems over sets of concave functions and related questions*, Math. Nachr., 173 (1993), pp. 71–89.

[6] G. Carlier, *On a theorem of Alexandrov*, J. Nonlinear Convex. Anal., 5 (2004), pp. 49–58.

[7] G. Carlier and T. Lachand-Robert, *Convex bodies of optimal shape*, J. Convex Anal., 10 (2003), pp. 265–273.

[8] G. Carlier and T. Lachand-Robert, *Regularity of solutions for some variational problems subject to convexity constraint*, Comm. Pure Appl. Math., 54 (2001), pp. 583–594.

[9] G. Carlier, T. Lachand-Robert, and B. Maury, H^1*-projection into sets of convex functions: A saddle point formulation*, in CEMRACS 1999 (Orsay), ESAIM Proc. 10, Soc. Math. Appl. Indust., Paris, 1999, pp. 277–289.

[10] G. Carlier, T. Lachand-Robert, and B. Maury, *A numerical approach to variational problems subject to convexity constraint*, Numer. Math., 88 (2001), pp. 299–318.

[11] J. Cheeger, *A lower bound for the smallest eigenvalue of the Laplacian*, in Problems in Analysis, A Symposium in Honor of Salomon Bochner, R. C. Gunning, ed., Princeton University Press, Princeton, NJ, 1970, pp. 195–199.

[12] P. Choné and J.-C. Rochet, *Ironing, sweeping and multidimensional screening*, Econometrica, 66 (1998), pp. 783–826.

[13] P. Choné and H. Le Meur, *Non-convergence result for conformal approximation of variational problems subject to a convexity constraint*, Numer. Funct. Anal. Optim., 22 (2001), pp. 529–547.

[14] V. Fridman and B. Kawohl, *Isoperimetric estimates for the first eigenvalue of the p-Laplace operator and the Cheeger constant*, Comment. Math. Univ. Carolin., 44 (2003), pp. 659–667.

[15] J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.

[16] I. Ionescu and T. Lachand-Robert, *Generalized Cheeger sets related to landslides*, Calc. Var. Partial Differential Equations, to appear.

[17] B. Kawohl, *On a family of torsional creep problems*, J. Reine Angew. Math., 410 (1990), pp. 1–22.

[18] B. Kawohl and T. Lachand-Robert, *Characterization of Cheeger sets for convex subsets of the plane*, Pacific J. Math., to appear.

[19] T. Lachand-Robert and M. A. Peletier, *Newton's problem of the body of minimal resistance in the class of convex developable functions*, Math. Nachr., 226 (2001), pp. 153–176.

[20] P.-L. Lions, *Identification du cône dual des fonctions convexes et applications*, C. R. Acad. Sci. Paris Sér. I Math., 326 (1998), pp. 1385–1390.

[21] H. Minkowski, *Allgemeine Lehrsätze über die Konvexen Polyeder*, Nach. Ges. Wiss., Göttingen, 1897, pp. 198–219.

[22] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

SIAM J. OPTIM.
Vol. 16, No. 2, pp. 380–399

ON LOVÁSZ–SCHRIJVER LIFT-AND-PROJECT PROCEDURES ON THE DANTZIG–FULKERSON–JOHNSON RELAXATION OF THE TSP*

KEVIN K. H. CHEUNG†

Abstract. We study the Lovász–Schrijver lift-and-project procedure N_+ on the linear relaxation of the Dantzig–Fulkerson–Johnson formulation of the traveling salesman problem (TSP). A long standing conjecture states that the integrality gap of this relaxation is $\frac{4}{3}$ in the case of metric costs. In this paper, we show that the N_+-rank of 2-matching inequalities relative to this relaxation can be arbitrarily high and obtain as a corollary that even after applying N_+ to the relaxation a fixed number of times, the integrality gap of the resulting relaxation is at least $\frac{4}{3}$.

Key words. integrality gap, relaxation, traveling salesman problem, lift-and-project

AMS subject classifications. 90C27, 90C59

DOI. 10.1137/040605849

1. Introduction.

Given a convex set $P \subseteq [0,1]^d$, let P_I denote the convex hull of all the "0-1" points in P. Lovász and Schrijver [16] introduced lift-and-project procedures N and N_+ (to be defined in the next section) for obtaining tighter and tighter convex relaxations of P_I starting from P. They showed that at most d iterative applications of N (or N_+) to P results in P_I. Cook and Dash [5] and Goemans and Tunçel [10] independently gave examples that require exactly d iterations for both N and N_+.

Given a linear inequality $a^T x \leq b$ valid for P_I, the N_+-rank of $a^T x \leq b$ relative to P is the smallest number of iterative applications of N_+ to P that results in a relaxation for which the inequality $a^T x \leq b$ is valid. Lovász and Schrijver [16] showed that the N_+-rank relative to the fractional stable set polytope of many well-known facet-inducing inequalities for the stable set polytope, including the odd-hole, odd-antihole, odd-wheel, clique, and orthonormal-representation inequalities, is equal to 1. However, the N_+-ranks of certain well-known classes of facet-inducing inequalities in the context of other combinatorial problems are not bounded by a constant. For instance, Stephen and Tunçel [20] showed that the N_+-rank of the blossom inequalities relative to the fractional matching polytope can be arbitrarily high.

In this paper, we study the N_+ lift-and-project procedure in the context of the traveling salesman problem (TSP). In particular, we look at the effects of N_+ on the subtour-elimination polytope, which is the feasible region of the linear programming relaxation of the Dantzig–Fulkerson–Johnson [6] formulation of the symmetric TSP. Previous work by Cook and Dash [5] shows that if N_+ is combined with the Gomory–Chvátal cutting-plane procedure, where n denotes the number of cities, at least $\lfloor n/8 \rfloor$ iterations are required to obtain the convex hull of all the integral points.

We show that the N_+-rank of 2-matching inequalities relative to the subtour-elimination polytope can be arbitrarily high. As a corollary, we obtain that, for metric

*Received by the editors March 29, 2004; accepted for publication (in revised form) February 3, 2005; published electronically October 17, 2005. The research of this paper was supported in part by NSERC PDF.

http://www.siam.org/journals/siopt/16-2/60584.html

†School of Mathematics and Statistics, Carleton University, 1125 Colonel By Drive, Ottawa, ON K1S 5B6, Canada (kcheung@math.carleton.ca).

TSP, the integrality gap of the relaxation obtained from applying N_+ iteratively to the subtour-elimination polytope a fixed number of times is at least $\frac{4}{3}$, which is the same as the best lower bound that is known for the integrality gap of the subtour-elimination polytope. Our result is similar in flavor to a result of Arora, Bollobás, and Lovász [1] on the vertex cover problem. They showed that the integrality gap is $2-o(1)$ for three families of linear relaxations, one of which is obtained from applying N iteratively to a classical relaxation of the vertex cover problem a fixed number of times.

The rest of the paper is organized as follows. In section 2, we set up some notation and review the Lovász–Schrijver lift-and-project procedures N and N_+. We also describe the Dantzig–Fulkerson–Johnson relaxation of the TSP and 2-matching inequalities. In section 3, we show that 2-matching inequalities with exactly three teeth can have arbitrarily high N_+-rank relative to the subtour-elimination polytope. In section 4, we discuss the integrality gap of the relaxation obtained from applying N_+ iteratively to the subtour-elimination polytope a fixed number of times. Finally, in section 5, we make some concluding remarks and point out some open questions.

2. Notation and basic properties. In this paper, vectors are written as columns. Let S and T be finite sets. Let $\mathbb{R}^S$ denote the $|S|$-dimensional Euclidean space with entries indexed by the elements of S. Let $\mathbb{R}^{S\times T}$ denote the set of $|S|\times|T|$ real matrices with rows indexed by the elements of S and columns indexed by the elements of T. For a matrix $Y\in\mathbb{R}^{S\times S}$, let $\operatorname{diag}(Y)$ denote the vector $z\in\mathbb{R}^S$ such that $z_i = Y_{i,i}$ for each $i\in S$. For any vector $x\in\mathbb{R}^S$ and any $A\subseteq S$, we use the abbreviation $x(A)$ to mean $\sum_{e\in A} x_e$.

The Lovász–Schrijver lift-and-project procedures N and N_+ can be defined as follows: Let S be a finite set not containing 0 as an element. Let $K\subseteq\mathbb{R}^{\{0\}\cup S}$ be a convex cone such that $K\cap\{x\in\mathbb{R}^{\{0\}\cup S} : x_0 = 1\}\subseteq[0,1]^{\{0\}\cup S}$. For each $j\in\{0\}\cup S$, let $e_j\in\mathbb{R}^{\{0\}\cup S}$ denote the incidence vector of $\{j\}$. Let $\mathbf{e}$ denote the vector of all 1's. Let $\Sigma^{\{0\}\cup S}$ denote the set of symmetric matrices in $\mathbb{R}^{(\{0\}\cup S)\times(\{0\}\cup S)}$. Define

$$\begin{aligned} M(K) := \{Y\in\Sigma^{\{0\}\cup S} : {} & Ye_0 = \operatorname{diag}(Y),\\ & Ye_i\in K\ \forall\ i\in\{0\}\cup S,\\ & Y(e_0 - e_i)\in K\ \forall\ i\in S\}\end{aligned}$$

and

$$N(K) := \{Ye_0 : Y\in M(K)\}.$$

$M(K)$ gives a lifting of the next relaxation of the cone of all the “0-1” vectors in K. Note that $N(K)$ is the projection of $M(K)$ back onto the space of K. One can obtain tighter relaxations by requiring the matrix Y to be positive semidefinite. Let $\Sigma_+^{\{0\}\cup S}$ denote the set of matrices in $\Sigma^{\{0\}\cup S}$ that are positive semidefinite. Define

$$\begin{aligned} M_+(K) := \{Y\in\Sigma_+^{\{0\}\cup S} : {} & Ye_0 = \operatorname{diag}(Y),\\ & Ye_i\in K\ \forall\ i\in\{0\}\cup S,\\ & Y(e_0 - e_i)\in K\ \forall\ i\in S\}\end{aligned}$$

and

$$N_+(K) := \{Ye_0 : Y\in M_+(K)\}.$$

For a convex set $P \subseteq [0,1]^S$, let $\overline{P} \in \mathbb{R}^{\{0\}\cup S}$ denote the cone of $\{\left[\begin{smallmatrix}1\\x\end{smallmatrix}\right] : x \in P\}$. By applying the N operator to P, we mean applying the lift-and-project procedure N to $\overline{P}$ and then projecting the intersection of the resulting cone with $\{y \in \mathbb{R}^{\{0\}\cup S} : y_0 = 1\}$ back onto the space of P. The resulting convex subset of $[0,1]^S$ is denoted by $N(P)$. Define $N_+(P)$ similarly. Define iteratively (similarly for N_+) $N^0(P) := P$ and $N^t(P) := N(N^{t-1}(P))$ for $t \geq 1$.

Lovász and Schrijver [16] showed the following.

THEOREM 1. *If $P \subset [0,1]^d$ is a convex set, then*

$$P \supseteq N(P) \supseteq N^2(P) \supseteq \cdots \supseteq N^d(P) = P_I$$

and

$$P \supseteq N_+(P) \supseteq N_+^2(P) \supseteq \cdots \supseteq N_+^d(P) = P_I.$$

An important property of the N and N_+ operators is that if one can optimize over P in polynomial time, then one can optimize over $N^r(P)$ and $N_+^r(P)$ in polynomial time for any fixed constant r. Consequently, for some $\mathcal{NP}$-hard combinatorial optimization problems, one can use the N and N_+ operators to obtain possibly tighter polynomial-time solvable relaxations. For example, the procedures on the stable set problem have been studied by Lovász and Schrijver [16] and recently in more details by Lipták and Tunçel [15]. Laurent [13] studied the procedures for MAX-CUT. Arora, Bollobás, and Lovász [1] considered the N operator on the vertex cover problem.

Let $a^T x \leq b$ be a valid inequality for P_I. The smallest nonnegative integer r such that $a^T x \leq b$ is valid for $N_+^r(P)$ is called the *N_+-rank of $a^T x \leq b$ relative to P*. It follows from Theorem 1 that the N_+-rank of any valid inequality for P_I is at most d.

We now cite some other properties of the N and N_+ operators.

LEMMA 2 (Cook and Dash [5], Goemans and Tunçel [10]). *Let $r > 0$ be an integer. Let F be any face of $[0,1]^n$ and $P \subseteq [0,1]^n$ be a convex set. Then*

$$N_+^r(P \cap F) = N_+^r(P) \cap F,$$

similarly for N.

Let $S = \{s_1, \ldots, s_n\}$ and $T = \{t_1, \ldots, t_n, t_{n+1}, \ldots, t_{n+k_1}, t_{n+k_1+1}, \ldots, t_{n+k}\}$, where $0 \leq k_1 \leq k$. A function $f : \mathbb{R}^S \to \mathbb{R}^T$ is called an *embedding* operation if

$$y = f(x) \text{ implies that } y_{t_i} = \begin{cases} x_{s_i} & \text{if } 1 \leq i \leq n, \\ 0 & \text{if } n < i \leq n + k_1, \\ 1 & \text{if } n + k_1 < i \leq n + k. \end{cases}$$

The following is well known. (See Cook and Dash [5] for a discussion.)

LEMMA 3. *Let $f : \mathbb{R}^S \to \mathbb{R}^T$ be an embedding operation and $P \in [0,1]^S$ be a convex set. Then $N_+(f(P)) = f(N_+(P))$, similarly for N.*

Iterating Lemma 3 presents the following corollary.

COROLLARY 4. *Let $r > 0$ be an integer. Let $f : \mathbb{R}^S \to \mathbb{R}^T$ be an embedding operation and $P \in [0,1]^S$ be a convex set. Then $N_+^r(f(P)) = f(N_+^r(P))$, similarly for N.*

We now turn our attention to TSP. Let $G = (V, E)$ be a 2-connected simple graph. For a subset S of V, let $\gamma(S)$ denote the set of edges with both ends in S and let $\delta(S)$ denote the set of edges with one end in S and one end not in S. We abbreviate $\delta(\{v\})$ to $\delta(v)$, where $v \in V$. Define the *subtour-elimination polytope of* G, denoted by $\mathrm{SEP}(G)$, to be the set

$$\begin{aligned}\{x \in \mathbb{R}^E : x(\delta(v)) = 2 \ \forall\ v \in V,\\ x(\delta(S)) \geq 2 \ \forall\ S \subset V,\ 3 \leq |S| \leq |V| - 3,\\ 0 \leq x_e \leq 1 \ \forall\ e \in E\}.\end{aligned}$$

We abbreviate $\mathrm{SEP}(K_n)$ to $\mathrm{SEP}(n)$, where K_n denotes the complete graph on n vertices.

Let $\mathrm{TSP}(G)$ denote the convex hull of incidence vectors of Hamiltonian circuits of G. We abbreviate $\mathrm{TSP}(K_n)$ to $\mathrm{TSP}(n)$. It is easy to see that $\mathrm{SEP}(G)_I = \mathrm{TSP}(G)$. If $c \in \mathbb{R}^E$, then the Dantzig–Fulkerson–Johnson [6] relaxation of the TSP on G with respect to the cost vector c is

$$\min\{c^T x : x \in \mathrm{SEP}(G)\}.$$

It is well known that one can optimize a linear function over $\mathrm{SEP}(G)$ in polynomial time using the equivalence of separation and optimization (Grötschel, Lovász, and Schrijver [11]). As $\mathrm{SEP}(G) \subseteq [0, 1]^E$, one can apply the N_+ operator to $\mathrm{SEP}(G)$ to obtain tighter and tighter relaxations of $\mathrm{TSP}(G)$ that are solvable in polynomial time.

We now make two technical observations that will be useful later.

LEMMA 5. *Let $\bar{x} \in \mathbb{R}^E$. Suppose $uv \notin E$. Let $G' = G + uv$. Define $x' \in \mathbb{R}^{E(G')}$ as follows: $x'_e = \bar{x}_e$ if $e \in E$ and $x'_e = 0$ if $e = uv$. If $\bar{x} \in N_+^r(\mathrm{SEP}(G))$ for some integer $r > 0$, then $x' \in N_+^r(\mathrm{SEP}(G'))$. We say x' is obtained from $\bar{x}$ by adding a* 0*-edge.*

Proof. Let $f : \mathbb{R}^E \to \mathbb{R}^{E(G')}$ be an embedding operation such that

$$y = f(x) \text{ implies that } y_e = \begin{cases} x_e & \text{if } e \in E, \\ 0 & \text{if } e = uv. \end{cases}$$

Clearly, $x' = f(\bar{x})$. By Corollary 4,

$$x' \in f(N_+^r(\mathrm{SEP}(G))) = N_+^r(f(\mathrm{SEP}(G))).$$

Let $F = \{x \in \mathbb{R}^{E(G')} : x_{uv} = 0\}$. Observe that $f(\mathrm{SEP}(G)) = \mathrm{SEP}(G') \cap F$. Hence, $x' \in N_+^r(\mathrm{SEP}(G') \cap F)$. It follows from Lemma 2 that $x' \in N_+^r(\mathrm{SEP}(G'))$. □

LEMMA 6. *Let $\bar{x} \in \mathbb{R}^E$. Suppose $\bar{x}_{uv} = 1$ for some $uv \in E$. Let G' denote the graph $(V \cup \{v'\}, E \backslash \{uv\} \cap \{uv', vv'\})$, where $v' \notin V$. Define $x' \in \mathbb{R}^{E(G')}$ as follows: $x'_e = \bar{x}_e$ if $e \in E \backslash \{uv\}$ and $x'_e = 1$ if $e \in \{uv', vv'\}$. If $\bar{x} \in N_+^r(\mathrm{SEP}(G))$ for some integer $r > 0$, then $x' \in N_+^r(\mathrm{SEP}(G'))$. We say x' is obtained from $\bar{x}$ by subdividing a* 1*-edge.*

Proof. Let $f : \mathbb{R}^E \to \mathbb{R}^{E(G')}$ be an embedding operation such that

$$y = f(x) \text{ implies that } y_e = \begin{cases} x_e & \text{if } e \in E \backslash \{uv\}, \\ x_{uv} & \text{if } e = uv' \\ 1 & \text{otherwise.} \end{cases}$$

Clearly, $x' = f(\bar{x})$. Let $F = \{x \in \mathbb{R}^E : x_{uv} = 1\}$. Note that $\bar{x} \in N_+^r(\mathrm{SEP}(G)) \cap F$. Hence, by Lemma 2, $\bar{x} \in N_+^r(\mathrm{SEP}(G) \cap F)$. It follows from Corollary 4 that

$$x' \in f(N_+^r(\mathrm{SEP}(G) \cap F)) = N_+^r(f(\mathrm{SEP}(G) \cap F)).$$

Observe that $f(\mathrm{SEP}(G) \cap F) = \mathrm{SEP}(G') \cap \{x \in \mathbb{R}^{E(G')} : x_{vv'} = 1\} \cap \{x \in \mathbb{R}^{E(G')} : x_{uv'} = 1\}$. Thus, by Lemma 2, $x' \in N_+^r(\mathrm{SEP}(G'))$. □

Many classes of valid inequalities for $\mathrm{TSP}(G)$ have been discovered over the years. Among the earliest known is the class of 2-matching inequalities. Edmonds [7] introduced 2-matching inequalities to obtain a complete linear description of the 2-matching polytope and they are defined as follows: Let $H, T_1, \ldots, T_s \subset V$ be such that $s \geq 3$ is odd, $T_i \cap T_j = \emptyset$ for all distinct $i, j \in \{1, \ldots, s\}$, and $|H \cap T_i| = |T_i \backslash H| = 1$ for $i = 1, \ldots, s$. H is called the *handle* and each T_i is called a *tooth.* The 2-matching inequality with respect to $H, T_1, \ldots, T_s$ is

$$x(\gamma(H)) + \sum_{i=1}^{s} x(\gamma(T_i)) \leq |H| + \frac{s-1}{2}.$$

Grötschel and Padberg [12] showed the following.

THEOREM 7. *2-matching inequalities induce facets of* $\mathrm{TSP}(n)$ *for all* $n \geq 6$.

3. Main result. In this section, we show that 2-matching inequalities with exactly three teeth can have arbitrarily high N_+-rank relative to the subtour-elimination polytope.

Let k be a positive integer. For each $d \in \{0, 1\}$, let $S_{k,d}$ denote the set $\{6 + d, 8 + d, 10 + d, \ldots, 6k + 4 + d\}$. Let H_k denote the graph with vertex-set $V_k = \{v_0, v_1, \ldots, v_{6k+5}\}$ and edge-set $E_k = A \cup B_{k,0} \cup B_{k,1} \cup C_{k,0} \cup C_{k,1}$, where $A = \{v_0v_1, v_2v_3, v_4v_5\}$ and, for each $d \in \{0, 1\}$, $B_{k,d} = \{v_pv_q : p \in \{d, 2{+}d, 4{+}d\}, q \in S_{k,d}\}$, and $C_{k,d} = \{v_pv_q : p \neq q,\ p, q \in S_{k,d}\}$. The graphs H_1 and H_2 are illustrated in Figure 1.

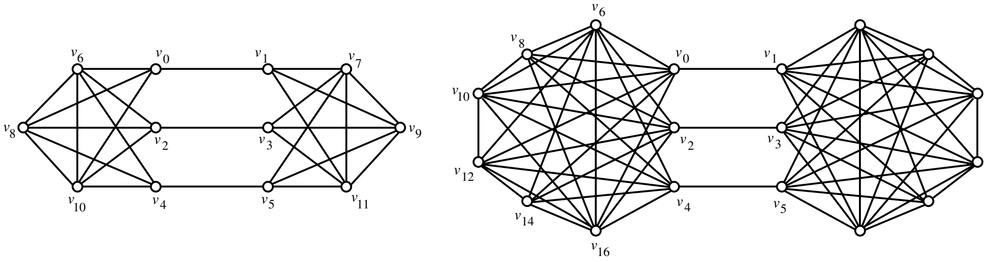


FIG. 1. *H_1 and H_2.*

Define $x^k \in \mathbb{R}^{E_k}$ as follows:

$$x_e^k = \begin{cases} 1 & \text{if } e \in A, \\ \frac{1}{3k} & \text{if } e \in B_{k,0} \cup B_{k,1}, \\ \frac{2k-1}{k(3k-1)} & \text{if } e \in C_{k,0} \cup C_{k,1}. \end{cases}$$

The point x^k is illustrated in Figure 2. The next theorem is the main result of this paper.

THEOREM 8. $x^k \in N_+^{k-1}(\mathrm{SEP}(H_k))$ *for every* $k \geq 1$.

Before we prove this theorem, observe that if G denotes the complete graph with vertex-set V_k and $\bar{x}^k \in \mathbb{R}^{E(G)}$ is such that $\bar{x}^k_e = x^k_e$ for all $e \in E_k$ and $\bar{x}^k_e = 0$ for all $e \in E(G)\backslash E_k$, then by Lemma 5 and Theorem 8, $\bar{x}^k \in N_+^{k-1}(\mathrm{SEP}(G))$. Now, $\bar{x}^k$ does not satisfy the 2-matching inequality with handle $H = S_{k,0}$ and teeth $T_i = \{2i-2, 2i-1\}$ for $i = 1, \ldots, 3$. Hence, the 2-matching inequality has N_+-rank at least k relative to $\mathrm{SEP}(G)$. It is not difficult to extend this result and obtain that this 2-matching inequality has N_+-rank at least k relative to $\mathrm{SEP}(G')$ for every complete graph G' having G as an induced subgraph.

In light of the work of Stephen and Tunçel [20] on the blossom inequalities for the matching polytope, Theorem 8 is perhaps not surprising. (In fact, our proof uses the same approach as theirs but has more technical details to take care of.) However, in the next section we shall use Theorem 8 to show a lower bound of $\frac{4}{3}$ for the integrality gap of a family of relaxations for metric TSP obtained from applying N_+ iteratively to the subtour-elimination polytope a fixed number of times. This result is similar in flavor to the work of Arora, Bollobás, and Lovász [1] on proving integrality gaps of families of linear relaxations for the vertex cover problem without knowing the linear programs explicitly.

Proof of Theorem 8. The proof is by induction on k. The case when $k = 1$ is easy to check. Assume that $x^k \in N_+^{k-1}(\mathrm{SEP}(H_k))$ for some $k \geq 1$. We now prove that $x^{k+1} \in N_+^{k}(\mathrm{SEP}(H_{k+1}))$.

For every $f = v_i v_j \in B_{k+1,d}$ with $i \in \{d, 2+d, 4+d\}$ and $j \in S_{k+1,d}$, where $d \in \{0, 1\}$, define y^f, $\bar{y}^f \in \mathbb{R}^{E_{k+1}}$ as follows:

$$y^f_e = \begin{cases} 1 & \text{if } e \in \{f\} \cup A, \\ \frac{1}{3k+2} & \text{if } e = v_j v_p \text{ for some } p \in S_{k+1,d}\backslash\{j\}, \\ \frac{1}{3k+2} & \text{if } e = v_p v_q \text{ for some } p \in \{d, 2+d, 4+d\}\backslash\{i\}, q \in S_{k+1,d}\backslash\{j\}, \\ \frac{6k+1}{(3k+1)(3k+2)} & \text{if } e = v_p v_q \text{ for some } p, q \in S_{k+1,d}\backslash\{j\},\ p \neq q, \\ \frac{1}{3(k+1)} & \text{if } e \in B_{k+1,1-d}, \\ \frac{2k+1}{(k+1)(3k+2)} & \text{if } e \in C_{k+1,1-d}, \\ 0 & \text{otherwise.} \end{cases}$$

$$\bar{y}^f_e = \begin{cases} 0 & \text{if } e = f, \\ 1 & \text{if } e \in A, \\ \frac{2(3k+1)}{(3k+2)^2} & \text{if } e = v_j v_p \text{ for some } p \in S_{k+1,d}\backslash\{j\}, \\ \frac{1}{3k+2} & \text{if } e = v_i v_p \text{ for some } p \in S_{k+1,d}\backslash\{j\}, \\ \frac{1}{3k+2} & \text{if } e = v_p v_j \text{ for some } p \in \{d, 2+d, 4+d\}\backslash\{i\}, \\ \frac{3k+1}{(3k+2)^2} & \text{if } e = v_p v_q \text{ for some } p \in \{d, 2+d, 4+d\}\backslash\{i\}, q \in S_{k+1,d}\backslash\{j\}, \\ \frac{18k^2+9k+2}{(3k+1)(3k+2)^2} & \text{if } e = v_p v_q \text{ for some } p, q \in S_{k+1,d}\backslash\{j\},\ p \neq q, \\ \frac{1}{3(k+1)} & \text{if } e \in B_{k+1,1-d}, \\ \frac{2k+1}{(k+1)(3k+2)} & \text{if } e \in C_{k+1,1-d}. \end{cases}$$

The points y^f and $\bar{y}^f$ with $f \in B_{k+1,0} \cup B_{k+1,1}$ are illustrated in Figures 3 and 4, respectively.

Claim 1. *y^f, $\bar{y}^f \in N_+^{k-1}(\mathrm{SEP}(H_{k+1}))$ for all $f \in B_{k+1,0} \cup B_{k+1,1}$.*

We postpone the proof of this claim to the end of this section.

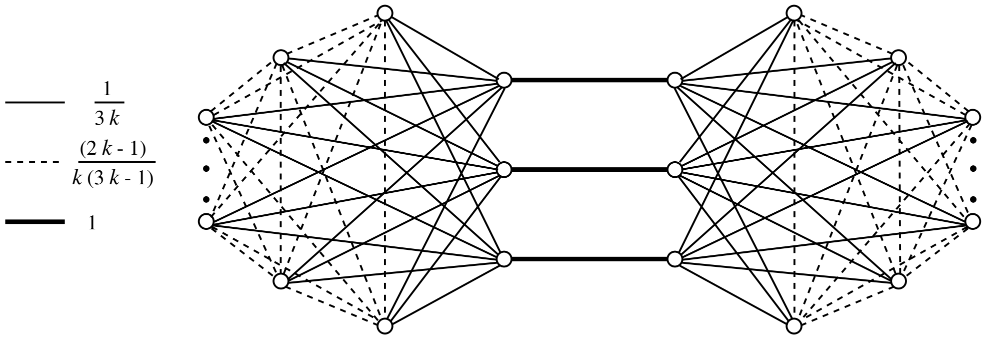


FIG. 2. *The point x^k.*

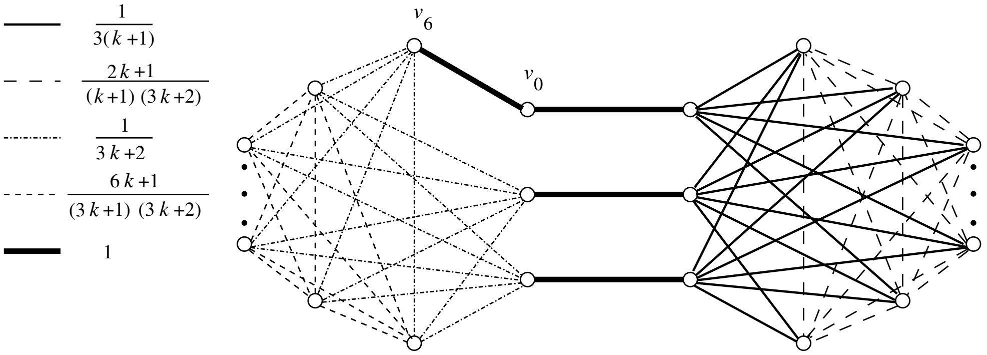


FIG. 3. *The point y^f where $f = v_0v_6$.*

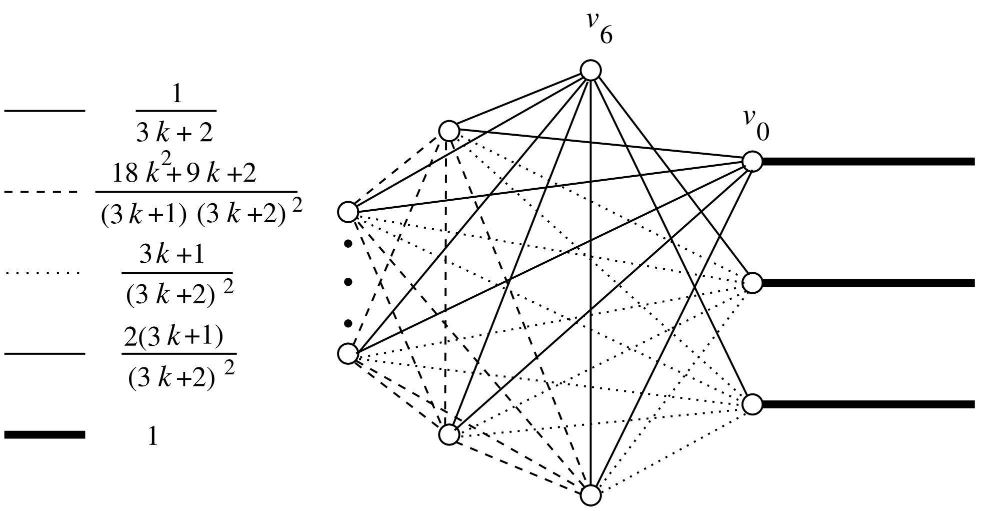

$\frac{1}{3k+2}$
$\frac{18k^2+9k+2}{(3k+1)(3k+2)^2}$
$\frac{3k+1}{(3k+2)^2}$
$\frac{2(3k+1)}{(3k+2)^2}$
1
v_6
v_0

FIG. 4. *Part of the point $\bar{y}^f$ where $f = v_0v_6$.*

One can now check that for every $f \in B_{k+1,0} \cup B_{k+1,1}$,

$$x^{k+1} = x_f^{k+1} y^f + (1 - x_f^{k+1}) \bar{y}^f. \tag{3.1}$$

Since x^{k+1} is a convex combination of y^f and $\bar{y}^f$, it follows from Claim 1 that $x^{k+1} \in N_+^{k-1}(\mathrm{SEP}(H_{k+1}))$.

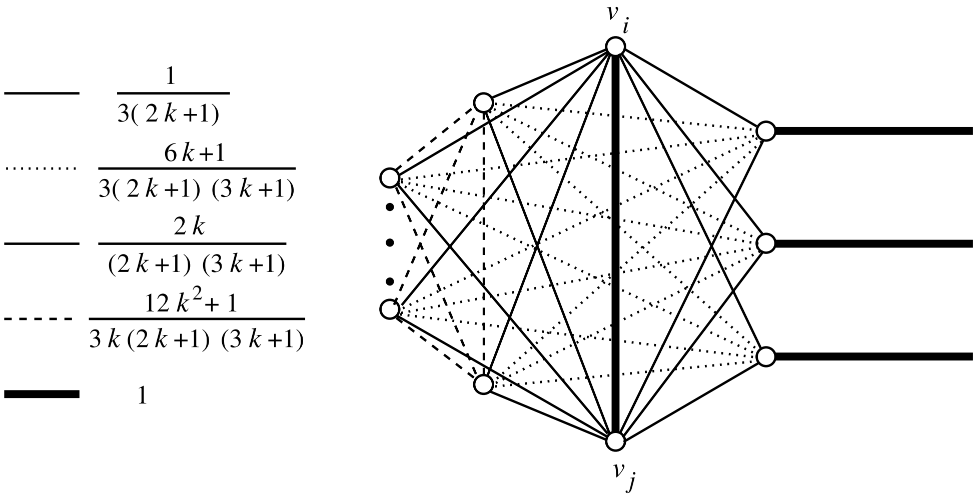

$\frac{1}{3(2k+1)}$
$\frac{6k+1}{3(2k+1)(3k+1)}$
$\frac{2k}{(2k+1)(3k+1)}$
$\frac{12k^2+1}{3k(2k+1)(3k+1)}$
1
v_i
v_j

FIG. 5. *Part of the point y^f where $f \in C_{k+1,0} \cup C_{k+1,1}$.*

Now, for every $f = v_i v_j \in C_{k+1,d}$, where $d \in \{0,1\}$, define y^f, $\bar{y}^f \in \mathbb{R}^{E_{k+1}}$ as follows:

$$y_e^f = \begin{cases} 1 & \text{if } e \in \{f\} \cup A, \\ \frac{1}{3(2k+1)} & \text{if } e = v_p v_q \text{ for some } p \in \{d, 2+d, 4+d\},\ q \in \{i,j\}, \\ \frac{6k+1}{3(2k+1)(3k+1)} & \text{if } e = v_p v_q \text{ for some } p \in \{d, 2+d, 4+d\}, q \in S_{k+1,d} \backslash \{i,j\}, \\ \frac{2k}{(2k+1)(3k+1)} & \text{if } e = v_p v_q \text{ for some } p \in \{i,j\}, q \in S_{k+1,d} \backslash \{i,j\}, \\ \frac{12k^2+1}{3k(2k+1)(3k+1)} & \text{if } e = v_p v_q \text{ for some } p, q \in S_{k+1,d} \backslash \{i,j\},\ p \neq q, \\ \frac{1}{3(k+1)} & \text{if } e \in B_{k+1,1-d}, \\ \frac{2k+1}{(k+1)(3k+2)} & \text{if } e \in C_{k+1,1-d}. \end{cases}$$

$$\bar{y}_e^f = \begin{cases} 0 & \text{if } e = f, \\ 1 & \text{if } e \in A, \\ \frac{3k+1}{3(3k^2+3k+1)} & \text{if } e = v_p v_q \text{ for some } p \in \{d, 2+d, 4+d\},\ q \in \{i,j\}, \\ \frac{9k^2+3k+1}{3(3k+1)(3k^2+3k+1)} & \text{if } e = v_p v_q \text{ for some } p \in \{d, 2+d, 4+d\}, q \in S_{k+1,d} \backslash \{i,j\}, \\ \frac{6k^2+3k+1}{(3k+1)(3k^2+3k+1)} & \text{if } e = v_p v_q \text{ for some } p \in \{i,j\}, q \in S_{k+1,d} \backslash \{i,j\}, \\ \frac{18k^3+3k^2+3k-1}{3k(3k+1)(3k^2+3k+1)} & \text{if } e = v_p v_q \text{ for some } p, q \in S_{k+1,d} \backslash \{i,j\},\ p \neq q, \\ \frac{1}{3(k+1)} & \text{if } e \in B_{k+1,1-d}, \\ \frac{2k+1}{(k+1)(3k+2)} & \text{if } e \in C_{k+1,1-d}. \end{cases}$$

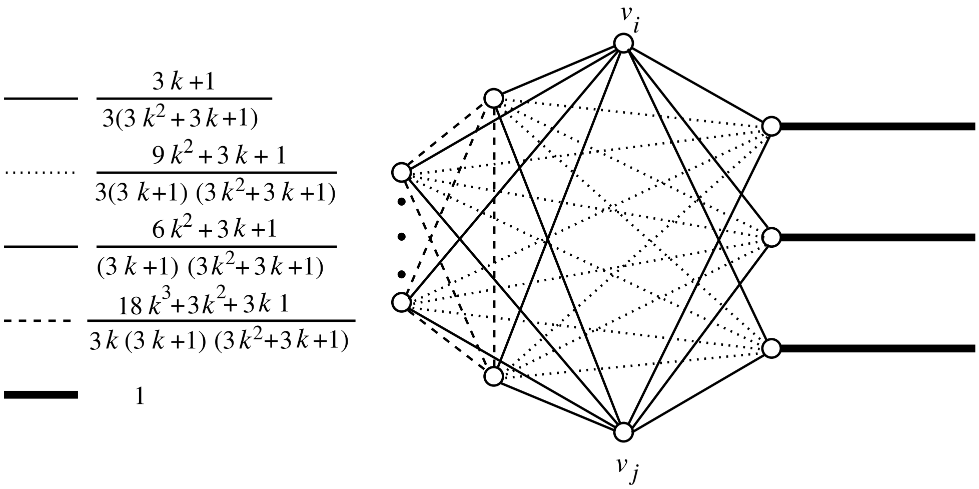

$\frac{3k+1}{3(3k^2+3k+1)}$
$\frac{9k^2+3k+1}{3(3k+1)(3k^2+3k+1)}$
$\frac{6k^2+3k+1}{(3k+1)(3k^2+3k+1)}$
$\frac{18k^3+3k^2+3k\ 1}{3k(3k+1)(3k^2+3k+1)}$
1
v_i
v_j

FIG. 6. *Part of the point* $\bar{y}^f$ *where* $f = v_i v_j \in C_{k+1,0} \cup C_{k+1,1}$.

The points y^f and $\bar{y}^f$ with $f \in C_{k+1,0} \cup C_{k+1,1}$ are illustrated in Figures 5 and 6, respectively.

CLAIM 2. $y^f, \bar{y}^f \in N_+^{k-1}(\mathrm{SEP}(H_{k+1}))$ *for all* $f \in C_{k+1,0} \cup C_{k+1,1}$.

We postpone the proof of this claim to the end of this section.

One can now check that for every $f \in C_{k+1,0} \cup C_{k+1,1}$

$$x^{k+1} = x_f^{k+1} y^f + (1 - x_f^{k+1})\bar{y}^f. \tag{3.2}$$

Define the matrix $Y \in \mathbb{R}^{(\{0\}\cup E_{k+1})\times(\{0\}\cup E_{k+1})}$ as follows:

For all $f \in \{0\} \cup A$, $Ye_f := \left[{1 \atop x^{k+1}}\right]$.

For all $f \in E_{k+1}\backslash A$, $Ye_f := x_f^{k+1}\left[{1 \atop y^f}\right]$.

Let K denote the cone $\overline{\mathrm{SEP}(H_{k+1})}$. We now show that $Y \in M_+(N_+^{k-1}(K))$.

First note that $\mathrm{diag}(Y) = Ye_0 = Y^T e_0$ as $y_f^f = 1$ for all $f \in E_{k+1}\backslash A$. In addition, it follows from Claims 1 and 2 and (3.1) and (3.2) that $Ye_f \in N_+^{k-1}(K)$ and $Y(e_0 - e_f) \in N_+^{k-1}(K)$ for all $f \in E_{k+1}$.

Next, we show that Y is symmetric. Clearly, if $f \in A$, then $Y_{f',f} = Y_{f,f'}$ for all $f' \in E_{k+1}$.

Let $f = v_i v_j \in B_{k+1,d}$ with $i \in \{d, 2+d, 4+d\}$ and $j \in S_{k+1,d}$ for some $d \in \{0, 1\}$.

Let $f' \in E_{k+1} \backslash A$. Then

$$Y_{f',f} = x_f^{k+1} y_{f'}^f = \frac{1}{3(k+1)} y_{f'}^f$$
$$= \begin{cases} \frac{1}{3(k+1)(3k+2)} & \text{if } f' = v_j v_p \text{ for some } p \in S_{k+1,d} \backslash \{j\}, \\ \frac{1}{3(k+1)(3k+2)} & \text{if } f' = v_p v_q \text{ for some } p \in \{d, 2+d, 4+d\} \backslash \{i\}, \\ & \qquad\qquad q \in S_{k+1,d} \backslash \{j\}, \\ \frac{6k+1}{3(k+1)(3k+1)(3k+2)} & \text{if } f' = v_p v_q \text{ for some } p, q \in S_{k+1,d} \backslash \{j\},\ p \neq q, \\ \frac{1}{9(k+1)^2} & \text{if } f' \in B_{k+1,1-d}, \\ \frac{2k+1}{3(k+1)^2(3k+2)} & \text{if } f' \in C_{k+1,1-d}, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$Y_{f,f'} = x_{f'}^{k+1} y_f^{f'}$$
$$= \begin{cases} \frac{2k+1}{(k+1)(3k+2)} \frac{1}{3(2k+1)} & \text{if } f' = v_j v_p \text{ for some } p \in S_{k+1,d} \backslash \{j\}, \\ \frac{1}{3(k+1)} \frac{1}{3k+2} & \text{if } f' = v_p v_q \text{ for some } p \in \{d, 2+d, 4+d\} \backslash \{i\}, \\ & \qquad\qquad q \in S_{k+1,d} \backslash \{j\}, \\ \frac{2k+1}{(k+1)(3k+2)} \frac{6k+1}{3(2k+1)(3k+1)} & \text{if } f' = v_p v_q \text{ for some } p, q \in S_{k+1,d} \backslash \{j\},\ p \neq q, \\ \frac{1}{9(k+1)^2} & \text{if } f' \in B_{k+1,1-d}, \\ \frac{2k+1}{(k+1)(3k+2)} \frac{1}{3(k+1)} & \text{if } f' \in C_{k+1,1-d}, \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, $Y_{f',f} = Y_{f,f'}$.

Next, let $f = v_i v_j \in C_{k+1,d}$ for some $d \in \{0,1\}$ and $f' \in E_{k+1} \backslash (A \cup B_{k+1,0} \cup B_{k+1,1})$. Then

$$Y_{f',f} = x_f^{k+1} y_{f'}^f = \frac{2k+1}{(k+1)(3k+2)} y_{f'}^f$$
$$= \begin{cases} \frac{2k}{(k+1)(3k+1)(3k+2)} & \text{if } f' = v_p v_q \text{ for some } p \in \{i,j\},\ q \in S_{k+1,d} \backslash \{i,j\}, \\ \frac{12k^2+1}{3k(k+1)(3k+1)(3k+2)} & \text{if } f' = v_p v_q \text{ for some } p, q \in S_{k+1,d} \backslash \{i,j\}, p \neq q, \\ (\frac{(2k+1)}{(k+1)(3k+2)})^2 & \text{if } f' \in C_{k+1,1-d}, \end{cases}$$

and

$$Y_{f,f'} = x_{f'}^{k+1} y_f^{f'} = \frac{2k+1}{(k+1)(3k+2)} y_f^{f'}$$
$$= \begin{cases} \frac{2k+1}{(k+1)(3k+2)} \frac{2k}{(2k+1)(3k+1)} & \text{if } f' = v_p v_q \text{ for some } p \in \{i,j\},\ q \in S_{k+1,d} \backslash \{i,j\}, \\ \frac{2k+1}{(k+1)(3k+2)} \frac{12k^2+1}{3k(2k+1)(3k+1)} & \text{if } f' = v_p v_q \text{ for some } p, q \in S_{k+1,d} \backslash \{i,j\}, p \neq q, \\ (\frac{2k+1}{(k+1)(3k+2)})^2 & \text{if } f' \in C_{k+1,1-d}. \end{cases}$$

Again, we have $Y_{f,f'} = Y_{f',f}$. Hence, Y is symmetric.

Finally, we show that Y is positive semidefinite by identifying its eigenvalues and the corresponding eigenspaces.

The eigenvalues of Y are 0, $\frac{k(3k+4)}{(k+1)(3k+1)(3k+2)}$, $\frac{1}{3k+2}$, $\frac{18k^3+15k^2+3k+1}{3k(k+1)(3k+1)(3k+2)}$, and $\frac{24k^2+38k+15}{(k+1)(3k+2)}$ with the dimensions of the corresponding eigenspaces being $3(2k+5)$, $2(3k+2)$, $4(3k+2)$, $9k(k+1)$, and 1, respectively.

We first identify vectors in the nullspace of Y; that is, the eigenspace corresponding to the eigenvalue 0. For each $e \in A$, let $u^e \in \mathbb{R}^{\{0\} \cup E_{k+1}}$ be such that $u^e_0 = 1$, $u^e_e = -1$, and 0 everywhere else. Clearly, $Yu^e = 0$ for all $e \in A$. For each $v \in V_{k+1} \backslash \{v_0\}$, define $w^v \in \mathbb{R}^{\{0\} \cup E_{k+1}}$ as follows: Set $w^v_e := 1$ for every edge $e \in \delta(\{v_0, v\})$ that is incident to v_0. Set $w^v_e := -1$ for every edge $e \in \delta(\{v_0, v\})$ that is incident to v. Set the 0th entry and all the other entries to 0. One can check that $Yw^v = 0$ for every $v \in V_{k+1} \backslash \{v_0\}$. Finally,

$$\begin{bmatrix} -6(k+2) \\ \mathbf{e} \end{bmatrix}$$

is also a vector in the nullspace of Y. It is not difficult to see that the above $3 + (6k + 11) + 1 = 6k + 15$ vectors are linearly independent.

For every $d \in \{0, 1\}$ and $i \in \{4, \ldots, 3k+5\}$, define the vector $u^{d,i} \in \mathbb{R}^{\{0\} \cup E_{k+1}}$ as follows:

$$u^{d,i}_e = \begin{cases} 1 & \text{if } e = v_{6+d}v_q \quad \text{for some } q \in \{d, 2+d, 4+d\}, \\ -\frac{3}{3k+1} & \text{if } e = v_{6+d}v_q \quad \text{for some } q \in S_{k+1,d} \backslash \{6+d, 2i+d\}, \\ -1 & \text{if } e = v_{2i+d}v_q \quad \text{for some } q \in \{d, 2+d, 4+d\}, \\ \frac{3}{3k+1} & \text{if } e = v_{2i+d}v_q \quad \text{for some } q \in S_{k+1,d} \backslash \{6+d, 2i+d\}, \\ 0 & \text{otherwise.} \end{cases}$$

There are $2(3k+2)$ such vectors and they are linearly independent. One can check that they are eigenvectors with eigenvalue $\frac{k(3k+4)}{(k+1)(3k+1)(3k+2)}$.

For every $d \in \{0, 1\}$, $a \in \{2+d, 4+d\}$, and $i \in \{4, \ldots, 3k+5\}$, define the vector $w^{d,a,i} \in \mathbb{R}^{\{0\} \cup E_{k+1}}$ as follows:

$$w^{d,a,i}_e = \begin{cases} 1 & \text{if } e \in \{v_d v_{6+d}, v_a v_{2i+d}\}, \\ -1 & \text{if } e \in \{v_d v_{2i+d}, v_a v_{6+d}\}, \\ 0 & \text{otherwise.} \end{cases}$$

There are $4(3k+2)$ such vectors and they are linearly independent. One can check that they are eigenvectors with eigenvalue $\frac{1}{3k+2}$.

If $|C_{k+1,d}| = 3(k+1)$ is odd, then let $\mathcal{C}_d = \{v_{6+d}v_{8+d}, v_{8+d}v_{10+d}, \ldots, v_{6k+10+d}v_{6+d}\}$. If $|C_{k+1,d}|$ is even, then let $\mathcal{C}_d = \{v_{6k+8+d}v_{6k+10+d}\} \cup \{v_{6+d}v_{8+d},\ v_{8+d}v_{10+d},\ \ldots,\ v_{6k+8+d}v_{6+d}\}$. (See Figure 7.) Observe that in either case, for each $e \in C_{k+1,d} \backslash \mathcal{C}_d$, there is one odd and one even circuit containing e in $\mathcal{C}_d \cup \{e\}$. Let C^e denote the even circuit containing e.

Now, for all $e \in C_{k+1,0} \cup C_{k+1,1} \backslash (\mathcal{C}_0 \cup \mathcal{C}_1)$, consider the signed incidence vectors of the even circuits C^e, where the signing is performed so that consecutive edges have opposite signs. (The 0th component is set to 0.) Since there are $\binom{3(k+1)}{2} - 3(k+1)$ edges in $C_{k+1,d} \backslash \mathcal{C}_d$ for each $d \in \{0, 1\}$, there are in total $9k(k+1)$ such signed incidence vectors and they are linearly independent (since each e appears only in one circuit.) One can check that they are eigenvectors with eigenvalue $\frac{18k^3+15k^2+3k+1}{3k(k+1)(3k+1)(3k+2)}$. Finally, $\begin{bmatrix} 1 \\ x^{k+1} \end{bmatrix}$ is an eigenvector corresponding to the eigenvalue $\frac{24k^2+38k+15}{(k+1)(3k+2)}$.

Since the sum of the lower bounds for the dimensions of the eigenspaces obtained above is equal to the order of the matrix Y, we have identified all the eigenvalues of Y. As all the eigenvalues are nonnegative, Y is positive semidefinite. Thus, $Y \in M_+(N_+^{k-1}(K))$. It follows that $Ye_0 \in N_+^k(K)$, implying that $x^{k+1} \in N_+^k(\mathrm{SEP}(H_{k+1}))$. This completes the induction.

Proof of Claim 1. Without loss of generality, we may assume that $f = v_0v_j$, where $j \in S_{k+1,0}$.

Observe that

$$\bar{y}^f = \sum_{p \in S_{k+1,0}\backslash\{j\}} \frac{1}{3k+2} y^{v_0v_p}.$$

Thus, it suffices to show that $y^f \in N_+^{k-1}(\mathrm{SEP}(H_{k+1}))$.

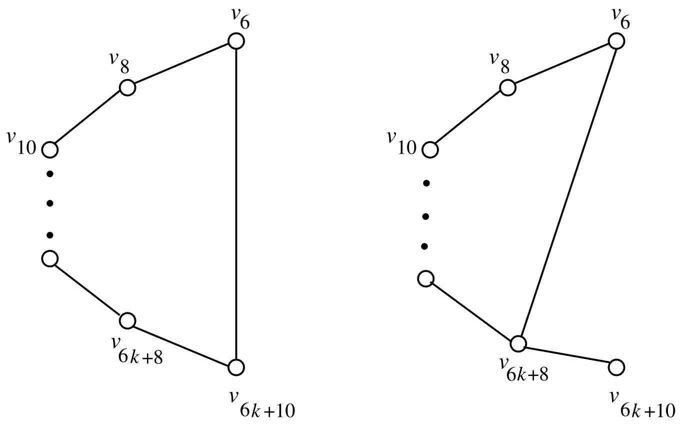
Given distinct $a, b \in S_{k+1,0}\backslash\{j\}$ and distinct $r, s, t \in S_{k+1,1}$, define $w(a,b,r,s,t) \in \mathbb{R}^{E_{k+1}}$ as follows:

$$w(a,b,r,s,t)_e = \begin{cases} 1 & \text{if } e \in \{v_0v_j, v_2v_a, v_4v_b, v_1v_r, v_3v_s, v_5v_t\} \cup A, \\ \frac{1}{3k} & \text{if } e = v_pv_q \text{ for some } p \in \{a,b,j\}, q \in S_{k+1,0}\backslash\{a,b,j\}, \\ \frac{1}{3k} & \text{if } e = v_pv_q \text{ for some } p \in \{r,s,t\}, q \in S_{k+1,1}\backslash\{r,s,t\}, \\ \frac{2k-1}{k(3k-1)} & \text{if } e = v_pv_q \text{ for some } p, q \in S_{k+1,0}\backslash\{a,b,j\}, p \neq q, \\ \frac{2k-1}{k(3k-1)} & \text{if } e = v_pv_q \text{ for some } p, q \in S_{k+1,1}\backslash\{r,s,t\}, p \neq q, \\ 0 & \text{otherwise.} \end{cases}$$

The point $w(a,b,r,s,t)$ is illustrated in Figure 8. Clearly, $w(a,b,r,s,t)$ can be obtained from x^k by adding 0-edges and subdividing 1-edges. By Lemmas 5 and 6, $w(a,b,r,s,t) \in N_+^{k-1}(\mathrm{SEP}(H_{k+1}))$.

It is not difficult to check that y^f is the average of all the points $w(a,b,r,s,t)$ as a,b,r,s,t run over all possibilities. It follows that $y^f \in N_+^{k-1}(\mathrm{SEP}(H_{k+1}))$.

Proof of Claim 2. Without loss of generality, we may assume that $f = v_iv_j \in C_{k+1,0}$. Given $l \in \{i,j\}$, $a \in \{0,2,4\}$, $b \in S_{k+1,0}\backslash\{i,j\}$, and distinct $r,s,t \in S_{k+1,1}$,



3(k +1) is odd 3(k +1) is even

FIG. 7. $\mathcal{C}_0$.

let $l' \in \{i, j\}$ be such that $l' \neq l$ and define $w(l, a, b, r, s, t) \in \mathbb{R}^{E_{k+1}}$ as follows:

$$w(l,a,b,r,s,t)_e = \begin{cases} 1 & \text{if } e \in \{v_a v_l, v_i v_j, v_{l'} v_b, v_1 v_r, v_3 v_s, v_5 v_t\} \cup A, \\ \frac{1}{3k} & \text{if } e = v_p v_q \text{ for some } p \in \{b, 0, 2, 4\} \backslash \{a\}, \\ & \qquad\qquad q \in S_{k+1,0} \backslash \{b, i, j\}, \\ \frac{1}{3k} & \text{if } e = v_p v_q \text{ for some } p \in \{r, s, t\}, q \in S_{k+1,1} \backslash \{r, s, t\}, \\ \frac{2k-1}{k(3k-1)} & \text{if } e = v_p v_q \text{ for some } p, q \in S_{k+1,0} \backslash \{b, i, j\}, p \neq q, \\ \frac{2k-1}{k(3k-1)} & \text{if } e = v_p v_q \text{ for some } p, q \in S_{k+1,1} \backslash \{r, s, t\}, p \neq q, \\ 0 & \text{otherwise.} \end{cases}$$

The point $w(i, 2, b, r, s, t)$ is illustrated in Figure 9.

Let w denote the average of all the points $w(l, a, b, r, s, t)$ as l, a, b, r, s, t run over

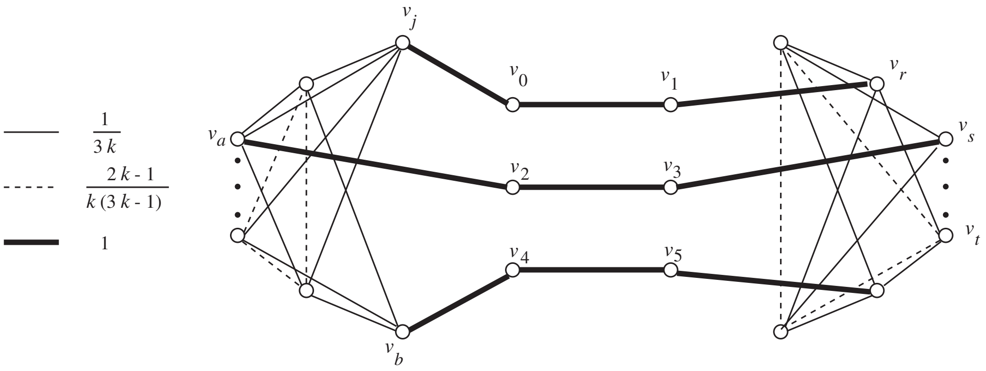


FIG. 8. $w(a, b, r, s, t)$.

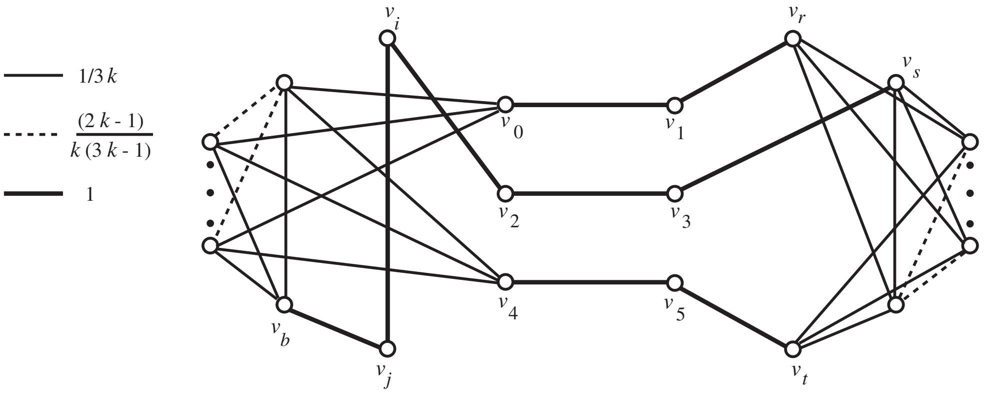


FIG. 9. $w(i, 2, b, r, s, t)$.

all possibilities. It is not difficult to check that

$$w_e = \begin{cases} 1 & \text{if } e \in \{f\} \cup A, \\ \frac{1}{6} & \text{if } e = v_p v_q \text{ for some } p \in \{0,2,4\},\ q \in \{i,j\}, \\ \frac{2}{3(3k+1)} & \text{if } e = v_p v_q \text{ for some } p \in \{0,2,4\}, q \in S_{k+1,0} \backslash \{i,j\}, \\ \frac{1}{2(3k+1)} & \text{if } e = v_p v_q \text{ for some } p \in \{i,j\}, q \in S_{k+1,0} \backslash \{i,j\}, \\ \frac{6k-1}{3k(3k+1)} & \text{if } e = v_p v_q \text{ for some } p,q \in S_{k+1,0} \backslash \{i,j\},\ p \neq q, \\ \frac{1}{3(k+1)} & \text{if } e \in B_{k+1,1}, \\ \frac{2k+1}{(k+1)(3k+1)} & \text{if } e \in C_{k+1,1}. \end{cases}$$

Clearly, $w(l,a,b,r,s,t)$ can be obtained from x^k by adding 0-edges and subdividing 1-edges. By Lemmas 5 and 6, $w(l,a,b,r,s,t) \in N_+^{k-1}(\mathrm{SEP}(H_{k+1}))$. It follows that $w \in N_+^{k-1}(\mathrm{SEP}(H_{k+1}))$.

Next, given $l \in \{i,j\}$, $a \in \{0,2,4\}$, $b \in S_{k+1,0} \backslash \{i,j\}$, and distinct $r,s,t \in S_{k+1,1}$, let $l' \in \{i,j\}$ be such that $l' \neq l$ and define $u(l,a,b,r,s,t) \in \mathbb{R}^{E_{k+1}}$ as follows:

$$u(l,a,b,r,s,t)_e = \begin{cases} 1 & \text{if } e \in \{v_a v_b, v_i v_j, v_b v_l, v_1 v_r, v_3 v_s, v_5 v_t\} \cup A, \\ \frac{1}{3k} & \text{if } e = v_p v_q \text{ for some } p \in \{l',0,2,4\} \backslash \{a\}, \\ & \qquad\qquad\qquad q \in S_{k+1,0} \backslash \{b,i,j\}, \\ \frac{1}{3k} & \text{if } e = v_p v_q \text{ for some } p \in \{r,s,t\}, q \in S_{k+1,1} \backslash \{r,s,t\}, \\ \frac{2k-1}{k(3k-1)} & \text{if } e = v_p v_q \text{ for some } p,q \in S_{k+1,0} \backslash \{b,i,j\}, p \neq q, \\ \frac{2k-1}{k(3k-1)} & \text{if } e = v_p v_q \text{ for some } p,q \in S_{k+1,1} \backslash \{r,s,t\}, p \neq q, \\ 0 & \text{otherwise.} \end{cases}$$

The point $u(j,2,b,r,s,t)$ is illustrated in Figure 10.

Let u denote the average of all the points $u(l,a,b,r,s,t)$ as l,a,b,r,s,t run over all possibilities. It is not difficult to check that

$$u_e = \begin{cases} 1 & \text{if } e \in \{f\} \cup A, \\ \frac{1}{(3k+1)} & \text{if } e = v_p v_q \text{ for some } p \in \{0,2,4\}, q \in S_{k+1,0} \backslash \{i,j\}, \\ \frac{1}{(3k+1)} & \text{if } e = v_p v_q \text{ for some } p \in \{i,j\}, q \in S_{k+1,0} \backslash \{i,j\}, \\ \frac{2k-1}{k(3k+1)} & \text{if } e = v_p v_q \text{ for some } p,q \in S_{k+1,0} \backslash \{i,j\},\ p \neq q, \\ \frac{1}{3(k+1)} & \text{if } e \in B_{k+1,1}, \\ \frac{2k+1}{(k+1)(3k+1)} & \text{if } e \in C_{k+1,1}, \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, $u(l,a,b,r,s,t)$ can be obtained from x^k by adding 0-edges and subdividing 1-edges. By Lemmas 5 and 6, $u(l,a,b,r,s,t) \in N_+^{k-1}(\mathrm{SEP}(H_{k+1}))$. It follows that $u \in N_+^{k-1}(\mathrm{SEP}(H_{k+1}))$.

One can now check that

$$y^f = \frac{2}{2k+1} w + \left(1 - \frac{2}{2k+1}\right) u.$$

Since $w, u \in N_+^{k-1}(\mathrm{SEP}(H_{k+1}))$ and y^f is a convex combination of w and u, we have $y^f \in N_+^{k-1}(\mathrm{SEP}(H_{k+1}))$.

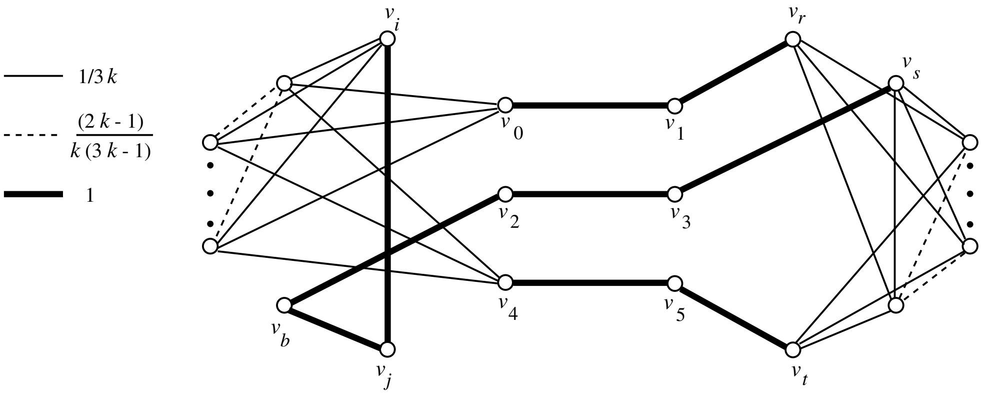


FIG. 10. $u(j, 2, b, r, s, t)$.

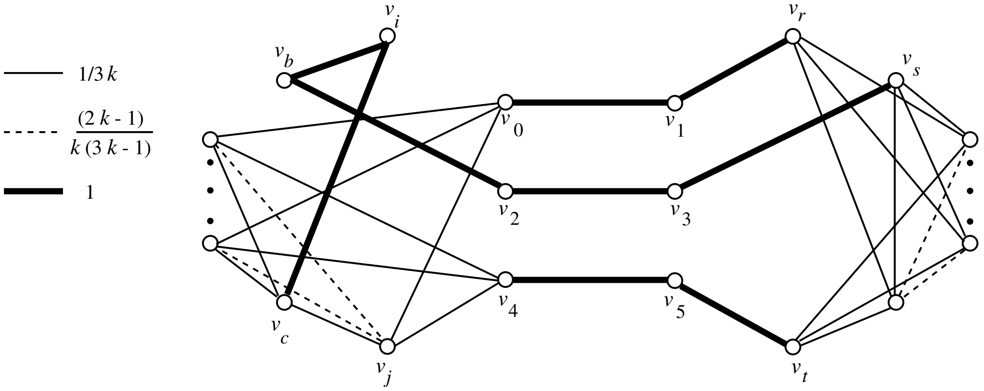


FIG. 11. $\bar{w}(i, 2, b, c, r, s, t)$.

Given $l \in \{i, j\}$, $a \in \{0, 2, 4\}$, distinct $b, c \in S_{k+1,0} \backslash \{i, j\}$, and distinct $r, s, t \in S_{k+1,1}$, define $\bar{w}(l, a, b, c, r, s, t) \in \mathbb{R}^{E_{k+1}}$ as follows:

$$\bar{w}(l, a, b, c, r, s, t)_e = \begin{cases} 1 & \text{if } e \in \{v_a v_b, v_b v_l, v_l v_c, v_1 v_r, v_3 v_s, v_5 v_t\} \cup A, \\ \frac{1}{3k} & \text{if } e = v_p v_q \text{ for some } p \in \{c, 0, 2, 4\} \backslash \{a\}, \\ & \qquad\qquad\qquad q \in S_{k+1,0} \backslash \{b, c, l\}, \\ \frac{1}{3k} & \text{if } e = v_p v_q \text{ for some } p \in \{r, s, t\}, q \in S_{k+1,1} \backslash \{r, s, t\}, \\ \frac{2k-1}{k(3k-1)} & \text{if } e = v_p v_q \text{ for some } p, q \in S_{k+1,0} \backslash \{b, c, l\}, p \neq q, \\ \frac{2k-1}{k(3k-1)} & \text{if } e = v_p v_q \text{ for some } p, q \in S_{k+1,1} \backslash \{r, s, t\}, p \neq q, \\ 0 & \text{otherwise.} \end{cases}$$

The point $\bar{w}(i, 2, b, c, r, s, t)$ is illustrated in Figure 11.

Let $\bar{w}$ denote the average of all the points $\bar{w}(l, a, b, c, r, s, t)$ as l, a, b, c, r, s, t run over all possibilities. It is not difficult to check that

$$\bar{w}_e = \begin{cases} 1 & \text{if } e \in A, \\ \frac{1}{9k} & \text{if } e = v_p v_q \text{ for some } p \in \{0,2,4\},\ q \in \{i,j\}, \\ \frac{9k-2}{9k(3k+1)} & \text{if } e = v_p v_q \text{ for some } p \in \{0,2,4\}, q \in S_{k+1,0} \backslash \{i,j\}, \\ \frac{6k-1}{3k(3k+1)} & \text{if } e = v_p v_q \text{ for some } p \in \{i,j\}, q \in S_{k+1,0} \backslash \{i,j\}, \\ \frac{18k^2-15k+4}{9k^2(3k+1)} & \text{if } e = v_p v_q \text{ for some } p,q \in S_{k+1,0} \backslash \{i,j\},\ p \neq q, \\ \frac{1}{3(k+1)} & \text{if } e \in B_{k+1,1}, \\ \frac{2k+1}{(k+1)(3k+1)} & \text{if } e \in C_{k+1,1}, \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, $\bar{w}(l,a,b,c,r,s,t)$ can be obtained from x^k by adding 0-edges and subdividing 1-edges. By Lemmas 5 and 6, $\bar{w}(l,a,b,c,r,s,t) \in N_+^{k-1}(\text{SEP}(H_{k+1}))$. Hence, $\bar{w} \in N_+^{k-1}(\text{SEP}(H_{k+1}))$.

Next, given distinct $a,b,c \in \{0,2,4\}$, $l \in S_{k+1,0} \backslash \{i,j\}$, and distinct $r,s,t \in S_{k+1,1}$, define $\bar{u}(a,b,c,l,r,s,t) \in \mathbb{R}^{E_{k+1}}$ as follows:

$$\bar{u}(a,b,c,l,r,s,t)_e = \begin{cases} 1 & \text{if } e \in \{v_a v_i, v_b v_j, v_c v_l, v_1 v_r, v_3 v_s, v_5 v_t\} \cup A, \\ \frac{1}{3k} & \text{if } e = v_p v_q \text{ for some } p \in \{i,j,l\}, q \in S_{k+1,0} \backslash \{i,j,l\}, \\ \frac{1}{3k} & \text{if } e = v_p v_q \text{ for some } p \in \{r,s,t\}, q \in S_{k+1,1} \backslash \{r,s,t\}, \\ \frac{2k-1}{k(3k-1)} & \text{if } e = v_p v_q \text{ for some } p,q \in S_{k+1,0} \backslash \{i,j,l\}, p \neq q, \\ \frac{2k-1}{k(3k-1)} & \text{if } e = v_p v_q \text{ for some } p,q \in S_{k+1,1} \backslash \{r,s,t\}, p \neq q, \\ 0 & \text{otherwise.} \end{cases}$$

Let $\bar{u}$ denote the average of all the points $\bar{u}(a,b,c,l,r,s,t)$ as a,b,c,l,r,s,t run over all possibilities. It is not difficult to check that

$$\bar{u}_e = \begin{cases} 1 & \text{if } e \in A, \\ \frac{1}{3} & \text{if } e = v_p v_q \text{ for some } p \in \{0,2,4\},\ q \in \{i,j\}, \\ \frac{1}{3(3k+1)} & \text{if } e = v_p v_q \text{ for some } p \in \{0,2,4\}, q \in S_{k+1,0} \backslash \{i,j\}, \\ \frac{1}{3k+1} & \text{if } e = v_p v_q \text{ for some } p \in \{i,j\}, q \in S_{k+1,0} \backslash \{i,j\}, \\ \frac{6k-1}{3k(3k+1)} & \text{if } e = v_p v_q \text{ for some } p,q \in S_{k+1,0} \backslash \{i,j\},\ p \neq q, \\ \frac{1}{3(k+1)} & \text{if } e \in B_{k+1,1}, \\ \frac{2k+1}{(k+1)(3k+1)} & \text{if } e \in C_{k+1,1}. \end{cases}$$

Clearly, $\bar{u}(a,b,c,l,r,s,t)$ can be obtained from x^k by adding 0-edges and subdividing 1-edges. By Lemmas 5 and 6, $\bar{u}(a,b,c,l,r,s,t) \in N_+^{k-1}(\text{SEP}(H_{k+1}))$. Hence, $\bar{u} \in N_+^{k-1}(\text{SEP}(H_{k+1}))$.

One can now check that

$$\bar{y}^f = \frac{9k^3}{(3k-1)(3k^2+3k+1)} \bar{w} + \left(1 - \frac{9k^3}{(3k-1)(3k^2+3k+1)}\right) \bar{u}.$$

Since $\bar{w}, \bar{u} \in N_+^{k-1}(\text{SEP}(H_{k+1}))$ and $\bar{y}^f$ is a convex combination of $\bar{w}$ and $\bar{u}$, we have $\bar{y}^f \in N_+^{k-1}(\text{SEP}(H_{k+1}))$.

4. Integrality gap of $N_+^r(\mathbf{SEP}(n))$. Let $c \in \mathbb{R}_+^{E(K_n)}$ be a metric; that is, $c_{ij} + c_{jk} \geq c_{ik}$ for all distinct $i, j, k \in V(K_n)$. Then the problem $\min\{c^T x : x \in \mathrm{TSP}(n)\}$ is a metric TSP instance. Papadimitriou and Vempala [18] showed that 1.01-approximation for metric TSP is $\mathcal{NP}$-hard.

For each $n \geq 3$, let $\mathcal{C}_n$ denote the set $\{c \in \mathbb{R}_+^{E(K_n)} : c \text{ is a nonzero metric}\}$. For every integer $r \geq 0$, define

$$\rho_r := \sup_{n\geq 3} \alpha_{n,r},$$

where

$$\alpha_{n,r} := \sup_{c\in\mathcal{C}_n} \frac{\min\{c^T x : x \in \mathrm{TSP}(n)\}}{\min\{c^T x : x \in N_+^r(\mathrm{SEP}(n))\}}.$$

The quantity ρ_0 is often known as the integrality gap of the Dantzig–Fulkerson–Johnson relaxation for metric TSP. The next result is well known.

Theorem 9 (Shmoys and Williamson [19], Wolsey [21]). $\rho_0 \leq \frac{3}{2}$.

It is not known if ρ_0 is indeed equal to $\frac{3}{2}$. In fact, the problem of determining the exact value of ρ_0 has been open for more than a decade. It is known that $\rho_0 \geq \frac{4}{3}$ and the following has been conjectured.

Conjecture 10. $\rho_0 = \frac{4}{3}$.

Partial results supporting the conjecture have been obtained by Boyd and Carr [2], Boyd and Labonté [3], Carr and Vempala [4], and Goemans [9].

In this section, we show the following.

Theorem 11. *$\rho_r \geq \frac{4}{3}$ for every $r \geq 0$.*

Proof. Let G_l be the graph obtained from H_{r+1} by replacing each of v_0v_1, v_2v_3, and v_4v_5 by a path of length l. Since $x^{r+1} \in N_+^r(\mathrm{SEP}(H_{r+1}))$ (Theorem 8), by Lemma 6, one can obtain a point $\tilde{x} \in N_+^r(\mathrm{SEP}(G_l))$ from x^{r+1} by subdividing 1-edges.

Let G_l' be the complete graph with vertex-set $V(G_l)$. Let $\bar{x} \in \mathbb{R}^{E(G_l')}$ be such that $\bar{x}_e = \tilde{x}_e$ for all $e \in E(G_l)$ and $\bar{x}_e = 0$ for all $e \notin E(G_l)$; by Lemma 5, $\bar{x} \in N_+^r(\mathrm{SEP}(G_l'))$.

Let $\bar{c} \in \mathbb{R}^{E(G_l')}$ be such that for each $e \in E(G_l)$, $\bar{c}_e = 1$, and for each $e = uw \notin E(G_l)$, $\bar{c}_e$ is equal to the length of the shortest path between u and w in G_l. Clearly, $\bar{c}$ is metric. We now establish the following claim.

Claim 3. $\min\{\bar{c}^T x : x \in \mathrm{TSP}(G_l')\} \geq 4l$.

Proof. Equivalently, we show that the number of edges in every Eulerian spanning subgraph of G_l is at least $4l$. (A spanning subgraph is Eulerian if it is connected and the degree at each vertex is even. In this definition, we allow edges to be used more than once.) We prove this by induction on l.

The case when $l = 1$ is obvious. Suppose $l \geq 2$. Assume that the claim is true for $l - 1$. Consider any three edges f_1, f_2, f_3, one from each of the three length-l paths in G_l. Since $\{f_1, f_2, f_3\}$ is a cut of G_l, any Eulerian spanning subgraph T of G_l must use an even number of edges from $\{f_1, f_2, f_3\}$, counting multiplicity. Since $l \geq 2$, there exist f_1, f_2, f_3 such that T uses at least four edges from $\{f_1, f_2, f_3\}$. Let H be the graph obtained from G_l by contracting the edges f_1, f_2, f_3. Let T' be the image of T after contraction. Note that T' is a Eulerian spanning subgraph of H. However, H is isomorphic to G_{l-1}. By induction, T' must have at least $4(l-1)$ edges. Hence, T must have at least $4l$ edges. The claim now follows. □

Since $\bar{c}^T\bar{x} = 3l + 3(2r+3)$, we have $\alpha_{|V(G_l)|,r} \geq \frac{4l}{3l+3(2r+3)}$. Hence,

$$\rho_r \geq \sup_{l\geq 1} \alpha_{|V(G_l)|,r} \geq \sup_{l\geq 1} \frac{4l}{3l+3(2r+3)} = \frac{4}{3}. \qquad \square$$

Remark. It is not difficult to modify the above proof to show that, for sufficiently large n, the integrality gap of $N_+^m(\mathrm{SEP}(n))$ is at least $\frac{4}{3} - o(1)$ provided that $m = o(n)$.

5. Concluding remarks.

Theorem 8 shows that 2-matching inequalities do not have bounded N_+-rank. Nevertheless, an upper bound depending on the size of the handle and the number of teeth can be given. Consider a graph G and a 2-matching inequality $\mathcal{I}$ with handle H and teeth $T_1, \ldots, T_s$, where $s \geq 3$ is odd. An upper bound on the N_+-rank of $\mathcal{I}$ is $|H| + (s-1)/2$. To show this, we use the following result from Goemans and Tunçel (Theorem 3.6 in [10]).

THEOREM 12. *Let $P \subseteq [0,1]^d$, $\alpha \in \mathbb{R}$, and $a \in \mathbb{R}^d$ be such that $a \geq 0$. Let $I_+ = \{i : a_i > 0\}$. If $a^T x \leq \alpha$ is valid for $P \cap \{x : x_i = 1$ for all $i \in I\}$ for all sets $I \subseteq I_+$ satisfying either of the following two conditions:*

(1) $|I| = r$,

(2) $|I| \leq (r-1)$ *and* $\sum_{i\in I} a_i > \alpha$,

then $a^T x \leq \alpha$ is valid for $N_+^r(P)$.

For the inequality $\mathcal{I}$, we have $I_+ = \gamma(H) \cup \gamma(T_1) \cup \cdots \cup \gamma(T_s)$. Let $r = |H| + (s-1)/2$. Observe that there is no $I \subseteq I_+$ satisfying (2) in the theorem. Let $I \subseteq I_+$ be such that $|I| = r$. Let $\bar{x} \in \mathrm{SEP}(G) \cap \{x : x_i = 1$ for all $i \in I\}$. The graph $G' = (H, I \cap \gamma(H))$ is a union of disjoint paths. (Some paths may have zero length.) Let n_0, n_1, and n_2 denote the number of these paths having 0, 1, or 2 end-vertices in $H \cap (T_1 \cup \cdots \cup T_s)$, respectively. (Paths of length zero have only one end-vertex.) As G' is a forest, $|I \cap \gamma(H)| = |H| - n_0 - n_1 - n_2$. Clearly, $|I \backslash \gamma(H)| \leq n_1 + 2n_2$. Hence, $r = |I| \leq |H| - n_0 - n_1 - n_2 + n_1 + 2n_2 = |H| - n_0 + n_2 \leq r$. We must have equality throughout, implying that $n_0 = 0$, $n_2 = (s-1)/2$, and $|I\backslash\gamma(H)| = n_1 + 2n_2$. It follows that $n_1 \in \{0,1\}$ and $|I\backslash\gamma(H)| = n_1 + s - 1$. Note that $\bar{x}_e = 0$ for all $e \in I_+\backslash I$. (This follows from $\bar{x}(\delta(v)) = 2$ for all v and the following: If $n_1 = 0$, then $|I \cap \delta(v)| = 2$ for all $v \in H$; and if $n_1 = 1$, then $\bar{x}_e = 1$ for all $I\backslash\gamma(H)$, implying that only one vertex in H is not incident with two edges in I.) Hence, $\bar{x}(I_+) = |I| = r$, implying that $\mathcal{I}$ is valid for $\mathrm{SEP}(G) \cap \{x : x_i = 1$ for all $i \in I\}$. By Theorem 12, $\mathcal{I}$ is valid for $N_+^r(\mathrm{SEP}(G))$.

Padberg and Rao [17] gave a polynomial-time algorithm for separating 2-matching inequalities. Recently, Fleischer, Letchford, and Lodi [8] gave a polynomial-time separation algorithm for a special class of comb inequalities that includes 2-matching inequalities. If $\mathrm{SC}(n)$ denotes the set of points in $\mathrm{SEP}(n)$ satisfying all such comb inequalities, is $\mathrm{SC}(n) \subseteq N_+(\mathrm{SEP}(n))$? We show that the answer is "no." Let $k \geq 4$ be an integer. The point x shown in Figure 12 is in $\mathrm{SC}(4k)$. (See [8, 14] for details.)

By a result of Lovász and Schrijver [16], if $x \in N_+(\mathrm{SEP}(4k))$, then x is a convex combination of some $\tilde{x}, \hat{x} \in \mathrm{SEP}(4k)$ with $\tilde{x}_{\{1,2\}} = 1$ and $\hat{x}_{\{1,2\}} = 0$. Since $\tilde{x}_e = 1$ whenever $x_e = 1$ and $\tilde{x}(\delta(v)) = 2$ for all v, we have $\tilde{x}_{\{2,6\}} = \tilde{x}_{\{3,7\}} = 0$, implying that $\tilde{x}_{\{6,7\}} = \tilde{x}_{\{7,11\}} = \tilde{x}_{\{6,10\}} = \tilde{x}_{\{10,11\}} = 1$. But this contradicts that $\tilde{x} \in \mathrm{SEP}(4k)$.

Now, it might be more desirable (at least theoretically) to consider using $\mathrm{SC}(n)$ instead of $\mathrm{SEP}(n)$ to obtain lower bounds for metric TSP in polynomial time.

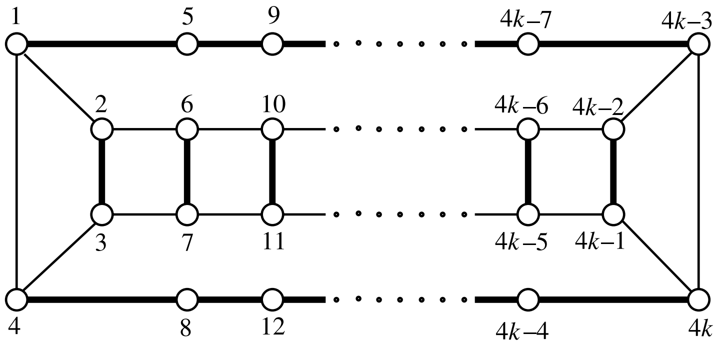


FIG. 12. *Fractional point in* SEP$(4k)$. *Thick edges indicate* $x_e = 1$ *and thin edges indicate* $x_e = 1/2$.

However, using the point x above, Fleischer, Letchford, and Lodi [8] showed that

$$\rho := \sup_{n\geq 3} \sup_{c\in\mathcal{C}_n} \frac{\min\{c^T x : x \in \mathrm{TSP}(n)\}}{\min\{c^T x : x \in \mathrm{SC}(n)\}} \geq \frac{4}{3}.$$

If Conjecture 10 is true, then the integrality gap ρ is $\frac{4}{3}$ as well. It might be interesting to determine if

$$\sup_{n\geq 3} \sup_{c\in\mathcal{C}_n} \frac{\min\{c^T x : x \in \mathrm{TSP}(n)\}}{\min\{c^T x : x \in N_+^r(\mathrm{SC}(n))\}} \geq \frac{4}{3}$$

for every integer $r > 0$.

Acknowledgments. The author would like to thank Levent Tunçel for stimulating conversations and the anonymous referees for their comments and suggestions that helped improve the quality of this paper.

REFERENCES

[1] S. Arora, B. Bollobás, and L. Lovász, *Proving integrality gaps without knowing the linear program,* in Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Los Alamitos, CA, 2002, pp. 313–322.

[2] S. Boyd and R. Carr, *Finding Low Cost TSP and* 2*-Matching Solutions Using Certain Half-Integer Subtour Vertices,* Technical report TR-96-12, University of Ottawa, Ottawa, 2002.

[3] S. Boyd and G. Labonté, *Finding the exact integrality gap for small traveling salesman problems,* in Proceedings of the 9th Conference on Integer Programming and Combinatorial Optimization, Lecture Notes in Comput. Sci. 2337, W. J. Cook and A. S. Schulz, eds., Springer-Verlag, Berlin, 2002, pp. 83–92.

[4] R. Carr and S. Vempala, *Towards a* 4/3 *approximation for the asymmetric traveling salesman problem,* in Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, ACM, New York, SIAM, Philadelphia, 2000, pp. 116–125.

[5] W. Cook and S. Dash, *On the matrix-cut rank of polyhedra,* Math. Oper. Res., 26 (2001), pp. 19–30.

[6] G. B. Dantzig, D. R. Fulkerson, and S. M. Johnson, *Solution of a large-scale traveling-salesman problem,* J. Operations Res. Soc. Amer., 2 (1954), pp. 393–410.

[7] J. Edmonds, *Maximum matching and polyhedron with* 0, 1 *vertices,* J. Res. Nat. Bur. Standards Sect. B, 69B (1965), pp. 125–130.

[8] L. K. Fleischer, A. N. Letchford, and A. Lodi, *Polynomial-Time Separation of a Superclass of Simple Comb Inequalities* IBM Research Report RC23250, 2004. Available online at http://www.lancs.ac.uk/staff/letchfoa/simple_dp.pdf.

[9] M. X. Goemans, *Worst-case comparison of valid inequalities for the TSP,* Math. Programming Ser. A., 69 (1995), pp. 335–349.

[10] M. X. Goemans and L. Tunçel, *When does the positive semidefiniteness constraint help in lifting procedures?*, Math. Oper. Res., 26 (2001), pp. 796–815.

[11] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin, 1988.

[12] M. Grötschel and M. Padberg, *On the symmetric travelling salesman problem* II: *Lifting theorems and facets,* Math. Programming, 16 (1979), pp. 281–302.

[13] M. Laurent, *Tighter linear and semidefinite relaxations for MAX-CUT based on the Lovász–Schrijver lift-and-project procedure,* SIAM J. Optim., 12 (2001), pp. 345–375.

[14] A. N. Letchford and A. Lodi, *Polynomial-time separation of simple comb inequalities*, in Proceedings of the 9th Conference on Integer Programming and Combinatorial Optimization, Lecture Notes in Comput. Sci. 2337, W. J. Cook and A. S. Schulz, eds., Springer-Verlag, Berlin, 2002, 93–108.

[15] L. Lipták and L. Tunçel, *The stable set problem and the lift-and-project ranks of graphs,* Math. Programming Ser. B, 98 (2003), pp. 319–353.

[16] L. Lovász and A. Schrijver, *Cones of matrices and set-functions and* 0-1 *optimization,* SIAM J. Optim., 1 (1991), pp. 166–190.

[17] M. W. Padberg and M. R. Rao, *Odd minimum cut-sets and b-matchings,* Math. Oper. Res., 7 (1982), pp. 67-80.

[18] C. H. Papadimitriou and S. Vempala, *On the approximability of the traveling salesman problem,* in Proceedings of the 32nd Annual ACM Symposium on Theory of Computing, ACM, New York, 2000, pp. 126–133.

[19] D. B. Shmoys and D. P. Williamson, *Analyzing the Held–Karp TSP bound: A monotonicity property with application,* Inform. Process. Lett., 35 (1990), pp. 281–285.

[20] T. Stephen and L. Tunçel, *On a representation of the matching polytope via semidefinite liftings,* Math. Oper. Res., 24 (1999), pp. 1–7.

[21] L. A. Wolsey, *Heuristic analysis, linear programming and branch and bound,* Math. Programming Stud., 13 (1980), pp. 121–134.

SIAM J. Optim.
Vol. 16, No. 2, pp. 400–417

AN $O(\sqrt{n}L)$ ITERATION PRIMAL-DUAL PATH-FOLLOWING METHOD, BASED ON WIDE NEIGHBORHOODS AND LARGE UPDATES, FOR MONOTONE LCP[*]

WENBAO AI[†] AND SHUZHONG ZHANG[‡]

Abstract. In this paper we propose a new class of primal-dual path-following interior point algorithms for solving monotone linear complementarity problems. At each iteration, the method would select a target on the central path with a large update from the current iterate, and then the Newton method is used to get the search directions, followed by adaptively choosing the step sizes, which are, e.g., the largest possible steps before leaving a neighborhood that is as wide as the $\mathcal{N}_\infty^-$ neighborhood. The only deviation from the classical approach is that we treat the classical Newton direction as the sum of two other directions, corresponding to, respectively, the negative part and the positive part of the right-hand side. We show that if these two directions are equipped with different and appropriate step sizes, then the method enjoys the low iteration bound of $O(\sqrt{n}\log L)$, where n is the dimension of the problem and $L = \frac{(x^0)^T s^0}{\varepsilon}$ with ε the required precision and (x^0, s^0) the initial interior solution. For a predictor-corrector variant of the method, we further prove that, besides the predictor steps, each corrector step also reduces the duality gap by a rate of $1 - 1/O(\sqrt{n})$. Additionally, if the problem has a strict complementary solution, then the predictor steps converge Q-quadratically.

Key words. monotone linear complementarity problem, primal-dual interior point method, wide neighborhood

AMS subject classifications. 90C33, 90C51, 90C05

DOI. 10.1137/040604492

1. Introduction.

In this paper we consider the following monotone linear complementarity problem (LCP):

$$(LCP) \qquad \begin{cases} s = Mx + q, \\ x \geq 0,\, s \geq 0,\, x^T s = 0, \end{cases}$$

where $q \in \Re^n$ and $M \in \Re^{n \times n}$ is a monotone matrix, i.e., $M + M^T$ is positive semidefinite, or equivalently, $x^T M x \geq 0$ for any $x \in \Re^n$.

A particular choice of M is a block skew symmetric matrix, $M = \left[\begin{smallmatrix} 0 & A \\ -A^T & 0 \end{smallmatrix}\right]$. In that case, the corresponding monotone LCP problem is nothing but a linear programming problem.

The primal-dual interior point method for linear programming was first introduced by Kojima, Mizuno, and Yoshise [5] and Megiddo [7], which essentially aims at solving the following parameterized problem by Newton's method, for shrinking values of the parameter $\mu > 0$:

$$(LCP)_\mu \qquad \begin{cases} s = Mx + q, \\ x_i > 0,\, s_i > 0,\, x_i s_i = \mu,\, i = 1, \dots, n. \end{cases}$$

[*] Received by the editors February 26, 2004; accepted for publication (in revised form) February 18, 2005; published electronically October 17, 2005. This research was supported by Hong Kong RGC Earmarked grant CUHK4174/03E.

http://www.siam.org/journals/siopt/16-2/60449.html

[†] School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, People's Republic of China (wenbaoai@vip.sina.com).

[‡] Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, Hong Kong (zhang@se.cuhk.edu.hk; http://www.se.cuhk.edu.hk/∼zhang).

The exact solution of the above problem is known as the analytic central path, with the varying path parameter $\mu > 0$. At each iteration, the method would choose a target on the central path and apply the Newton method to move closer to the target, while confining the iterate to stay within a certain neighborhood of the analytic central path. This method was found to be not only elegant in its simplicity and symmetricity, but also extremely efficient in practical implementations. There has been, however, an inconsistency between theory and practice: fast algorithms in practice may actually render worse complexity bounds. In their first paper [5], Kojima, Mizuno, and Yoshise proposed that the iterates reside in a *wide* neighborhood of the central path, known as the $\mathcal{N}_\infty^-$-neighborhood (details of the notion will be discussed later), and the targets on the central path are shifted towards the origin by a large update (percentage reduction) at each iteration. The worst case iteration bound was proved to be $O(nL)$, where n is the larger dimension of a standard linear programming problem, and L is its input-length. Then, in a subsequent paper, [6], the same authors proposed a variant of the method, where the iterates are restricted to a much smaller neighborhood, known as the $\mathcal{N}_2$-neighborhood, and at each step the target is shifted with a small update. The algorithm became too conservative to be efficient in practice. However, the worst case iteration bound of the variant was improved to $O(\sqrt{n}L)$. In fact, many early primal-dual interior point methods either use a narrow neighborhood, or take small step sizes; see, e.g., the primal-dual method by Monteiro and Adler [9, 10]. The first practically efficient $O(\sqrt{n}L)$ primal-dual interior point algorithm was the celebrated predictor-corrector algorithm of Mizuno, Todd, and Ye [8]. In the predictor step of the algorithm, an adaptive step size is taken, ensuring its practical efficiency. The iteration bound is still retained to be $O(\sqrt{n}L)$ since the $\mathcal{N}_2$ small neighborhoods are used to control the centrality of the iterates. Gonzaga [2] proposed to compute and combine the predictor and corrector steps based on the information of the same iterate, thus reducing the effort required by the Cholesky factorization. Along a related but different line, Ye et al. [18] proved that the predictor step in the predictor-corrector scheme reduces the duality gap with a quadratic convergence rate. This result was extended by Ye and Anstreicher in [17] to the monotone LCP problem, assuming a strict complementary solution exists. We refer the reader to Wright [14] for an excellent exposition on the primal-dual interior point method for linear programming and LCP problems.

The issue of the neighborhood size in the method has generated some research interests on its own. It is believed that in its original form, the primal-dual interior point algorithm of Kojima, Mizuno, and Yoshise [5] may indeed not enjoy the low iteration bound of $O(\sqrt{n}L)$. However, it is possible to modify the algorithm to gain both the theoretical and the practical advantages. A first such attempt was made by Xu [15], who proposed an $O(\sqrt{n}L)$ method, in which the small neighborhood was only used as a safeguard, and the iterates are allowed to go far beyond. However, the new neighborhood, though much larger than the small neighborhood, does not necessarily contain the wide neighborhood: the neighborhood is still much narrower than the $\mathcal{N}_\infty^-$ wide neighborhood. Hung and Ye [4] proposed to use higher-order corrections on the Newton method, and showed that the iteration bound of their high order primal-dual interior point method with the $\mathcal{N}_\infty^-$ wide neighborhood can be reduced to $O(n^{\frac{n+1}{2n}}L)$. Sturm and Zhang [13] proposed to follow a central region, instead of the central path. The central region is defined to be precisely an $\mathcal{N}_\infty^-$ wide neighborhood of the central path. Then, they introduced a (narrow) neighborhood of the central region (the whole area is thus wider than the central region which is a wide neighborhood itself) in which all the iterates reside. By choosing the direction towards a target in the central region properly, Sturm and Zhang [13] managed to show that their algorithm has an iteration bound of $O(\sqrt{n}L)$. Since the iterates are required to take adaptive (thus long) steps, maximum possible within the wide neighborhood, the algorithm is highly efficient in practice. Later, Sturm generalized the method to solve semidefinite programming (SDP) problems, and the method has become one of the pillars for his famous SDP solver SeDuMi [12]. Ai [1] proposed a new wide neighborhood interior point algorithm with $O(\sqrt{n}L)$ iteration bound. The current paper is inspired by [1], in the definition of the new wide neighborhood; however, they differ greatly in both the scope and the results to be achieved. Recently, Peng, Terlaky, and Zhao [11] introduced a variant of the predictor-corrector approach, based on a self-regular function to define the neighborhood, which is wide. They showed that their algorithm enjoys the iteration bound of $O(\sqrt{n}\log nL)$ for linear programming problems.

As far as we know, in the context of the path-following approach, none had succeeded in retaining the $O(\sqrt{n}L)$ complexity while allowing a large update (meaning a reduction by a universal percentage, independent of the problem parameters) of the target along the central path at all iterations, even if one is allowed to stay within narrow neighborhoods. Indeed, deriving a path-following interior point method with the $O(\sqrt{n}L)$ iteration bound while working with large updates and wide neighborhood at each iteration is one of the objectives to be achieved in this paper. In other words, the current paper aims at modifying the original primal-dual interior point method with minimum changes, maintaining large updates and working with wide neighborhood at all iterations, to gain the $O(\sqrt{n}L)$ iteration bound and retain practical efficiency. We organize the paper as follows. Our new methodology will be introduced in section 2, where the main underlying ideas will be explained. In section 3, we present the technical lemmas that will be important for the subsequent analysis, and in section 4 we discuss some easily implementable variants of the general method and show the low computational complexity status. In a similar spirit, we discuss another variant in section 5, based on the predictor-corrector methodology. Novel properties of the algorithm will be discussed, including its progressive property even for the corrector steps. The low complexity bound will be proven for this variant, and the superlinear convergence property will be shown, provided that a strictly complementary solution exists.

The notation used in this paper is fairly standard: the ith component of vector $x \in \Re^n$ is denoted by x_i; e is the all one vector with an appropriate dimension; if $d \in \Re^n$, then we denote D to be an $n \times n$ diagonal matrix with d as the diagonal components; for $x, y \in \Re^n$, xy is the component product in $\Re^n$, and so is true for other operations, e.g., $1/(xy)$ and $(xy)^{-0.5}$; $x \geq (>)\, y$ means that the inequality holds componentwise; for any $a \in \Re$, a^+ denotes its nonnegative part, i.e., $a^+ := \max\{a, 0\}$, and a^- denotes its nonpositive part, i.e., $a^- := \min\{a, 0\}$; the same notation is used for vector $x \in \Re^n$, namely x^+ is the nonnegative part of x and x^- is the nonpositive part of x; the L_p-norm of $x \in \Re^n$ is denoted by $\|x\|_p$, and in particular we write $\|x\|$ for $\|x\|_2$—the Euclidean norm.

2. Separating large and small components: A new paradigm.

Let us denote

$$\mathcal{F}_{++} := \{(x, s) \mid s = Mx + q,\ x > 0,\ s > 0\},$$

which is assumed to be nonempty throughout this paper.

The central path for (LCP) is defined as

$$\mathcal{C} := \{(x, s) \in \mathcal{F}_{++} \mid xs = \mu e\}$$

and its small neighborhood is defined as

$$\mathcal{N}_2(\beta) := \{(x,s) \in \mathcal{F}_{++} \mid \|xs - \mu e\| \leq \beta\mu\},$$

where $\beta \in (0,1)$ is a given constant and $\mu := x^T s/n$. The so-called wide neighborhood is defined as follows:

$$\mathcal{N}_\infty^-(1-\tau_2) := \{(x,y,s) \in \mathcal{F}_{++} \mid xs \geq \tau_2 \mu e\},$$

where $0 < \tau_2 < 1$.

Before proceeding, let us recall the classical primal-dual interior point method with wide neighborhood and large update of the targets. Let $0 \leq \tau \leq 1$ and $0 < \tau_2 < 1$ be two given parameters. Suppose that the current iterate is $(x,s) \in \mathcal{N}_\infty^-(1-\tau_2)$. The search direction $(\Delta x, \Delta s)$ is the solution of the following system of linear equations:

$$(1) \qquad \begin{cases} \Delta s = M\Delta x, \\ s\Delta x + x\Delta s = \tau\mu e - xs, \end{cases}$$

where

$$\mu = x^T s/n.$$

Then the next iterate will be given by

$$(x + \bar{\alpha}\Delta x, s + \bar{\alpha}\Delta s),$$

where $\bar{\alpha}$ is the solution of the subproblem

$$\begin{array}{ll} \text{minimize} & (x+\alpha\Delta x)^T(s+\alpha\Delta s) \\ \text{subject to} & (x+\alpha\Delta x, s+\alpha\Delta s) \in \mathcal{N}_\infty^-(1-\tau_2) \\ & \alpha \in [0,1]. \end{array}$$

Naturally, all the iterates are contained in the wide neighborhood $\mathcal{N}_\infty^-(1-\tau_2)$.

An important ingredient of this paper is to introduce a new neighborhood for the central path, defined as

$$(2) \quad \mathcal{N}(\tau_1,\tau_2,\eta) := \mathcal{N}_\infty^-(1-\tau_2) \bigcap \left\{(x,s) \in \mathcal{F}_{++} \mid \left\|(\tau_1\mu e - xs)^+\right\| \leq \eta(\tau_1-\tau_2)\mu\right\},$$

where $\eta \geq 1$ and τ_1 satisfying $0 < \tau_2 < \tau_1 < 1$ are two more parameters.

The above defined neighborhood is itself a wide neighborhood, since one can easily verify that

$$\mathcal{N}_\infty^-(1-\tau_1) \subseteq \mathcal{N}(\tau_1,\tau_2,\eta) \subseteq \mathcal{N}_\infty^-(1-\tau_2).$$

Moreover, if $\tau_1 - \eta(\tau_1-\tau_2)/\sqrt{n-1} > \tau_2$, then

$$\mathcal{N}_\infty^-\left(1-\tau_1+\eta(\tau_1-\tau_2)/\sqrt{n-1}\right) \subseteq \mathcal{N}(\tau_1,\tau_2,\eta),$$

and if $\tau_1 - \eta(\tau_1-\tau_2)/\sqrt{n-1} \leq \tau_2$, then

$$\mathcal{N}(\tau_1,\tau_2,\eta) = \mathcal{N}_\infty^-(1-\tau_2).$$

Specially, if we choose $\eta = 1$, the neighborhood can be expressed more simply as follows:

$$(3) \qquad \mathcal{N}(\tau_1,\tau_2,1) = \{(x,s) \in \mathcal{F}_{++} \mid \left\|(\tau_1\mu e - xs)^+\right\| \leq (\tau_1-\tau_2)\mu\} \supseteq \mathcal{N}_\infty^-(1-\tau_1).$$

In this paper, the newly introduced neighborhood $\mathcal{N}(\tau_1, \tau_2, \eta)$ will play an important role. The reason for working with $\mathcal{N}(\tau_1, \tau_2, \eta)$ is that the measure for the components in xs that are "dangerously" close to zero are captured by the quantity $\|(\tau_1 \mu e - xs)^+\|$, and we are less concerned about the "large" components of xs present in $(\tau_1 \mu e - xs)^-$. In fact, we will see later that this separation is crucial. The part $(\tau_1 \mu e - xs)^+$ is used to control the centrality, and the other part $(\tau_1 \mu e - xs)^-$ is important for the progress towards optimality.

Suppose that our current iterate is (x, s). Let $0 \leq \tau \leq 1$. Another key ingredient of our method is to decompose the Newton step, from xs to the target on the central path $\tau \mu e$ (large update), into the following two equations:

$$\left\{ \begin{array}{l} \Delta s_- = M \Delta x_-, \\ s \Delta x_- + x \Delta s_- = (\tau \mu e - xs)^- \end{array} \right. \tag{4}$$

and

$$\left\{ \begin{array}{l} \Delta s_+ = M \Delta x_+, \\ s \Delta x_+ + x \Delta s_+ = (\tau \mu e - xs)^+. \end{array} \right. \tag{5}$$

Since $\tau \mu e - xs = (\tau \mu e - xs)^- + (\tau \mu e - xs)^+$, the usual Newton direction is simply $(\Delta x_-, \Delta s_-) + (\Delta x_+, \Delta s_+)$. In this paper, however, we propose to treat these two directions separately. Essentially those are what we need to modify the original large update and wide neighborhood path-following method. The payoff for the changes will become clear later. At this stage, we only remark that the extra computational effort is very marginal, compared to the computation of a single Newton direction. For reference, we shall call $(\Delta x_-, \Delta s_-)$ and $(\Delta x_+, \Delta s_+)$ the *Newton constituent directions.*

Let $\alpha := (\alpha_1, \alpha_2) \in \Re^2_+$ be the step sizes taken along $(\Delta x_-, \Delta s_-)$ and $(\Delta x_+, \Delta s_+)$, respectively. The step is

$$(x(\alpha), s(\alpha)) := (x, s) + \alpha_1 (\Delta x_-, \Delta s_-) + \alpha_2 (\Delta x_+, \Delta s_+).$$

The best α can be obtained by solving the following two-dimensional optimization problem:

$$\begin{array}{ll} \text{minimize} & x(\alpha)^T s(\alpha) \\ \text{subject to} & (x(\alpha), s(\alpha)) \in \mathcal{N}(\tau_1, \tau_2, \eta) \\ & 0 \leq \alpha_1 \leq 1,\ 0 \leq \alpha_2 \leq 1. \end{array} \tag{6}$$

Remark that due to the monotonicity, the above objective function $x(\alpha)^T s(\alpha)$ is convex in α.

Below we describe a generic framework for our wide-neighborhood and large-update primal-dual path-following method.

Algorithm 2.1.

Input parameters: required precision $\varepsilon > 0$*, neighborhood parameters* $\eta^k \geq 1$, $0 < \tau_2^k < \tau_1^k < 1$*, and target parameters* $0 \leq \tau^k \leq 1$, $k = 0, 1, \ldots,$ *and the initial solution* $(x^0, s^0) \in \mathcal{N}(\tau_1^0, \tau_2^0, \eta^0)$.

Output: a sequence of iterates $\{(x^k, s^k) \mid k = 0, 1, 2, \ldots\}$.

Step 0. *Set* $k = 0$.

Step 1. *If* $(x^k)^T s^k \leq \varepsilon$*, then stop.*

Step 2. *Solve* $(\Delta x_-^k, \Delta s_-^k)$ *and* $(\Delta x_+^k, \Delta s_+^k)$ *based on* (4) *and* (5).

Find step size vector $\alpha^k \in \Re^2_{++}$*, such that* $(x(\alpha^k), s(\alpha^k)) \in \mathcal{N}(\tau_1^k, \tau_2^k, \eta^k)$.

Step 3. *Set* $(x^{k+1}, s^{k+1}) := (x(\alpha^k), s(\alpha^k))$.
Let $k := k + 1$ *and go to* **Step 1**.

We remark here that the optimal step sizes according to (6) may be used in **Step 2**, and the parameters η^k, τ_1^k, τ_2^k, and τ^k may be set to constants. It is, however, convenient to allow for the flexibilities at this stage.

The main result of this paper is to prove that the above generic method can be specified into easy implementable variants with given parameters, in such a way that the iteration bound will be $O(\sqrt{n}\log\frac{(x^0)^T s^0}{\varepsilon})$. Moreover, the method can also be implemented in the predictor-corrector style. In that case, in addition to the above iteration bound one also obtains a quadratic convergence rate for the predictor steps, provided that a strict complementary solution exists. These specific implementations will be discussed in sections 4 and 5, respectively. To facilitate the analysis, we need to study the properties of the two separated Newton constituent directions. This will be the topic of the next section.

3. Technical lemmas.

In this section we choose to set $\tau = \tau_1$.

First, it is useful for our subsequent analysis to note the following triangle inequalities for the "minus" and "plus" operations on the vectors.

PROPOSITION 3.1. *For any* $u, v \in \Re^n$ *and* $p \geq 1$, *we have*

$$\|(u+v)^+\|_p \leq \|u^+\|_p + \|v^+\|_p$$

and

$$\|(u+v)^-\|_p \leq \|u^-\|_p + \|v^-\|_p.$$

Proof. As $0 \leq (u+v)^+ \leq u^+ + v^+$ we conclude that

$$\|(u+v)^+\|_p \leq \|u^+ + v^+\|_p \leq \|u^+\|_p + \|v^+\|_p.$$

Similarly, we have

$$\|(u+v)^-\|_p \leq \|u^-\|_p + \|v^-\|_p. \qquad \square$$

The next proposition is concerned with the feasibility of the iterates along given Newton directions. It would be undesirable if the iterates would leave the feasible region and then return to it again.

PROPOSITION 3.2. *Suppose that* $(x, s) \in \mathcal{F}_{++}$ *and* $z + xs \geq 0$. *Let* $(\Delta x, \Delta s)$ *be the solution of* $\Delta s = M\Delta x$, $s\Delta x + x\Delta s = z$. *If* $(x + t_0\Delta x)(s + t_0\Delta s) > 0$ *for some* $0 < t_0 \leq 1$, *then* $x + t\Delta x > 0$ *and* $s + t\Delta s > 0$ *for all* $0 \leq t \leq t_0$.

Proof. Let $(\bar{x}, \bar{s}) := (x + t_0\Delta x, s + t_0\Delta s)$.

We have

$$\begin{aligned} &(x + \delta t_0\Delta x)(s + \delta t_0\Delta s) \\ &= xs + \delta t_0(s\Delta x + x\Delta s) + \delta^2 t_0^2 \Delta x\Delta s \\ &= xs + \delta t_0 z + \delta^2(\bar{x}\bar{s} - t_0 z - xs) \\ &= (1-\delta)xs + \delta(1-\delta)(t_0 z + xs) + \delta^2\bar{x}\bar{s} > 0 \end{aligned} \tag{7}$$

for all $0 \leq \delta \leq 1$.

If there are $0 < t_1 \leq t_0$ and $1 \leq i \leq n$ with either $(x + t_1\Delta x)_i < 0$ or $(s + t_1\Delta s)_i < 0$, then, since $(x, s) > 0$, by continuity there must exist $0 < t_2 < t_1 \leq t_0$, such that

$(x + t_2 \Delta x)_i (s + t_2 \Delta s)_i = 0$. Letting $\delta = t_2/t_0$, we have $(x + \delta t_0 \Delta x)_i (s + \delta t_0 \Delta s)_i = 0$, which would contradict (7). The proposition is thus proven. □

The term $z + xs$ is sometimes called the *target* to be tracked, and it is naturally nonnegative for most interior point methods. In particular, in Algorithm 2.1, this property boils down to verifying $xs + \alpha_1(\tau_1 \mu e - xs)^- + \alpha_2(\tau_1 \mu e - xs)^+ \geq 0$.

Let us denote

$$h(\alpha) := xs + \alpha_1(\tau_1 \mu e - xs)^- + \alpha_2(\tau_1 \mu e - xs)^+ \tag{8}$$

and

$$I^+ := \{i \mid \tau_1 \mu - x_i s_i > 0\}. \tag{9}$$

Since $(x, s) \in \mathcal{F}_{++}$ we have

$$h_i(\alpha) = \begin{cases} x_i s_i + \alpha_2(\tau_1 \mu - x_i s_i) = (1 - \alpha_2) x_i s_i + \alpha_2 \tau_1 \mu > 0, & i \in I^+, \\ x_i s_i + \alpha_1(\tau_1 \mu - x_i s_i) \geq x_i s_i + \tau_1 \mu - x_i s_i = \tau_1 \mu > 0, & i \notin I^+ \end{cases} \tag{10}$$

for all $\alpha \in [0, 1]^2$. Proposition 3.2 thus asserts that $(x(\alpha), s(\alpha)) \in \mathcal{N}(\tau_1, \tau_2, \eta)$ if and only if $x(\alpha)s(\alpha) \geq \tau_2 \mu(\alpha) e$ and $\|(\tau_1 \mu(\alpha) e - x(\alpha)s(\alpha))^+\| \leq \eta(\tau_1 - \tau_2)\mu(\alpha)$, where

$$\mu(\alpha) := x(\alpha)^T s(\alpha)/n. \tag{11}$$

Furthermore, we have

$$\begin{aligned} \mu(\alpha) &:= (x + \Delta x(\alpha))^T (s + \Delta s(\alpha))/n \\ &= \left(x^T s + s^T \Delta x(\alpha) + x^T \Delta s(\alpha) + \Delta x(\alpha)^T \Delta s(\alpha)\right)/n \\ &= \mu + \alpha_1 e^T (\tau_1 \mu e - xs)^-/n + \alpha_2 e^T (\tau_1 \mu e - xs)^+/n + \Delta x(\alpha)^T \Delta s(\alpha)/n, \end{aligned} \tag{12}$$

where $\Delta x(\alpha) = \alpha_1 \Delta x_- + \alpha_2 \Delta x_+$ and $\Delta s(\alpha) = \alpha_1 \Delta s_- + \alpha_2 \Delta s_+$.

LEMMA 3.3. *It holds that*

$$\mu(\alpha) \geq (1 - \alpha_1)\mu.$$

Proof. By the monotonicity we have $\Delta x(\alpha)^T \Delta s(\alpha) \geq 0$. Therefore, from (12) we have

$$\begin{aligned} \mu(\alpha) &\geq \mu + \alpha_1 e^T (\tau_1 \mu e - xs)^-/n \\ &\geq \mu + \alpha_1 e^T (-xs)/n \\ &= (1 - \alpha_1)\mu. \quad \square \end{aligned}$$

We note the following simple but useful relationships:

$$\begin{cases} e^T (\tau_1 \mu e - xs)^- = -(1 - \tau_1) x^T s - e^T (\tau_1 \mu e - xs)^+, \\ \|(xs)^{-0.5} (\tau_1 \mu e - xs)^-\| = \|(\sqrt{xs} - \tau_1 \mu e/\sqrt{xs})^+\| \leq \|\sqrt{xs}\| = x^T s, \\ e^T (\tau_1 \mu e - xs)^+ \leq \sqrt{n} \|(\tau_1 \mu e - xs)^+\|. \end{cases} \tag{13}$$

For convenience we set

$$\beta := \frac{\tau_1 - \tau_2}{\tau_1}. \tag{14}$$

Obviously we have $\beta \in (0,1)$, $\tau_1 - \tau_2 = \beta\tau_1$, and $\tau_2 = (1-\beta)\tau_1$. Let

$$\hat{\eta} = \max\left\{ \frac{\|(\tau_1\mu e - xs)^+\|}{\beta\tau_1\mu}, 1 \right\}. \tag{15}$$

It follows that $\hat{\eta} \leq \eta$ if $(x,s) \in \mathcal{N}(\tau_1, \tau_2, \eta)$.

LEMMA 3.4. *If $\mu(\alpha) \leq \mu$, then it holds that*

$$\|(\tau_1\mu(\alpha)e - h(\alpha))^+\| \leq (1-\alpha_2)\hat{\eta}\beta\tau_1\mu(\alpha).$$

Proof. As $\mu(\alpha) \leq \mu$ it follows from (10) that

$$\tau_1\mu(\alpha) - h_i(\alpha) \leq \begin{cases} \tau_1\mu(\alpha) - \frac{\mu(\alpha)}{\mu}h_i(\alpha) = \frac{\mu(\alpha)}{\mu}(1-\alpha_2)(\tau_1\mu - x_is_i) & \text{if } i \in I^+, \\ 0 & \text{else,} \end{cases}$$

which implies that

$$\|(\tau_1\mu(\alpha)e - h(\alpha))^+\| \leq \frac{\mu(\alpha)}{\mu}(1-\alpha_2)\|(\tau_1\mu - xs)^+\| \leq (1-\alpha_2)\hat{\eta}\beta\tau_1\mu(\alpha). \qquad \square$$

LEMMA 3.5. *Let $u, v \in \Re^n$ be such that $u^Tv \geq 0$, and let $r = u + v$. Then, we have*

$$\|(uv)^-\|_1 \leq \|(uv)^+\|_1 \leq \frac{1}{4}\|r\|^2.$$

Proof. Let the index set J be

$$J := \{i \mid u_iv_i > 0\}.$$

As $u^Tv \geq 0$, we have

$$\|(uv)^-\|_1 \leq \|(uv)^+\|_1 = \sum_{i\in J} u_iv_i \leq \frac{1}{4}\sum_{i\in J}(u_i+v_i)^2 = \frac{1}{4}\sum_{i\in J}(r_i)^2 \leq \frac{1}{4}\|r\|^2. \qquad \square$$

LEMMA 3.6. *Suppose $\beta \leq \frac{1}{2}$ and $\alpha_1 = t\alpha_2\hat{\eta}\sqrt{\frac{\beta\tau_1}{n}}$ for some $t \geq 0$. Then we have*

$$\|(\varDelta x(\alpha)\varDelta s(\alpha))^-\|_1 \leq \|(\varDelta x(\alpha)\varDelta s(\alpha))^+\|_1 \leq (t^2+1)\alpha_2^2\hat{\eta}^2\beta\tau_1\mu/4.$$

Proof. We have

$$s\varDelta x(\alpha) + x\varDelta s(\alpha) = \alpha_1(\tau_1\mu e - xs)^- + \alpha_2(\tau_1\mu e - xs)^+.$$

Multiply both sides of the above equality by $(xs)^{-0.5}$. Denote $u := x^{-0.5}s^{0.5}\varDelta x(\alpha)$, $v := x^{0.5}s^{-0.5}\Delta x(\alpha)$ and $r := (xs)^{-0.5}(\alpha_1(\tau_1\mu e - xs)^- + \alpha_2(\tau_1\mu e - xs)^+)$. So we have $u + v = r$. Notice that $u^Tv = \varDelta x(\alpha)^T\varDelta s(\alpha) \geq 0$. Therefore, by Lemma 3.5 we have

$$\begin{aligned}
&\|(\varDelta x(\alpha)\varDelta s(\alpha))^-\|_1 \\
&\leq \|(\varDelta x(\alpha)\varDelta s(\alpha))^+\|_1 \\
&\leq \frac{1}{4}\|(xs)^{-0.5}\alpha_1(\tau_1\mu e - xs)^- + (xs)^{-0.5}\alpha_2(\tau_1\mu e - xs)^+\|^2 \\
&= \frac{1}{4}\left(\alpha_1^2\|(\sqrt{xs} - \tau_1\mu e/\sqrt{xs})^+\|^2 + \alpha_2^2\|(xs)^{-0.5}(\tau_1\mu e - xs)^+\|^2\right) \\
&\leq \frac{1}{4}\left(\alpha_1^2\|\sqrt{xs}\|^2 + \alpha_2^2\|(\tau_1\mu e - xs)^+\|^2/(\tau_2\mu)\right) \\
&\leq \frac{1}{4}\left(t^2\alpha_2^2\hat{\eta}^2\beta\tau_1\mu + \alpha_2^2\hat{\eta}^2\beta\tau_1\mu\right) \\
&= (t^2+1)\alpha_2^2\hat{\eta}^2\beta\tau_1\mu/4. \qquad \square
\end{aligned} \tag{16}$$

Lemma 3.7. *Suppose that* $\tau_1 \le 1/4$ *and* $\beta \le 1/2$. *If* $\alpha_1 = \alpha_2\hat{\eta}\sqrt{\frac{\beta\tau_1}{n}}$ *and* $\alpha_2 \le 1/\hat{\eta}^2$, *then we have*

$$\mu(\alpha) \le \left(1 - \frac{\hat{\eta}\sqrt{\beta\tau_1}}{10\sqrt{n}}\alpha_2\right)\mu.$$

Proof. By (12), (13), and Lemma 3.6 we have

$$\begin{aligned}\mu(\alpha) &\le \mu + \alpha_1 e^T(\tau_1\mu e - xs)^-/n + \alpha_2 e^T(\tau_1\mu e - xs)^+/n + \|(\Delta x(\alpha)\Delta s(\alpha))^+\|_1/n\\ &\le \mu - \alpha_1(1-\tau_1)\mu + \alpha_2\|e\|\|(\tau_1\mu e - xs)^+\|/n + \|(\Delta x(\alpha)\Delta s(\alpha))^+\|_1/n\\ &\le \mu - \alpha_1(1-\tau_1)\mu + \alpha_2\hat{\eta}\beta\tau_1\mu/\sqrt{n} + \alpha_2^2\hat{\eta}^2\beta\tau_1\mu/2n\\ &\le \mu - 3\alpha_1\mu/4 + 3\alpha_2\hat{\eta}\beta\tau_1\mu/2\sqrt{n}\\ &= \mu - \alpha_2\frac{3\hat{\eta}\sqrt{\beta\tau_1}(1-2\sqrt{\beta\tau_1})\mu}{4\sqrt{n}}\\ &\le \mu - \alpha_2\frac{3\hat{\eta}\sqrt{\beta\tau_1}(1-1/\sqrt{2})\mu}{4\sqrt{n}}\\ &\le \left(1 - \alpha_2\frac{\hat{\eta}\sqrt{\beta\tau_1}}{10\sqrt{n}}\right)\mu. \qquad \square\end{aligned}$$

Lemma 3.8. *Suppose that* $(x,s) \in \mathcal{N}(\tau_1,\tau_2,\eta)$, $\tau_1 \le 1/4$, *and* $\beta \le 1/2$. *If* $\alpha_1 = \alpha_2\hat{\eta}\sqrt{\frac{\beta\tau_1}{n}}$ *and* $\alpha_2 \le 1/\hat{\eta}^2$, *then* $(x(\alpha),s(\alpha)) \in \mathcal{N}(\tau_1,\tau_2,\eta)$.

Proof. By Lemma 3.7 it follows that $\mu(\alpha) \le \mu$. Further, it follows from (10) and Lemma 3.6 that

$$\begin{aligned}x(\alpha)s(\alpha) &= h(\alpha) + \Delta x(\alpha)\Delta s(\alpha)\\ &\ge (\tau_2\mu + \alpha_2(\tau_1-\tau_2)\mu)e - \|(\Delta x(\alpha)\Delta s(\alpha))^-\|e\\ &\ge (\tau_2\mu + \alpha_2\beta\tau_1\mu)e - (\alpha_2^2\hat{\eta}^2\beta\tau_1\mu/2)e\\ &\ge (\tau_2\mu + \alpha_2\beta\tau_1\mu - \alpha_2\beta\tau_1\mu/2)e\\ &\ge \tau_2\mu e\\ &\ge \tau_2\mu(\alpha)e,\end{aligned}$$

which also implies $(x(\alpha),s(\alpha)) > 0$ according to Proposition 3.2. Therefore, $(x(\alpha),s(\alpha)) \in \mathcal{N}_\infty^-(1-\tau_2)$.

At the same time, by Lemma 3.3 we have

$$\mu(\alpha) \ge (1-\alpha_1)\mu \ge \left(1 - \frac{\hat{\eta}\sqrt{\beta\tau_1}}{\sqrt{n}}\right)\mu \ge (1-\sqrt{\beta\tau_1})\mu \ge \mu/2.$$

Using Lemmas 3.4 and 3.6 we obtain

$$\begin{aligned}&\|\,(\tau_1\mu(\alpha)e - x(\alpha)s(\alpha))^+\,\|\\ &= \|\,(\tau_1\mu(\alpha)e - h(\alpha) - \Delta x(\alpha)\Delta s(\alpha))^+\,\|\\ &\le \|(\tau_1\mu(\alpha)e - h(\alpha))^+ + (-\Delta x(\alpha)\Delta s(\alpha))^+\|\\ &\le \|(\tau_1\mu(\alpha)e - h(\alpha))^+\| + \|(\Delta x(\alpha)\Delta s(\alpha))^-\|\\ &\le (1-\alpha_2)\hat{\eta}\beta\tau_1\mu(\alpha) + \alpha_2^2\hat{\eta}^2\beta\tau_1\mu/2\\ &\le (1-\alpha_2)\hat{\eta}\beta\tau_1\mu(\alpha) + \alpha_2\hat{\eta}\beta\tau_1\mu(\alpha)\\ &= \hat{\eta}\beta\tau_1\mu(\alpha)\\ &\le \eta\beta\tau_1\mu(\alpha),\end{aligned}$$

proving that $(x(\alpha), s(\alpha)) \in \mathcal{N}(\tau_1, \tau_2, \eta)$. □

4. The iteration bound and an implementation. Now we are in a position to present our complexity results.

First, let us consider the generic Algorithm 2.1.

THEOREM 4.1. *Suppose that $\eta \geq 1$, $\tau = \tau_1 \leq 1/4$, and $\beta \leq 1/2$ are fixed for all iterations. Furthermore, suppose that the plane-search procedure* (6) *is applied at each iteration of Algorithm* 2.1. *Then, Algorithm* 2.1 *terminates in $O(\sqrt{n} \log \frac{(x^0)^T s^0}{\varepsilon})$ iterations.*

Proof. By Lemma 3.8, at each iteration, if we let $\hat{\alpha} = (\sqrt{\beta\tau_1/n}/\hat{\eta}, 1/\hat{\eta}^2)$, then we have

$$(x(\hat{\alpha}), s(\hat{\alpha})) \in \mathcal{N}(\tau_1, \tau_2, \eta).$$

Furthermore, according to Lemma 3.7 we also have

$$\mu(\hat{\alpha}) \leq \left(1 - \frac{\sqrt{\beta\tau_1}}{10\hat{\eta}\sqrt{n}}\right)\mu \leq \left(1 - \frac{\sqrt{\beta\tau_1}}{10\eta\sqrt{n}}\right)\mu.$$

Therefore, the exact plane search would lead to at least the same amount of reduction in $\mu(\alpha)$, and hence the theorem is proven. □

The plane-search subproblem being an optimization problem with a convex objective and only two variables can be solved relatively easily. However, it is also possible to reduce the number of search variables in the subproblem to only one without sacrificing the practical efficiency too much.

The main observation here is that under some mild conditions, the objective function in the subproblem (6) is monotone with respect to α_1 for any fixed $\alpha_2 \in [0, 1]$.

More precisely, we have the following result.

THEOREM 4.2. *Suppose $(x, s) \in \mathcal{N}_\infty^-(1 - \tau_2)$, $\tau = \tau_1 \leq 1/4$, and $\tau_1 \leq 2\tau_2$ (i.e., $\beta \leq 1/2$). For any fixed $\alpha_2 \in [0, 1]$, $x(\alpha)^T s(\alpha)$ is a monotonically decreasing function in α_1 for $\alpha_1 \in [0, 1]$.*

Proof. We have

$$\begin{aligned} x(\alpha)^T s(\alpha) &= (x + \alpha_1 \Delta x_- + \alpha_2 \Delta x_+)^T (s + \alpha_1 \Delta s_- + \alpha_2 \Delta s_+) \\ &= x^T s + \alpha_1 (x^T \Delta s_- + s^T \Delta x_-) + \alpha_2 (x^T \Delta s_+ + s^T \Delta x_+) \\ &\quad + \alpha_1^2 \Delta x_-^T \Delta s_- + \alpha_1 \alpha_2 \left(\Delta x_-^T \Delta s_+ + \Delta x_+^T \Delta s_-\right) + \alpha_2^2 \Delta x_+^T \Delta s_+. \end{aligned}$$

Therefore, since $0 \leq \alpha_1 \leq 1$ and $0 \leq \alpha_2 \leq 1$,

$$\begin{aligned} &\frac{\partial (x(\alpha)^T s(\alpha))}{\partial \alpha_1} = e^T (\tau_1 \mu e - xs)^- + 2\alpha_1 \Delta x_-^T \Delta s_- + \alpha_2 (\Delta x_-^T \Delta s_+ + \Delta x_+^T \Delta s_-) \\ &\leq e^T (\tau_1 \mu e - xs)^- + 2\Delta x_-^T \Delta s_- + \left|(D^{-1}\Delta x_-)^T (D\Delta s_+) + (D\Delta s_-)^T (D^{-1}\Delta x_+)\right| \\ &\leq e^T (\tau_1 \mu e - xs)^- + 2\Delta x_-^T \Delta s_- + \|(D^{-1}\Delta x_-, D\Delta s_-)\| \|(D^{-1}\Delta x_+, D\Delta s_+)\|, \end{aligned} \tag{17}$$

where $D = X^{1/2} S^{-1/2}$, and we also used the monotonicity of M (thus $\Delta x_-^T \Delta s_- \geq 0$) in the second step.

By Lemmas 3.5 and 3.6, we have

$$\begin{aligned}
\Delta x_-^T \Delta s_- &\le \|(\Delta x_- \Delta s_-)^+\|_1 \\
&= \|\left((D^{-1}\Delta x_-)(D\Delta s_-)\right)^+\|_1 \\
&\le \|(D^{-1}\Delta x_-)(D\Delta s_-)\|_1 \\
&\le \frac{1}{4}\left(\|D^{-1}\Delta x_-\|^2 + \|D\Delta s_-\|^2\right) \\
&\le \frac{1}{4}\|D^{-1}\Delta x_- + D\Delta s_-\|^2 \\
&= \frac{1}{4}\|(\tau_1\mu e - xs)^- / \sqrt{xs}\|^2 \\
&= \frac{1}{4}\|\sqrt{(xs-\tau_1\mu e)^+}\sqrt{(xs-\tau_1\mu e)^+/xs}\|^2 \\
&\le \frac{1}{4}\|\sqrt{(xs-\tau_1\mu e)^+}\|^2 \\
&= e^T(xs-\tau_1\mu e)^+/4,
\end{aligned}\tag{18}$$

where we used the monotonicity $(D^{-1}\Delta x_-)^T D\Delta s_- = \Delta x_-^T \Delta s_- \ge 0$ in the fifth step, and the fact that $0 \le (xs-\tau_1\mu e)^+/xs \le e$ in the eighth step. In fact, one concludes from the chain of inequalities in (18) that

$$\|(D^{-1}\Delta x_-, D\Delta s_-)\|^2 \le e^T(xs-\tau_1\mu e)^+. \tag{19}$$

Similarly,

$$\begin{aligned}
\|(D^{-1}\Delta x_+, D\Delta s_+)\|^2 &\le \|D^{-1}\Delta x_+ + D\Delta s_+\|^2 \\
&= \|(\tau_1\mu e - xs)^+/\sqrt{xs}\|^2 \\
&\le \|\sqrt{(\tau_1\mu e - xs)^+}\|^2 = e^T(\tau_1\mu e - xs)^+,
\end{aligned}$$

where the third step is due to the fact that since $\tau_1 \le 2\tau_2$ and $(x,s) \in \mathcal{N}_\infty^-(1-\tau_2)$, and so

$$(\tau_1\mu e - xs)^+ \le (2\tau_2\mu e - xs)^+ = (2(\tau_2\mu e - xs) + xs)^+ \le xs.$$

Furthermore,

$$\begin{aligned}
e^T(\tau_1\mu e - xs)^+ &\le n(\tau_1-\tau_2)\mu \\
&\le n\mu/8 \\
&\le (1-\tau_1)n\mu/6 \\
&= e^T(xs-\tau_1\mu e)/6 \\
&\le e^T(xs-\tau_1\mu e)^+/6.
\end{aligned}$$

Therefore we have

$$\|(D^{-1}\Delta x_+, D\Delta s_+)\|^2 \le e^T(xs-\tau_1\mu e)^+/6. \tag{20}$$

Substituting (18), (19), and (20) into (17) finally yields that

$$\frac{\partial(x(\alpha)^T s(\alpha))}{\partial\alpha_1} \le -\left(\frac{1}{2} - \frac{1}{\sqrt{6}}\right) e^T(xs-\tau_1\mu e)^+ < 0. \qquad \square$$

In view of Theorems 4.1 and 4.2, we may solve subproblem (6) approximately in the following way. First, set $\alpha_2 = 1/\hat{\eta}^2$. Second, find the greatest $\bar{\alpha}_1$ in $[0,1]$ such that $(x(\bar{\alpha}_1, 1/\hat{\eta}^2), s(\bar{\alpha}_1, 1/\hat{\eta}^2)) \in \mathcal{N}(\tau_1, \tau_2, \eta)$. One may, for instance, use bisection on α_1 for this purpose. Then, set $\alpha_1 = \bar{\alpha}_1$. Theorem 4.2 and Lemma 3.8 guarantee that $\bar{\alpha}_1 \geq \hat{\eta}^{-1}\sqrt{\beta\tau_1/n}$. It is clear that if the plane search procedure as described in Theorem 4.1 is replaced by this line search procedure, then the $O(\sqrt{n}\log\frac{(x^0)^T s^0}{\varepsilon})$ iteration bound still holds. Particularly, if we let $\eta^k \equiv 1$, then $\hat{\eta} \equiv 1$ and so we can always choose $\alpha_2 \equiv 1$. Another benefit of $\eta^k \equiv 1$ is that the corresponding neighborhoods $\mathcal{N}(\tau_1^0, \tau_2^0, 1)$ are simply expressed by (21) and (3). A concrete practical implementation is recommended as follows. Its numerical performance will be discussed in section 6.

ALGORITHM 4.3.

Input parameters: required precision $\varepsilon > 0$, target parameter and neighborhood parameters $0 < \tau = \tau_1 < 1$ and $\tau_2 = 0.5\tau_1$ (i.e., $\beta = 0.5$), and the initial solution $(x^0, s^0) \in \mathcal{N}(\tau_1^0, \tau_2^0, 1)$.

Step 0. *Set $k = 0$.*

Step 1. *If $(x^k)^T s^k \leq \varepsilon$, then stop.*

Step 2. *Solve $(\Delta x_-^k, \Delta s_-^k)$ and $(\Delta x_+^k, \Delta s_+^k)$ based on (4) and (5).*
Set $\alpha_2^k = 1$ and find the largest α_1^k on the closed interval $[\sqrt{\beta\tau_1/n}\,,\ 1]$, such that $(x(\alpha^k), s(\alpha^k)) \in \mathcal{N}(\tau_1, \tau_2, 1)$.

Step 3. *Set $(x^{k+1}, s^{k+1}) := (x(\alpha^k), s(\alpha^k))$.*
Let $k := k+1$ and go to **Step 1**.

5. A predictor-corrector scheme. In this section, we shall slightly change the notation. For simplicity, we shall always choose $\eta = 1$, i.e., we consider the neighborhood $\mathcal{N}(\tau_1, \tau_2, 1)$. We introduce a new notation $\mathcal{N}(\tau_1; \beta)$ to indicate the set $\mathcal{N}(\tau_1, \tau_2, 1)$ (see (3)), i.e.,

$$\mathcal{N}(\tau_1; \beta) = \left\{(x, s) \in \mathcal{F}_{++} \;\middle|\; \|(\tau_1\mu e - xs)^+\| \leq \beta\tau_1\mu\right\}, \tag{21}$$

where $\beta = (\tau_1 - \tau_2)/\tau_1$, as given in (14).

Below we describe another variant of Algorithm 2.1, which is essentially a predictor-corrector type algorithm.

ALGORITHM 5.1.

Input parameters: required precision $\varepsilon > 0$, neighborhood parameters $0 < \tau_1 \leq 1/4$, $0 < \beta \leq 1/2$, and the initial solution $(x^0, s^0) \in \mathcal{N}(\tau_1; \beta/2)$.

Output: a sequence of iterates $\{(x^k, s^k) \mid k = 0, 1, 2, \dots\}$.

Step 0. *Set $k = 0$.*

Step 1. *If $(x^k)^T s^k \leq \varepsilon$, then stop. Otherwise, if k is even (including 0), go to* **Step 2**; *if k is odd, go to* **Step 3**.

Step 2. *(predictor step). Set $\tau^k = 0$. Solve $(\Delta x_-^k, \Delta s_-^k)$ based on (4).*
Find largest step size $0 < \alpha_1^k \leq 1$, such that $(x(\alpha_1^k), s(\alpha_1^k)) \in \mathcal{N}(\tau_1; \beta)$. Go to **Step 4**.

Step 3. *(corrector step). Set $\tau^k = \tau_1$. Solve $(\Delta x_-^k, \Delta s_-^k)$ and $(\Delta x_+^k, \Delta s_+^k)$ based on (4) and (5).*
Find step size vector $\alpha^k = (\alpha_1^k, 1) \in [0,1]^2$, such that $(x(\alpha^k), s(\alpha^k)) \in \mathcal{N}(\tau_1; \beta/2)$ and α_1^k is maximum. Go to **Step 4**.

Step 4. *Set $(x^{k+1}, s^{k+1}) := (x(\alpha^k), s(\alpha^k))$.*
Let $k := k+1$ and go to **Step 1**.

Remark that in the predictor step, since τ^k is set to be 0, we have $(\tau^k\mu_k e - x^k s^k)^+ = 0$, and so the Newton constituent direction with respect to the positive part is simply zero. In both the corrector and the predictor steps, we only need to search for a single step size. An important feature of the above algorithm is that in the corrector step we also aim at a large update of the target. In other words, the gap function is expected to be reduced for the corrector steps as well.

We shall now prove that Algorithm 5.1 indeed works correctly.

Let us denote

$$\lambda := \|(\tau_1 \mu e - xs)^+\|_1 / \|(\tau_1 \mu e - xs)^-\|_1, \tag{22}$$

which means

$$\lambda e^T (\tau_1 \mu e - xs)^- + e^T (\tau_1 \mu e - xs)^+ = 0.$$

So when we choose $\alpha_1 \geq \lambda$ we have

$$\mu(\alpha_1, 1) \leq \mu + \mathit{\Delta} x(\alpha_1, 1)^T \mathit{\Delta} s(\alpha_1, 1)/n. \tag{23}$$

If $(x, s) \in \mathcal{N}(\tau_1; \beta)$, $\tau_1 \leq 1/4$, and $\beta \leq 1/2$, then we derive from (13) that

$$\begin{aligned} \lambda &\leq \sqrt{n} \|(\tau_1 \mu e - xs)^+\| / \left((1 - \tau_1) x^T s\right) \\ &\leq \frac{\sqrt{\beta \tau_1}}{1 - \tau_1} \sqrt{\frac{\beta \tau_1}{n}} \\ &\leq \frac{\sqrt{2}}{3} \sqrt{\frac{\beta \tau_1}{n}}. \end{aligned} \tag{24}$$

If $(x, s) \in \mathcal{N}(\tau_1; \beta)$, then computing $\hat{\eta}$ from (15) would yield

$$\hat{\eta} = 1, \tag{25}$$

and so by (10) we obtain immediately that

$$h(\alpha_1, 1) \geq \tau_1 \mu e \tag{26}$$

for all $\alpha_1 \in [0, 1]$ and for all $(x, s) \in \mathcal{F}_{++}$.

LEMMA 5.2. *Suppose $(x, s) \in \mathcal{N}(\tau_1; \beta)$, $\tau = \tau_1 \leq 1/4$ and $\beta \leq 1/2$. Then, for any $\alpha_1 \in [\lambda, \sqrt{\frac{\beta \tau_1}{2n}}]$ we have $(x(\alpha_1, 1), s(\alpha_1, 1)) \in \mathcal{N}(\tau_1; \beta/2)$.*

Proof. First of all, observe that (24) guarantees that the interval $[\lambda, \sqrt{\frac{\beta \tau_1}{2n}}]$ is not empty. Due to (23), (26), (25), and Lemma 3.6, we have

$$\begin{aligned}
&\left\|(\tau_1\mu(\alpha_1,1)e - x(\alpha_1,1)s(\alpha_1,1))^+\right\| \\
&= \left\|(\tau_1\mu(\alpha_1,1)e - h(\alpha_1,1) - \Delta x(\alpha_1,1)\Delta s(\alpha_1,1))^+\right\| \\
&\le \left\|\left(\tau_1\mu e - h(\alpha_1,1) + \frac{\tau_1\Delta x(\alpha_1,1)^T\Delta s(\alpha_1,1)}{n}e - \Delta x(\alpha_1,1)\Delta s(\alpha_1,1)\right)^+\right\| \\
&\le \left\|\left(\frac{\tau_1\Delta x(\alpha_1,1)^T\Delta s(\alpha_1,1)}{n}e - \Delta x(\alpha_1,1)\Delta s(\alpha_1,1)\right)^+\right\| \\
&\le \left\|\frac{\tau_1\Delta x(\alpha_1,1)^T\Delta s(\alpha_1,1)}{n}e + (-\Delta x(\alpha_1,1)\Delta s(\alpha_1,1))^+\right\| \\
&\le \left\|\frac{\tau_1\Delta x(\alpha_1,1)^T\Delta s(\alpha_1,1)}{n}e\right\| + \left\|(-\Delta x(\alpha_1,1)\Delta s(\alpha_1,1))^+\right\| \\
&\le \Delta x(\alpha_1,1)^T\Delta s(\alpha_1,1) + \left\|(\Delta x(\alpha_1,1)\Delta s(\alpha_1,1))^-\right\| \\
&= \left\|(\Delta x(\alpha_1,1)\Delta s(\alpha_1,1))^+\right\|_1 - \left\|(\Delta x(\alpha_1,1)\Delta s(\alpha_1,1))^-\right\|_1 + \left\|(\Delta x(\alpha_1,1)\Delta s(\alpha_1,1))^-\right\| \\
&\le \left\|(\Delta x(\alpha_1,1)\Delta s(\alpha_1,1))^+\right\|_1 \\
&\le (\beta/2)\tau_1(3\mu/4).
\end{aligned}$$

Applying Lemma 3.3 yields

$$\mu(\alpha_1,1) \ge (1-\sqrt{\beta\tau_1/2n})\mu \ge (1-1/4)\mu = 3\mu/4.$$

Therefore, $(x(\alpha_1,1),s(\alpha_1,1)) \in \mathcal{N}(\tau_1;\beta/2)$. □

LEMMA 5.3. *Suppose that* $(x,s)\in\mathcal{N}(\tau_1;\beta)$, $\tau=\tau_1\le 1/4$, *and* $\beta\le 1/2$. *Then we have*

$$\mu\left(\sqrt{\frac{\beta\tau_1}{2n}},1\right) \le \left(1-\frac{\sqrt{2\beta\tau_1}}{32\sqrt{n}}\right)\mu.$$

Proof. Let us denote $\alpha^0 := (\sqrt{\frac{\beta\tau_1}{2n}},1)$. Notice that $\hat\eta=1$. Due to (13) and Lemma 3.6 we obtain

$$\begin{aligned}
\mu\left(\sqrt{\frac{\beta\tau_1}{2n}},1\right) &= \mu + \sqrt{\frac{\beta\tau_1}{2n}}\frac{e^T(\tau_1\mu e - xs)^-}{n} + \frac{e^T(\tau_1\mu e - xs)^+}{n} + \frac{\Delta x(\alpha^0)^T\Delta s(\alpha^0)}{n} \\
&\le \mu - \sqrt{\frac{\beta\tau_1}{2n}}\,(1-\tau_1)\mu + \|(\tau_1\mu e - xs)^+\|/\sqrt{n} + \|(\Delta x(\alpha^0)\Delta s(\alpha^0))^+\|_1/n \\
&\le \mu - \frac{3\sqrt{2}}{8}\sqrt{\frac{\beta\tau_1}{n}}\,\mu + \frac{\beta\tau_1\mu}{\sqrt{n}} + \frac{3\beta\tau_1\mu}{8n} \\
&\le \mu - \frac{3\sqrt{2}}{8}\sqrt{\frac{\beta\tau_1}{n}}\,\mu + \sqrt{\frac{\beta\tau_1}{8n}}\,\mu + \frac{3}{8\sqrt{8}}\sqrt{\frac{\beta\tau_1}{n}}\mu \\
&= \mu - \frac{\sqrt{2}}{32}\sqrt{\frac{\beta\tau_1}{n}}\,\mu.
\end{aligned}$$

The lemma is proven. □

A remarkable feature of Algorithm 5.1, as revealed by Lemma 5.3, is that the gap measurement μ is reduced by a rate of $1 - 1/O(\sqrt{n})$ even at the corrector steps.

LEMMA 5.4. *Let $(\Delta x^a, \Delta s^a)$ be the search direction of a predictor step in Algorithm 5.1, and $\bar{\alpha}$ be the actual step size taken in that predictor step. Then,*

$$\bar{\alpha} \geq \frac{2}{1 + \sqrt{1 + 4\delta/\beta\tau_1}},$$

where $\delta = \|\Delta x^a \Delta s^a\| / \mu$.

Proof. We have

$$\mu(\alpha) = x(\alpha)^T s(\alpha)/n = (1-\alpha)\mu + \alpha^2 (\Delta x^a)^T \Delta s^a / n.$$

Note that $(x, s) \in \mathcal{N}_2^-(\tau_1, \beta)$ and

$$\begin{aligned} \left\| \left(\tau_1((\Delta x^a)^T \Delta s^a/n)e - \Delta x^a \Delta s^a\right)^+ \right\|^2 &\leq \left\|\tau_1((\Delta x^a)^T \Delta s^a/n)e - \Delta x^a \Delta s^a\right\|^2 \\ &= \|\Delta x^a \Delta s^a\|^2 - \tau_1(2-\tau_1)((\Delta x^a)^T \Delta s^a)^2/n \\ &\leq \|\Delta x^a \Delta s^a\|^2 . \end{aligned}$$

Therefore,

$$\begin{aligned} &\left\| (\tau_1 \mu(\alpha) e - x(\alpha) s(\alpha))^+ \right\| \\ &= \left\| \left((1-\alpha)(\tau_1 \mu e - xs) + \alpha^2 \tau_1((\Delta x^a)^T \Delta s^a/n)e - \alpha^2 \Delta x^a \Delta s^a\right)^+ \right\| \\ &\leq (1-\alpha) \left\|(\tau_1 \mu e - xs)^+\right\| + \alpha^2 \left\| \left(\tau_1((\Delta x^a)^T \Delta s^a/n)e - \Delta x^a \Delta s^a\right)^+ \right\| \\ &\leq (1-\alpha)\beta\tau_1\mu + \alpha^2 \|\Delta x^a \Delta s^a\| . \end{aligned}$$

Applying similar reasoning as in Lemma 4.17 of [16], we see that for each α with

$$0 \leq \alpha \leq \frac{2}{1 + \sqrt{1 + 4\delta/\beta\tau_1}}$$

we will have

$$\begin{aligned} \left\| (\tau_1 \mu(\alpha) e - x(\alpha) s(\alpha))^+ \right\| &\leq (1-\alpha)\beta\tau_1\mu + \alpha^2 \|\Delta x^a \Delta s^a\| \\ &\leq 2\beta\tau_1(1-\alpha)\mu \\ &\leq 2\beta\tau_1\mu(\alpha), \end{aligned}$$

and therefore, $(x(\alpha), s(\alpha)) \in \mathcal{N}(\tau_1; 2\beta)$. Differently put, we have $\bar{\alpha} \geq \frac{2}{1+\sqrt{1+4\delta/\beta\tau_1}}$ as the lemma claims. □

Since $\delta \leq n/2$, together with Lemma 5.3, the next theorem follows immediately (see also the proof of Theorem 4.18 in [16]).

THEOREM 5.5. *Let $\beta = 1/4$. Then Algorithm 5.1 will terminate in $O(\sqrt{n} \log \frac{(x^0)^T s^0}{\varepsilon})$ iterations.*

Assume additionally that the LCP problem has a strictly complementary solution; that is, there is a partition B and N, and solution (x^*, s^*), such that $B \cup N =$

$\{1, 2, \ldots, n\}$, $B \cap N = \emptyset$, $x^*_B > 0$, $x^*_N = 0$, and $s^*_B = 0$, $s^*_N > 0$. Since the sequence generated by Algorithm 5.1 is contained in a wide neighborhood, it satisfies

$$(1 - \beta)\tau_1 \mu_k \leq x^k s^k \leq n\mu^k. \tag{27}$$

Using Lemma 2 of [3], we know that there exists some constant $0 < \xi < 1$ such that

$$\xi \leq x^k_j \leq 1/\xi \quad \text{for} \quad j \in B, \quad \text{and} \quad \xi \leq s^k_j \leq 1/\xi \quad \text{for} \quad j \in N. \tag{28}$$

For simplicity, we drop the index k. We apply the same proof as for Theorem 3.6 in [17], and so relations (27) and (28) give rise to

$$\|\varDelta x^a\| = O(\mu) \quad \text{and} \quad \|\varDelta x^a\| = O(\mu).$$

Then due to Lemma 5.4, we have the following result.

THEOREM 5.6. *Let $\{(x^k, s^k) \mid k = 0, 1, 2, \ldots\}$ be the sequence generated by Algorithm* 5.1*. Suppose that* (*LCP*) *has a strictly complementarity solution. Then, $(x^k)^T s^k \to 0$ Q-quadratically for the predictor steps.*

6. Preliminary numerical tests. We shall test our algorithms on some randomly generated instances, in order to get a feel of how the method might perform in practice.

To achieve this, we wrote simple MATLAB codes for four algorithms: (1) the Mizuno–Todd–Ye type predictor-corrector algorithm [8]; (2) the classical wide-neighborhood path-following algorithm of Kojima–Mizuno–Yoshise [5]; (3) Algorithm 4.3; and (4) Algorithm 5.1. These four algorithms will be denoted, respectively, by (1) Algorithm PC; (2) Algorithm WN; (3) Algorithm New-WN; and (4) Algorithm New-PC. All algorithms do not use Mehrotra's higher order correction technique. The neighborhoods are taken to be $\mathcal{N}_2(1/2)$ in predictor step and $\mathcal{N}_2(1/4)$ in corrector step for Algorithm PC, and $\mathcal{N}^-_\infty(1 - \tau/2)$ for Algorithm WN, and $\beta = 1/2$ for Algorithm New-WN and Algorithm New-PC. To test the role of the parameter τ, we tried three different values of τ: $\tau = \tau^1 = 0.005$, $\tau = \tau^2 = 0.001$, and $\tau = \tau^3 = 0.0005$, respectively. All algorithms terminate after the relative duality gap satisfies

$$\frac{x^T s}{(x^0)^T s^0 + 1} \leq 10^{-8}.$$

For each dimension n, the entry in the column "iter" is the average number of iterations of 10 randomly generated monotone LCPs with the same n, and the number in the bracket is the standard deviation of these 10 runs. In case a MATLAB numerical warning occurred in the procedure, then we mark a superscript * next to that corresponding entry.

The first set of testing monotone LCP problems are generated as follows. After one inputs any positive integer n, MATLAB generates an $n \times n$ matrix $A = \text{rand}(n)$ randomly. Then we take $M = A^T A$ and $b = e - Me$ to obtain a monotone LCP and its initial feasible solution (e, e). The numerical results of this set of problems are shown in Table 1.

To test the influence from the skewness of matrix M, in the next set of test problems we let $M = A^T A + m(B - B^T)$, where $B = \text{rand}(n)$ and $m = 1$. The numerical results are shown in Table 2. It turns out that the number of iterations actually decreases on average as compared with the case when M is purely positive semidefinite. In our experiences, we found that the numbers of iterations for all algorithms tested always decrease if m increases.

Table 1
Iteration numbers on monotone LCPs with $M = A^T A$.

	PC	WN			New-WN			New-PC		
		$\tau = \tau^1$	$\tau = \tau^2$	$\tau = \tau^3$	$\tau = \tau^1$	$\tau = \tau^2$	$\tau = \tau^3$	$\tau = \tau^1$	$\tau = \tau^2$	$\tau = \tau^3$
n	iter	iter	iter	iter	iter	iter	iter	iter	iter	iter
100	34.0 (1.70)	16.8 (0.70)	16.7 (0.87)	16.7 (1.33)	10.7 (0.43)	10.6 (0.45)	10.6 (0.45)	12.5 (0.47)	12.2 (0.49)	12.1 (0.50)
200	36.8 (0.61)	16.1 (0.57)	16.9 (0.37)	16.0 (0.66)	10.6 (0.20)	10.7 (0.22)	10.7 (0.22)	12.9 (0.23)	12.9 (0.23)	12.8 (0.22)
300	38.6 (0.61)	17.1 (0.30)	17.9 (0.49)	19.4 (0.64)	11.0 (0.12)	11.1 (0.10)	11.2 (0.11)	13.2 (0.18)	13.3 (0.16)	13.3 (0.16)
400	41.6 (0.65)	20.0 (0.42)	19.0 (0.45)	21.3 (0.57)	11.7 (0.20)	11.7 (0.19)	11.7 (0.19)	13.6 (0.23)	13.7 (0.21)	13.7 (0.21)
500	40.6* (0.50)	19.4 (0.54)	20.8 (0.40)	19.8 (0.63)	11.2 (0.12)	11.4 (0.13)	11.4 (0.13)	13.5 (0.17)	13.4 (0.18)	13.4 (0.18)
600	43.2 (0.45)	20.6 (0.34)	21.2 (0.44)	23.8 (0.45)	11.3 (0.10)	11.5 (0.12)	11.5 (0.12)	14.1 (0.16)	14.2 (0.17)	14.2 (0.17)
700	43.6 (0.30)	22.5 (0.49)	22.8 (0.54)	22.1 (0.40)	12.0 (0.10)	12.0 (0.10)	12.1 (0.10)	14.8 (0.13)	14.9 (0.15)	14.8 (0.16)
800	44.0* (0.34)	22.3 (0.41)	23.3 (0.47)	23.0 (0.26)	12.3 (0.07)	12.4 (0.09)	12.4 (0.09)	15.0 (0.14)	15.0 (0.14)	14.9 (0.15)
900	43.4* (0.32)	22.3 (0.29)	23.1 (0.34)	21.3 (0.41)	12.0 (0.12)	12.0 (0.12)	12.0 (0.12)	14.4 (0.18)	14.4 (0.18)	14.5 (0.17)
1000	44.4 (0.28)	25.0 (0.41)	23.3 (0.49)	23.8 (0.60)	12.2 (0.10)	12.1 (0.10)	12.1 (0.10)	15.3 (0.13)	15.4 (0.13)	15.4 (0.13)

Table 2
Iteration numbers on monotone LCPs with $M = A^T A + m(B - B^T)$.

	PC	WN			New-WN			New-PC		
		$\tau = \tau^1$	$\tau = \tau^2$	$\tau = \tau^3$	$\tau = \tau^1$	$\tau = \tau^2$	$\tau = \tau^3$	$\tau = \tau^1$	$\tau = \tau^2$	$\tau = \tau^3$
n	iter	iter	iter	iter	iter	iter	iter	iter	iter	iter
100	11.0 (0.00)	6.2 (0.46)	5.7 (0.28)	5.1 (0.22)	4.8 (0.13)	4.0 (0.00)	4.0 (0.00)	4.0 (0.00)	4.0 (0.00)	4.0 (0.00)
200	9.0 (0.00)	5.0 (0.17)	4.9 (0.21)	5.1 (0.19)	4.0 (0.00)	4.0 (0.00)	4.0 (0.00)	4.0 (0.00)	4.0 (0.00)	4.0 (0.00)
300	9.0 (0.00)	4.5 (0.12)	5.3 (0.18)	4.9 (0.19)	4.0 (0.00)	4.0 (0.00)	4.0 (0.00)	4.0 (0.00)	4.0 (0.00)	4.0 (0.00)
400	9.0 (0.00)	4.6 (0.10)	5.4 (0.16)	5.3 (0.16)	4.0 (0.00)	4.0 (0.00)	3.8 (0.00)	4.0 (0.00)	4.0 (0.00)	4.0 (0.00)
500	9.0 (0.00)	4.7 (0.09)	4.7 (0.11)	4.4 (0.09)	4.0 (0.00)	4.0 (0.00)	3.2 (0.00)	4.0 (0.00)	4.0 (0.00)	4.0 (0.00)
600	9.0 (0.00)	4.5 (0.06)	4.6 (0.12)	4.5 (0.09)	4.0 (0.00)	3.4 (0.06)	3.1 (0.04)	4.0 (0.00)	4.0 (0.00)	3.8 (0.05)
700	9.0 (0.00)	4.7 (0.08)	3.8 (0.07)	4.3 (0.09)	4.0 (0.00)	3.0 (0.00)	3.0 (0.00)	4.0 (0.00)	4.0 (0.00)	3.3 (0.05)
800	9.0 (0.00)	4.5 (0.06)	3.2 (0.07)	4.4 (0.11)	4.0 (0.00)	3.0 (0.00)	3.0 (0.00)	4.0 (0.00)	3.7 (0.05)	3.1 (0.03)
900	9.0 (0.00)	4.4 (0.05)	3.7 (0.07)	4.0 (0.16)	4.0 (0.00)	3.0 (0.00)	3.0 (0.00)	4.0 (0.00)	3.5 (0.05)	3.0 (0.00)
1000	9.0 (0.00)	4.0 (0.00)	3.4 (0.05)	3.9 (0.08)	4.0 (0.00)	3.0 (0.00)	3.0 (0.00)	4.0 (0.00)	3.5 (0.05)	3.0 (0.00)

Based on the numerical results we have generated so far, Algorithm New-WN is the fastest, and Algorithm New-WN and Algorithm New-PC are faster than Algorithm WN, while Algorithm PC appears to be the slowest. Moreover, Algorithm WN, Algorithm New-WN, and Algorithm New-PC had always run smoothly, but Algorithm PC got 3 warnings due to badly scaled matrices. Certainly, our implementations are very coarse. For instance, we did not fine tune the parameters, nor did we use any higher order corrections. In the future, we plan to study the performance of the method for practical problems with more refined linear algebras and careful implementations.

REFERENCES

[1] W. Ai, *Neighborhood-following algorithms for linear programming*, Sci. China Ser. A, 47 (2004), pp. 812–820.

[2] C. C. Gonzaga, *The largest step path following algorithm for monotone linear complementarity problems*, Math. Programming, 76 (1997), pp. 309–332.

[3] O. Güler and Y. Ye, *Convergence behavior of interior-point algorithms*, Math. Programming, 60 (1993), pp. 215–228.

[4] P. Hung and Y. Ye, *An asymptotical $O(\sqrt{n}L)$-iteration path-following linear programming algorithm that use wide neighborhoods*, SIAM J. Optim., 6 (1996), pp. 570–586.

[5] M. Kojima, S. Mizuno, and A. Yoshise, *A primal-dual interior point algorithm for linear programming*, in Progress in Mathematical Programming: Interior Point and Related Methods, N. Megiddo, ed., Springer-Verlag, New York, 1989, pp. 29–47.

[6] M. Kojima, S. Mizuno, and A. Yoshise, *A polynomial-time algorithm for a class of linear complementarity problems*, Math. Programming, 44 (1989), pp. 1–26.

[7] N. Megiddo, *Pathways to the optimal set in linear programming*, in Progress in Mathematical Programming: Interior Point and Related Methods, N. Megiddo, ed., Springer-Verlag, New York, 1989, pp. 131–158.

[8] S. Mizuno, M. J. Todd, and Y. Ye, *On adaptive step primal-dual interior-point algorithms for linear programming*, Math. Oper. Res., 18 (1993), pp. 964–981.

[9] R. D. C. Monteiro and I. Adler, *Interior path following primal-dual algorithms,* I. *Linear programming*, Math. Programming, 44 (1989), pp. 27–41.

[10] R. D. C. Monteiro and I. Adler, *Interior path following primal-dual algorithms,* II. *Convex quadratic programming*, Math. Programming, 44 (1989), pp. 43–66.

[11] J. M. Peng, T. Terlaky, and Y.B. Zhao, *A Predictor-Corrector Algorithm for Linear Optimization Based on a Specific Self-Regular Proximity Function*, Technical report, McMaster University, Ontario, Canada, 2003.

[12] J. F. Sturm, *Using SeDuMi* 1.02*, a MATLAB toolbox for optimization over symmetric cones*, Optim. Methods Softw., 11/12 (1999), pp. 625–653. Special issue on Interior Point Methods (CD supplement with software).

[13] J. F. Sturm and S. Zhang, *On a wide region of centers and primal-dual interior point algorithms for linear programming*, Math. Oper. Res., 22 (1997), pp. 408–431.

[14] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadephia, 1997.

[15] X. Xu, *An $O(\sqrt{n}L)$-Iteration Large-Step Infeasible Path-Following Algorithm for Linear Programming*, Technical report, Department of Management Sciences, University of Iowa, Iowa City, IA, 1994.

[16] Y. Ye, *Interior Point Algorithms: Theory and Analysis*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, New York, 1997.

[17] Y. Ye and K. Anstreicher, *On quadratic and $O(\sqrt{n}L)$ convergence of a predictor-corrector algorithm for LCP*, Math. Programming, 62 (1993), pp. 537–551.

[18] Y. Ye, O. Güler, R. A. Tapia, and Y. Zhang, *A quadartically convergent $O(\sqrt{n}L)$-iteration algorithm for linear programming*, Math. Programming, 59 (1993), pp. 151–162.

SIAM J. OPTIM.
Vol. 16, No. 2, pp. 418–433

VALIDATED LINEAR RELAXATIONS AND PREPROCESSING: SOME EXPERIMENTS[*]

R. BAKER KEARFOTT[†] AND SIRIPORN HONGTHONG[†]

Abstract. Based on work originating in the early 1970s, a number of recent global optimization algorithms have relied on replacing an original nonconvex nonlinear program by convex or linear relaxations. Such linear relaxations can be generated automatically through an automatic differentiation process. This process decomposes the objective and constraints (if any) into convex and nonconvex unary and binary operations. The convex operations can be approximated arbitrarily well by appending additional constraints, while the domain must somehow be subdivided (in an overall branch-and-bound process or in some other local process) to handle nonconvex constraints. In general, a problem can be hard if even a single nonconvex term appears. However, certain nonconvex terms lead to easier-to-solve problems than others. Recently, Neumaier, Lebbah, Michel, ourselves, and others have paved the way to utilizing such techniques in a validated context.

In this paper, we present a symbolic preprocessing step that provides a measure of the intrinsic difficulty of a problem. Based on this step, one of two methods can be chosen to relax nonconvex terms. This preprocessing step is similar to a method previously proposed by Epperly and Pistikopoulos [*J. Global Optim.,* 11 (1997), pp. 287–311] for determining subspaces in which to branch, but we present it from a different point of view that is amenable to simplification of the problem presented to the linear programming solver, and within a validated context. Besides an illustrative example, we have implemented general relaxations in a validated context, as well as the preprocessing technique, and we present experiments on a standard test set. Finally, we present conclusions.

Key words. nonconvex optimization, global optimization, computational complexity, automatic differentiation, GlobSol, symbolic computation, linear relaxation

AMS subject classifications. 90C30, 65K05, 90C26, 68Q25

DOI. 10.1137/030602186

1. Introduction.

1.1. The general global optimization problem.

1.1. The general global optimization problem. Our general global optimization problem can be stated as

$$\boxed{\begin{array}{l} \text{minimize } \varphi(x) \\ \text{subject to } c_i(x) = 0,\ i = 1, \ldots, m_1, \\ \qquad\qquad g_i(x) \le 0,\ i = 1, \ldots, m_2, \\ \text{where } \varphi : \boldsymbol{x} \to \mathbb{R} \text{ and } c_i, g_i : \boldsymbol{x} \to \mathbb{R}, \text{ and where } \boldsymbol{x} \subset \mathbb{R}^n \text{ is} \\ \text{the hyperrectangle (box) defined by} \\ \qquad\qquad \underline{x}_i \le x_i \le \overline{x}_i,\ 1 \le i \le n, \\ \text{where the } \underline{x}_i \text{ and } \overline{x}_i \text{ are constant bounds.} \end{array}} \tag{1.1}$$

We will call this problem a *general nonlinear programming problem*, abbreviated "general NLP" or "NLP."

1.2. Deterministic branch-and-bound methods. In deterministic branch-and-bound methods for finding global minima, an initial region $\boldsymbol{x}^{(0)}$ is adaptively subdivided into subregions $\boldsymbol{x}$ of the form in (1.1), while an upper bound $\overline{\varphi}$ to the

[*] Received by the editors December 2, 2003; accepted for publication (in revised form) March 7, 2005; published electronically October 17, 2005.

http://www.siam.org/journals/siopt/16-2/60218.html

[†] Department of Mathematics, University of Louisiana, U. L. Box 4-1010, Lafayette, LA 70504-1010 (rbk@louisiana.edu, sxh1113@hotmail.com).

global optimum of φ is maintained (say, by evaluating φ at a succession of feasible points). A lower bound $\underline{\varphi}(\boldsymbol{x})$ on the optimum of φ over the subregion $\boldsymbol{x}$ is then computed. If $\underline{\varphi}(\boldsymbol{x}) > \overline{\varphi}$, then $\boldsymbol{x}$ is rejected; otherwise, other techniques are used to reduce, eliminate, or subdivide $\boldsymbol{x}$. The original explanation for this technique appears in [4, 22], while a relatively early didactic explanation appears in [19]. For more recent explanations in which convex underestimators are employed, see, for example, [5], [23]. For explanations focusing on validation but restricted to traditional interval arithmetic-based techniques, see [8] or [10, Chapter 5].

The effectiveness of the above technique depends on the quality of the upper bound $\overline{\varphi}$ and the lower bound $\underline{\varphi}(\boldsymbol{x})$. The upper bound $\overline{\varphi}$ may be obtained by various techniques, such as by locating a feasible point (or local optimum) $\check{x}$, then evaluating φ at $\check{x}$. A naive way of obtaining $\underline{\varphi}(\boldsymbol{x})$ is to simply evaluate φ with interval arithmetic over $\boldsymbol{x}$, and use the lower bound of the value $\boldsymbol{\varphi}(\boldsymbol{x})$. However, $\boldsymbol{\varphi}(\boldsymbol{x})$ so obtained takes no account of the constraints, and (since the feasible portion of $\boldsymbol{x}$, although possibly nonempty, may be much smaller than $\boldsymbol{x}$ itself) the lower bound $\underline{\varphi}(\boldsymbol{x})$ may not be sharp enough to be of use. More effective techniques appear to be those that solve coupled systems that take account of both objective and constraints. Convex and linear underestimators are used in a common variant of such techniques.

1.3. Convex underestimators and overestimators. Convex underestimators and overestimators are a primary tool to replace problem (1.1) by a simpler problem, the global optimum of which is less than or equal to the global optimum of (1.1). For example, if φ is replaced by a quadratic or piecewise linear function $\varphi^{(\ell)}$ such that $\varphi^{(\ell)}(x) \leq \varphi(x)$ for $x \in \boldsymbol{x}$, then the resulting problem has global optimum that underestimates the global optimum of (1.1). Similarly, if $m_1 = 0$ (i.e., if there are no equality constraints) and, in addition to replacing φ by $\varphi^{(q)}$, each g_i is replaced by a linear function $g_i^{(\ell)}$ such that $g_i^{(\ell)}(x) \leq g_i(x)$ for $x \in \boldsymbol{x}$, then the resulting quadratic or linear program, termed a *relaxation* of (1.1), has optimum that is less than or equal to the optimum of (1.1). (If there are equality constraints, then each equality constraint can be replaced, at least in principle, by two linear inequality constraints.)

1.3.1. An arithmetic on underestimators and overestimators. Constraints or objective functions that represent simple binary operations (addition, subtraction, multiplication, and division) or unary operations (standard functions such as $y = e^x$ or $y = x^n$) can be bounded below or above on a particular interval by linear relations. For instance, if $\underline{g}_1$ is a linear underestimator for g_1 and $\underline{g}_2$ is a linear underestimator for g_2, then a linear underestimator for $g_1 + g_2$ is $\underline{g}_1 + \underline{g}_2$. Thus, addition of two linear underestimators can be defined simply by addition of the corresponding linear coefficients. Similarly, if $\underline{g}_1$ is a linear underestimator for g_1 and $\underline{-g}_2$ is a linear underestimator for $-g_2$, then $\underline{g}_1 + \underline{-g}_2$ is a linear underestimator for $g_1 - g_2$. Linear underestimators for multiplication are somewhat more involved but can similarly be obtained operationally. For convex functions such as e^g, for $g \in [a, b]$, a linear underestimator is the tangent line at any point $c \in [a, b]$, while for concave functions g, the best possible linear underestimator is the secant line connecting $(a, g(a))$ $(b, g(b))$. If $[a, b]$ is too wide to get sharp underestimators and overestimators, then $[a, b]$ may be subdivided, and linear underestimators can be supplied for each subinterval.

In actually generating a linear program whose solution underestimates the solution of (1.1), we replace an expression g by a new intermediate variable v for the underestimator wherever the expression g occurs; we then append the constraint $v \geq \underline{g}$ for the linear underestimator to the set of constraints. In case multiple linear con-

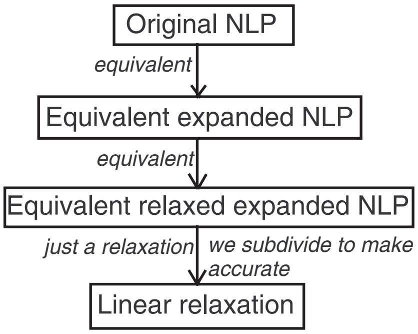


FIG. 1.1. *Our four stages in analyzing a linear relaxation of an NLP.*

straints are used for more accuracy, we introduce multiple variables v_i and w_i and corresponding multiple constraints.

An arithmetic can be used to automatically compute underestimators given by expressions or computer programs. The original idea for such an arithmetic predates the groundbreaking work of McCormick [15], [16]. Such an arithmetic may employ operator overloading or similar technology, such as explained, say, in [20], [10, section 1.4], or in the proceedings [1], [2], or [7]. A framework for such automatic computation is given in [23, section 4.1]. In such an arithmetic, given underestimators for expressions g_1 and g_2, formulas are implemented for computing underestimators of $g_1 + g_2$, $g_1 * g_2$, and g_1/g_2, as well as for computing underestimators of powers, exponentials, logarithms, and other such functions encountered in practice.

Many of the ideas for such an arithmetic appear in the work of McCormick [15], [16], and [17]. Significant portions of the books [5] and [23] are devoted to techniques for deriving underestimators and overestimators as we have just described, and for implementing automatic computation of these. For example, [23, Chapter 3] contains techniques for computing underestimators of sums of products, and [23, Chapter 4] summarizes rules for automatic computation of underestimators, based on convex envelopes and linear underestimations. The techniques in [23] are embodied in the highly successful software package BARON.

Lastly, Gatzke, Tolsma, and Barton [6] have implemented automated generation of both linear underestimating techniques as in [23] and convex underestimating techniques as in [5] in their DAEPACK system.

1.4. Our view of the process. In this work, to aid our analysis of the difficulty of particular problems, we view the process slightly differently. In particular, we first generate a list of operations (known as a *code list* or *tape* among experts in automatic differentiation), and we assign an *equality* constraint to each operation, leading to an *equivalent expanded NLP*. We may then analyze each such equality constraint in the equivalent expanded NLP to determine if we may replace the equality constraint by a "$\le$" constraint or a "$\ge$" constraint, to obtain an *equivalent relaxed expanded NLP*.

In a third step, we replace the nonlinear operations in the constraints in the equivalent relaxed expanded NLP by linear underestimators or linear overestimators. For nonlinear operations and equality constraints, both underestimators and overestimators are required, while only underestimators are required for "$\le$" constraints, and only overestimators are required for "$\ge$" constraints. We call the resulting linear program a *linear underestimating relaxation.* This four stage scheme is diagrammed in Figure 1.1.

For underestimators of convex operations or overestimators of concave operations, additional constraints can be appended in the linear underestimating relaxation to sharpen the approximation. However, in overestimations of convex operations or underestimations of concave operations, the linear underestimating relaxation cannot be made to more sharply underestimate the original problem by appending additional linear constraints; in these cases, in general, the domain must be subdivided, and a linear underestimating relaxation must be solved over each subproblem. In our procedure, we will explicitly identify those operations requiring solution of linear underestimating relaxations over subregions to obtain increased accuracy. We will also identify which (and how many) variables to subdivide to achieve the increased accuracy. The number of such variables gives the dimension of the subspace in which tessellation must occur, and thus gives a measure of how much effort needs to be expended to accurately approximate a solution.

Epperly and Pistikopoulos [3] proposed a method for subspace analysis that gives subspaces similar to ours; they also illustrated its effectiveness on various problems. However, their view of the process is different from the above, and they did not implement the process in a validated context.

1.5. Organization of this work. In section 2, we give a simple example that is used throughout the rest of the paper to illustrate the concepts. In section 3, we define and illustrate our concept of expanded NLP and equivalent relaxed expanded NLP, and we give a theorem that shows how we may replace equality constraints by inequality constraints in the expanded NLP to obtain an equivalent relaxed expanded NLP. In section 4 we give details on refining convex and concave constraints, while in section 5, we describe our algorithm structure for an automatic analysis. The results of an automatic analysis appears in section 6. We give conclusions and a brief outline of ongoing work in section 8.

2. An illustrative example. *Example* 1. *Minimize*

$$\varphi(x) = (x_1 + x_2 - 1)^2 - \left(x_1^2 + x_2^2 - 1\right)^2$$

for $x_1 \in [-1, 1]$ *and* $x_2 \in [-1, 1]$.

TABLE 2.1
A code list, interval enclosures, and expanded NLP for Example 1.

♯	Operation	Enclosures	Constraints	Convexity
1	$v_3 \leftarrow x_1 + x_2$	$[-2, 2]$	$x_1 + x_2 - v_3 = 0$	linear
2	$v_4 \leftarrow v_3 - 1$	$[-3, 1]$	$v_3 - 1 - v_4 = 0$	linear
3	$v_5 \leftarrow v_4^2$	$[0, 9]$	$v_4^2 - v_5 \leq 0$	convex
4	$v_6 \leftarrow x_1^2$	$[0, 1]$	$v_1^2 - v_6 = 0$	both
5	$v_7 \leftarrow x_2^2$	$[0, 1]$	$v_2^2 - v_7 = 0$	both
6	$v_8 \leftarrow v_6 + v_7$	$[0, 2]$	$v_6 + v_7 - v_8 = 0$	linear
7	$v_9 \leftarrow v_8 - 1$	$[-1, 1]$	$v_8 - 1 - v_9 = 0$	linear
8	$v_{10} \leftarrow -v_9^2$	$[-1, 0]$	$-v_9^2 - v_{10} \leq 0$	nonconvex
9	$v_{11} \leftarrow v_5 + v_{10}$	$[-1, 9]$	$v_5 + v_{10} - v_{11} \leq 0$	linear

Example 1, a small unconstrained problem except for bound constraints, is easily solved by GlobSol [11], a traditional interval branch-and-bound method. However, it is nonconvex, and can be used to illustrate underlying concepts in this work. To generate a linear relaxation of this problem, we first assign intermediate variables to intermediate operations, thus generating a code list. (This can be done within a compiler or by operator overloading.) Such a code list is seen in the second column of Table 2.1.

The third column of Table 2.1 contains enclosures for the corresponding intermediate variables, based on $x_1 \in [-1, 1]$ and $x_2 \in [-1, 1]$. (Here, we obtained these enclosures with traditional interval evaluations of the corresponding operations.) For example, the enclosure $[-1, 9]$ in the last row represents bounds on the objective for $x_1 \in [-1, 1]$ and $x_2 \in [-1, 1]$.

We explain the fourth and fifth columns of Table 2.1 below.

3. The expanded NLP and the equivalent relaxed expanded NLP. If we replace each operation in the code list by an equality constraint, we obtain an equivalent NLP, in the sense that the optimum and optimizing values of the independent variables for the resulting NLP are the same as the optimum and optimizing values of the original NLP.

DEFINITION 3.1. *Given the original NLP* (1.1)*, the* expanded NLP *is that NLP obtained by replacing the objective and constraints by corresponding intermediate variables for the individual operations and assigning equality constraints to the intermediate variables.*

In Example 1, an expanded NLP can be defined from the operations in Table 2.1, to obtain

$$\begin{aligned}
&\text{minimize } v_{11}\\
&\text{subject to } v_1 + v_2 - v_3 = 0,\\
&\qquad v_3 - 1 - v_4 = 0\\
&\qquad v_4^2 - v_5 = 0\\
&\qquad v_1^2 - v_6 = 0\\
&\qquad v_2^2 - v_7 = 0\\
&\qquad v_6 + v_7 - v_8 = 0\\
&\qquad v_8 - 1 - v_9 = 0\\
&\qquad -v_9^2 - v_{10} = 0\\
&\qquad v_5 + v_{10} - v_{11} = 0\\
&\qquad v_1 \in [-1, 1],\ v_2 \in [-1, 1].
\end{aligned} \tag{3.1}$$

As an intermediate step in producing a linear relaxation of the original NLP, we replace as many of the equality constraints as possible in the expanded NLP by inequality constraints subject to the resulting problem being equivalent to the original one. We do this according to the following definition and theorem.

DEFINITION 3.2. *Suppose we have an expanded NLP as in Definition* 3.1*, and we replace as many of the equality constraints as possible in the expanded NLP by inequality constraints, according to the following rules.*

1. *Unless the objective consists of an independent variable only, the top-level operation $\varphi = v_k = f(v_q, v_r)$ or $\varphi = f(v_q)$ (corresponding to the bottom of the code list and evaluation of the objective) may be replaced by an inequality constraint of the form $f \le v_k$. (In Table* 2.1*, φ corresponds to v_{11}, and $f(v_q, v_r) = v_5 + v_{10}$.)*
2. *In constrained problems, operations corresponding to $c_i(x) = 0$ or $g_i(x) \le 0$ are placed unaltered into the constraint set. For example, if g_i were defined by intermediate variable v_k in the code list, then the constraint $v_k \le 0$ would be placed into the set of constraints.*
3. *(Recursive conditions) If a binary operation $v_i = f_i(v_\ell, v_m)$ or a unary operation $v_i = f(v_\ell)$ computes a value v_i that enters only as an argument to*

operations $v_j = f_j(v_i, \cdot)$ or $v_j = f_j(v_i)$, such that for every j, f_j is monotonic in v_i, then

(a) *if, for those j for which f_j is monotonically increasing with respect to v_i, operation f_j corresponds to an inequality constraint of the form $f_j \le v_j$, and, for those j for which f_j is monotonically decreasing with respect to v_i, operation f_j corresponds to an inequality constraint of the form $f_j \ge v_j$, then v_i may correspond to an inequality constraint of the form $f_i \le v_i$;*

(b) *if, for those j, for which f_j is monotonically increasing with respect to v_i, operation f_j corresponds to an inequality constraint of the form $f_j \ge v_j$, and, for those j for which f_j is monotonically decreasing with respect to v_i, operation f_j corresponds to an inequality constraint of the form $f_j \le v_j$, then v_i may correspond to an inequality constraint of the form $f_i \ge v_i$.*

4. *All other operations correspond to equality constraints.*

Then the resulting NLP is called an equivalent relaxed expanded NLP *for the original NLP* (1.1).

The fourth column of Table 2.1 shows the constraints corresponding to the equivalent relaxed expanded NLP corresponding to the code list in the second column of Table 2.1.

Theorem 3.3. *The equivalent relaxed expanded NLP of Definition* 3.2 *is equivalent to the expanded NLP of Definition* 3.1 *in the sense that*

1. *the optimum of the equivalent relaxed expanded NLP is the same as the optimum of the expanded NLP;*
2. *the set of optimizing points of the equivalent relaxed expanded NLP contains the set of optimizing points of the expanded NLP;*
3. *under a "strict monotonicity" condition described in the proof of this theorem, the sets of optimizing points of the equivalent relaxed expanded NLP and of the expanded NLP are the same.*

Thus, since the expanded NLP is equivalent to the original NLP, the equivalent relaxed expanded NLP is equivalent to the original NLP in the same sense.

Before we prove Theorem 3.3, we use Example 1 to illustrate the process defined in Definition 3.2. Although a computer can easily label each operation as corresponding to equality or inequality by a backwards traversal of the code list, we illustrate the process with a computational graph. The computational graph corresponding to the code list in Table 2.1 appears in Figure 3.1. To label each node in the graph, we traverse the graph from the bottom up. The bottom node is labelled as "$\le$." We then check the nodes immediately above nodes already checked to see if they satisfy the recursive condition 3 of Definition 3.2. Any node that fails to satisfy the recursive condition is labelled an equality constraint, and all nodes above that node in the computational graph are labelled equality constraints. Figure 3.1 illustrates the result of this process.

We have actually implemented this automatic labeling process in a Fortran 95 module, and have used it in the experiments below. The Fortran 95 module is available from the authors upon request.

Using Definition 3.2 (and comparing with Figure 3.1 and to the fourth column of

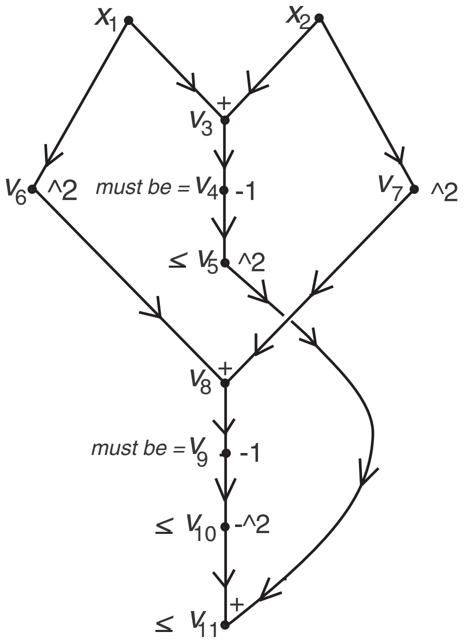


Fig. 3.1. *The computational graph corresponding to the code list in Table* 2.1.

Table 2.1), the equivalent relaxed expanded NLP for Example 1 is

$$
\begin{aligned}
&\text{minimize } v_{11}\\
&\text{subject to } v_1 + v_2 - v_3 = 0,\\
&\qquad v_3 - 1 - v_4 = 0\\
&\qquad v_4^2 - v_5 \le 0\\
&\qquad v_1^2 - v_6 = 0\\
&\qquad v_2^2 - v_7 = 0\\
&\qquad v_6 + v_7 - v_8 = 0\\
&\qquad v_8 - 1 - v_9 = 0\\
&\qquad -v_9^2 - v_{10} \le 0\\
&\qquad v_5 + v_{10} - v_{11} \le 0\\
&\qquad v_1 \in [-1, 1],\ v_2 \in [-1, 1].
\end{aligned}
\tag{3.2}
$$

Similarly, the proof of Theorem 3.3 proceeds by induction on the nodes of the computational graph, starting at the bottom.

Proof of Theorem 3.3. Assume first that the only change made to the expanded NLP is replacement of the equality constraint $v_{\text{final}} = f$ by an inequality constraint $f \le v_{\text{final}}$ according to rule 1, and suppose that the resulting NLP is not equivalent to the original expanded NLP. Then, since the resulting NLP is a relaxation of the original NLP, the resulting NLP must have an optimizer that is not in the feasible set of the original NLP. However, the only way this can be is if the inequality constraint $f \le v_{\text{final}}$ is strict. However, we may then reduce v_{final} and remain in the feasible set, contradicting the assumption that v_{final} represented an optimum.

Now suppose that we start with a problem $\mathcal{P}$ that is equivalent to the expanded NLP from application of some of the rules in Definition 3.2, and suppose we obtain a new problem $\mathcal{P}_{\text{new}}$ from $\mathcal{P}$ by applying rule 3(a) to $\mathcal{P}$, that is, by replacing $f_i = v_i$

by $f_i \le v_i$. Then, arguing as above, any optimizer of $\mathcal{P}_{\text{new}}$ that is not an optimizer of $\mathcal{P}$ would need to correspond to the strict inequality $f_i < v_i$. But then, we could decrease v_i until $v_i = f_i$, and each constraint in which v_i occurred would remain feasible. Thus, the optimum of $\mathcal{P}_{\text{new}}$ would have to correspond to the optimum of $\mathcal{P}$, and the optimizing points of $\mathcal{P}$ are optimizing points of $\mathcal{P}_{\text{new}}$.

For the stronger assertion about the optimizing sets, suppose that each f_j related to any f_i that occurs in rule 3(a) or rule 3(b) either strictly increasing or strictly decreasing, and that each intermediate computation is part of either the objective or a constraint. (By this last condition, we mean that there are no "dead ends" in the computation, i.e., there are no bottom nodes in the computational graph that correspond neither to an objective nor a constraint.) Then, by decreasing v_i, each f_j either decreases or increases, and we can decrease or increase the corresponding v_j's without affecting the feasibility of the problem. We can, in turn, decrease or increase variables depending on those v_j's that we have so adjusted, until we adjust a variable v_k upon which no other variables depend. Due to the "no dead ends" assumption, this variable v_k represents, without loss of generality, either the objective value φ or a constraint value or g. (It cannot represent an equality constraint $c = 0$, since then the constraint we have relaxed could not have been replaced by an inequality in the first place.) If this variable v_k represents the objective: $v_k \le \varphi$, then φ can be decreased; this would, however, contradict the assumption that we started with an optimizer of $\mathcal{P}_{\text{new}}$. On the other hand, if v_k represented a constraint $g \le 0$, then our adjustments will have decreased g, which means the adjusted point is further inside the interior of the feasible region of g; in turn, this means that the point must have been feasible for $\mathcal{P}$, contradicting our assumption.

A similar argument holds if we start with a problem $\mathcal{P}$ that is equivalent to the expanded NLP from application of some of the rules in Definition 3.2, and we obtain a new problem $\mathcal{P}_{\text{new}}$ from $\mathcal{P}$ by applying rule 3(b) to $\mathcal{P}$. This proves the theorem. □

NLPs whose code list generates many equality constraints corresponding to nonlinear operations are more difficult to solve, in a sense to be made explicit below. This is because, to relax a nonlinear equality constraint, we obtain both a convex operation and a concave (nonconvex) operation, and a more expensive kind of branching appears necessary for nonconvex operations.

The actual solution to the original NLP of Example 1 (as bounded by GlobSol) is $x_1, x_2 \in [0.269593, 0.269595]$, $\varphi(x) \in [-0.51805866866, -0.51805866865]$. When the approximate solver IPOPT [24] is given the equivalent relaxed expanded NLP (3.2), IPOPT happens to return values within these bounds.

4. Relaxations. We may replace each convex and nonconvex constraint in an expanded NLP by a linear relaxation. In validated computations, we also generally replace each linear equality constraint by a pair of linear inequality constraints that tightly contain the linear constraint but take account of roundoff error in computing the coefficients. In both validated and nonvalidated computations, we replace each nonlinear equality constraint by a pair of inequality constraints; in this case, if the original nonlinear equality constraint was convex, we obtain both a convex and a concave constraint.

Table 4.1 illustrates a possible set of relaxations for the expanded NLP of Table 2.1.

In the fourth column of Table 4.1, the underestimates for the convex terms v_4^2, x_1^2, and x_2^2 correspond to the tangent lines to the operations at the midpoint of the

Table 4.1
A linear relaxation corresponding to the expanded NLP in Table 1.

♯	Operation	Enclosures	Under/over estimators	Convexity
1	$v_3 \leftarrow x_1 + x_2$	$[-2, 2]$	$x_1 + x_2 - v_3 = 0$	linear
2	$v_4 \leftarrow v_3 - 1$	$[-3, 1]$	$v_3 - 1 - v_4 = 0$	linear
3	$v_5 \leftarrow v_4^2$	$[0, 9]$	$(4.5)^2 + 9(v_4 - 4.5) - v_5 \leq 0$	convex
4	$v_6 \leftarrow x_1^2$	$[0, 1]$	$(0.5)^2 + 1(v_1 - 0.5) - v_6 \leq 0$ $v_1 - v_6 \geq 0$	convex nonconvex
5	$v_7 \leftarrow x_2^2$	$[0, 1]$	$(0.5)^2 + 1(v_2 - 0.5) - v_7 \leq 0$ $v_2 - v_7 \geq 0$	convex nonconvex
6	$v_8 \leftarrow v_6 + v_7$	$[0, 2]$	$v_6 + v_7 - v_8 = 0$	linear
7	$v_9 \leftarrow v_8 - 1$	$[-1, 1]$	$v_8 - 1 - v_9 = 0$	linear
8	$v_{10} \leftarrow -v_9^2$	$[-1, 0]$	$-1 - v_{10} \leq 0$	nonconvex
9	$v_{11} \leftarrow v_5 + v_{10}$	$[-1, 9]$	$v_5 + v_{10} - v_{11} \leq 0$	linear

enclosure interval; for example, the expression $(4.5)^2 + 9(v_4 - 4.5)$ in the third row corresponds to the tangent line to v_4^2 at $v_4 = 4.5$. The nonconvex operations ($-v_9^2$, $-v_1^2$, and $-v_2^2$) are underestimated by the secant line connecting the endpoints of the graph. If the expanded NLP is replaced by "minimize v_{11} subject to $x_1 \in [-1, 1]$, $x_2 \in [-1, 1]$, and subject to each of the constraints in column 4," then the solution to the resulting linear program, which we call an *expanded LP*, underestimates the solution to the original NLP. When we gave IPOPT the expanded LP corresponding to Table 4.1, IPOPT obtained $(x_1, x_2) \approx (0.3894, 0.3894)$, $\varphi = -1$, an underestimator that is no better than the traditional interval evaluation of the objective over the box.

4.1. Refining convex constraints.

As explained in [23, section 4.2] and elsewhere, the nonlinear convex operations can be approximated more closely in the linear relaxation by appending more constraints corresponding to additional tangent lines. For example, in the nonlinear convex operation $v_5 \leftarrow v_4^2$ in Example 1, in addition to the constraint $v_5 \geq (4.5)^2 + 9(v_4 - 4.5)$ (corresponding to the tangent line at $v_4 = 4.5$), we may add the constraint $v_5 \geq (2.25)^2 + 4.5(v_4 - 2.25)$ (corresponding to the tangent line at $v_4 = 2.25$) and the constraint $v_5 \geq (6.75)^2 + 13.5(v_4 - 6.75)$ (corresponding to the tangent line at $v_4 = 6.755$), and any other similar tangent line. By spacing the tangent lines sufficiently close together, the corresponding convex constraint can be approximated arbitrary closely.

If there were no nonconvex operations in the code list, then all convex operations could be approximated arbitrarily closely by spacing tangent lines sufficiently close together. This process involves subdivision in a single variable for each convex nonlinear constraint, so the number of constraints in a linear programming relaxation whose solution approximated the solution to the original nonlinear program to a given accuracy would seem to be, essentially, linear in the number of operations in the code list.

Note that, with this univariate approximation technique, it is not necessary to branch on the variables corresponding to convex functions. That is, by using a number of linear outer approximations for each convex variable, accuracy is achieved without branching on those variables.

4.2. Refining nonconvex constraints.

However, the relaxations for nonconvex operations cannot be refined by appending additional constraints in the same way. Two possibilities come to mind:

- We may subdivide the original variables x_1 and x_2 to reduce the width of the enclosure for the domain of the operation corresponding to the nonconvex constraint, and thus reduce the slack in the linear underestimator for the nonconvex constraint. For example, if we bisected both x_1 and x_2, for Example 1, we would obtain four subdomains. We would obtain underestimates for the solution of the original problem over the subdomains as underestimates to the corresponding linear relaxations; an underestimate for the original problem over the original domain would consist of the minimum of the four underestimates over the subdomains.
- Alternately, we may subdivide the domain of the operation corresponding to the nonconvex constraint directly into two or more subintervals (or subboxes in the case of multiplication), and form relaxations corresponding to each of these subintervals or subboxes. For example, we could subdivide v_9 in Table 4.1 into $[-1, 0]$ and $[0, 1]$. We then underestimate $-v_9^2$ over each separate subinterval by its secant line. If we use this secant line, along with the original bounds $x_1 \in [-1, 1]$, $x_2 \in [-1, 1]$, we obtain a relaxation of the problem we would get by restricting x_1 and x_2 to those values leading to a range of v_9 in the selected subinterval (such as one of $[-1, 0]$ or $[0, 1]$). Thus, the minimum of the solutions to the relaxations so obtained will be an underestimate to the solution of the original nonlinear programming problem.

For example, consider, for the purpose of examining a single nonconvex operation, using the exact constraints in the expanded NLP of Table 2.1 except for that corresponding to operation 8, which we maintain as $-1-v_{10} \le 0$ as in Table 4.1. IPOPT gives an approximate solution $(x_1, x_2) \approx (0.5, 0.5)$, $\varphi \approx -1$ to this problem. We now subdivide v_9 into $v_9 \in [-1, 0]$ and $v_9 \in [0, 1]$. The relaxation of $v_{10} \ge -v_9^2$ over $[-1, 0]$ corresponding to the secant line through the endpoints is $v_{10} \ge v_9$. Replacing $v_{10} \ge -1$ (valid over $[-1, 1]$) by this and using a corresponding bound constraint on v_9, but otherwise keeping the same convex program, IPOPT gives an approximate solution $(x_1, x_2) \approx (0.3333, 0.3333)$, $\varphi \approx -0.6667$. Now replacing $v_{10} \ge -v_9^2$ over $[0, 1]$ by the relaxation corresponding to the secant line through the endpoints, namely $v_{10} \ge -v_9$, and using a corresponding bound constraint on v_9, IPOPT gives $(x_1, x_2) \approx (1, 0.6823)$, $\varphi \approx 9 \times 10^{-5}$. Thus, an underestimate for the solution to the original NLP, based on these linear relaxations, is approximately $\min\left\{-0.6667, 9 \times 10^{-5}\right\} = -0.6667$, a tighter estimate than that obtained by solving only a single problem, but obtained by subdividing in one variable only. A similar phenomenon would be seen if, instead of using exact inequalities for the convex operations, we used a sufficient number of linear underestimators.

5. Our preprocessing algorithms.

If there are only one or two nonconvex operations in the code list, but these nonconvex operations depend on many, if not all, of the dependent variables, then it is probably advantageous to use the second subdivision process (subdividing directly on the intermediate variables entering the nonconvex constraints). However, if there are many nonconvex constraints, all depending on the same small number of independent variables, then it is probably advantageous to do a traditional branch-and-bound within the subspace of independent variables corresponding to the nonconvex constraints. The problem will be “hard” if there are both a large number of nonconvex constraints and a large number of independent variables enter into these nonconvex constraints; otherwise, the problem is easily solvable by either branching and bounding directly on the intermediate variables entering the few convex constraints or by branching and bounding the few independent variables entering the nonconvex constraints.

With these considerations in mind, we have structured our preprocessing algorithms in the following order:

1. We first create the code list.
2. Evaluate the code list with interval arithmetic to obtain bounds on the intermediate variables[1].
3. Using Theorem 3.3 and the bounds from Step 2, start at the bottom of the computational graph to label each node in the computational graph as corresponding to an inequality or an equality constraint.
4. Using the ranges from Step 2, the labellings from Step 3, and considerations from section 4 (e.g., whether the function is convex or concave and whether an overestimate, underestimate, or both is required), label each node as requiring solution of problems on subdomains to obtain tighter approximations via linear programs or as requiring only the appending of additional constraints to a single problem over the original domain.
5. For those nodes in the computational graph requiring solution of problems over subdomains, trace up the computational graph to identify upon which independent variables the result depends.

Implementation of Steps 3 and 4 require a case-by-case consideration of the individual operations (exponential, odd, even, or real powers, etc.).

6. An example experiment. We have programmed each of the steps in section 5 within the GlobSol module structure, testing our programs with Example 1 and various other small problems with certain properties. For a reasonably simple but realistic test problem, we have tried the following problem, originally from [25].

Example 2. *Minimize*

$$\max_{1\le i\le m} \|f_i(x)\|\,, \text{ where}$$

$$\begin{aligned} f_i(x) &= x_1 e^{x_3 t_i} + x_2 e^{x_4 t_i} - \frac{1}{1+t_i},\\ t_i &= -0.5 + (i-1)/(m-1), \quad 1\le i\le m. \end{aligned}$$

We transformed this nonsmooth problem to a smooth problem with Lemaréchal's conditions [14] to obtain

$$\begin{aligned} &\text{minimize } v\\ &\text{subject to} \quad f_i(x) - v \le 0, 1\le i\le m\\ &\qquad\qquad -f_i(x) - v \le 0, 1\le i\le m. \end{aligned}$$

To test the preprocessing, we took $m = 21$, $x_i \in [-5, 5]$ for $1 \le i \le 4$, and $v \in [-100, 100]$. The resulting output had 221 blocks, each of the form

```
Row. no.,    OP,   CONSTRAINT_TYPE,  NEEDS_SUBPROBLEM

    1        A_X            EQUAL_V          F
    2        EXP            EQUAL_V          T
    3  X_TIMES_Y            EQUAL_V          T
    4        A_X            EQUAL_V          F
    5        EXP            EQUAL_V          T
```

[1] Constraint propagation may be used at this point.

```
    6  X_TIMES_Y           EQUAL_V         T
    7   X_PLUS_Y           EQUAL_V         F
    8   X_PLUS_B           EQUAL_V         F
    9  X_MINUS_Y      LESS_OR_EQ_V         F
   10    MINUS_X      LESS_OR_EQ_V         F
   11  X_MINUS_Y      LESS_OR_EQ_V         F

 Row. no., corresponding independent variables:
    2       3
    3       1     3
    5       4
    6       2     4
```

In rows 2 and 5 (and corresponding rows in the remaining 20 blocks), the dependence was only on variables 3 and 4. In rows 3 and 6 (and corresponding rows in the remaining 20 blocks), the binary operation is a multiplication. However (cf., e.g., [5, p. 45 ff.]), a multiplication can be both underestimated and overestimated arbitrarily closely by subdividing in only one of the two variables. Hence, this analysis reveals that we only need subdivide in variables 3 and 4 to obtain linear programs that approximate the original NLP arbitrarily closely. This can be interpreted to mean that, with a branch-and-bound algorithm based on linear underestimators and overestimators, the problem is inherently two-dimensional rather than four-dimensional.

In this case, the alternative would be to subdivide each of the intermediate variables corresponding to code list rows 2, 3, 5, 6, etc. Since this would result in subdivision in an 84-dimensional space, this alternative is clearly not appropriate for this problem.

6.1. Results within our branch-and-bound algorithm.

We implemented subdivision in the subspace, as we describe in section 7 below, within the search process that uses validated linear relaxations as we describe in [12].

7. Some systematic comparisons.

In [9], we detail some of the techniques we have used to provide machine-representable relaxations that are mathematically rigorous, while in [12] we describe our implementation of linear relaxations within GlobSol, and we give experimental results comparing use of linear relaxations within GlobSol to GlobSol without linear relaxations. However, in the experiments in [12], we worked in the full space and not in the subspace. In this section, we compare algorithm performance using branching only in the subspace to algorithm performance of the algorithm when we subdivide only along coordinate direction in the subspace.

We implemented the subspace analysis within GlobSol's overall search algorithm. In particular, we provided an option within GlobSol to apply the subspace analysis to each box processed in the branch-and-bound algorithm, and only coordinates corresponding to the identified subspaces were selected for further bisection[2].

We used essentially the same test set as in [12], namely, the "tiny" problems from Library 1 in the Neumaier test set [18]. The results appear in Table 7.1. We carried out the experiments in Table 7.1 on a dual 2.8 GHz processor[3] AMD Opteron machine running Linux (SuSe distribution 9.1), with 4 gigabytes of memory. We compiled the

[2] There is an advantage to doing the preprocessing to each box, since the labels on the computational graph depend on the ranges of the intermediate variables, and graphs may be more advantageously labelled over subdomains.

[3] The actual computations were not done in parallel, but the system load was such that, at all times, the GlobSol program had total resources of at least one processor.

Table 7.1
Results with and without subspace analysis.

Problem	n / n_r	m_1	m_2	Success?	# boxes (f/r)	CPU sec.(f/r)	Ratio	$n - n_r$
dispatch	4 / 3	1	1	Y / Y	13 / 13	0.5 / 0.49	1.0	1
ex14_1_1	3 / 2	0	4	N / Y	100000 / 1791	3569 / 102.4	0.0	1
ex14_1_2	6 / 3	1	8	N / N	43202 / 35488	3600 / 3600	1.0	3
ex14_1_3	3 / 2	0	4	Y / Y	568 / 564	3.87 / 3.9	1.0	1
ex14_1_5	6 / 4	4	2	Y / Y	101 / 100	2.99 / 2.86	1.0	2
ex14_1_9	2 / 1	0	2	Y / Y	57 / 102	0.49 / 0.81	1.7	1
ex14_2_1	5 / 4	1	6	N / N	33864 / 32949	3600 / 3600	1.0	1
ex14_2_2	4 / 3	1	4	Y / Y	1667 / 2220	106.7 / 116.6	1.1	1
ex14_2_3	6 / 5	1	8	N / N	18265 / 24816	3601 / 3601	1.0	1
ex14_2_5	4 / 3	1	4	Y / Y	1846 / 1890	97.96 / 100.1	1.0	1
ex2_1_1	5 / 5	0	1	Y / Y	234 / 234	0.96 / 0.96	1.0	0
ex2_1_2	6 / 5	0	2	Y / Y	173 / 173	0.56 / 0.56	1.0	1
ex2_1_4	6 / 1	0	4	Y / Y	222 / 222	2.18 / 2.21	1.0	5
ex3_1_1	8 / 5	0	6	N / N	61525 / 60871	3603 / 3603	1.0	3
ex3_1_2	5 / 4	0	6	Y / Y	78 / 78	0.45 / 0.43	1.0	1
ex3_1_3	6 / 6	0	6	Y / Y	253 / 253	0.5 / 0.5	1.0	0
ex4_1_2	1 / 1	0	0	Y / Y	6 / 6	0.28 / 0.27	1.0	0
ex4_1_4	1 / 1	0	0	Y / Y	7 / 7	0.01 / 0.01	1.0	0
ex4_1_5	2 / 1	0	0	Y / Y	39 / 39	0.09 / 0.09	1.0	1
ex4_1_6	1 / 1	0	0	Y / Y	5 / 5	0.01 / 0.01	1.0	0
ex4_1_7	1 / 1	0	0	Y / Y	4 / 4	0 / 0		0
ex4_1_8	2 / 1	1	0	Y / Y	5 / 5	0.01 / 0.01	1.0	1
ex4_1_9	2 / 1	0	2	Y / Y	38 / 38	0.13 / 0.13	1.0	1
ex5_4_2	8 / 5	0	6	Y / Y	550 / 511	23.16 / 19.32	0.8	3
ex6_1_1	8 / 8	6	0	N / N	19741 / 19170	3601 / 3601	1.0	0
ex6_1_2	4 / 4	3	0	Y / Y	122 / 122	2.65 / 2.66	1.0	0
ex6_1_4	6 / 3	4	0	Y / Y	623 / 600	37.82 / 38.39	1.0	3
ex7_2_1	7 / 7	0	14	N / N	7301 / 7267	3601 / 3601	1.0	0
ex7_2_2	6 / 2	4	1	Y / Y	101 / 101	4.47 / 4.68	1.0	4
ex7_2_5	5 / 5	0	6	Y / Y	153 / 153	3.1 / 3.23	1.0	0
ex7_2_6	3 / 3	0	1	Y / Y	36 / 36	0.13 / 0.14	1.1	0
ex7_3_3	5 / 1	2	6	N / Y	81771 / 55	3600 / 1.65	0.0	4
ex8_1_3	2 / 2	0	0	N / N	100000 / 100000	2414 / 2662	1.1	0
ex8_1_4	2 / 1	0	0	Y / Y	28 / 28	0.08 / 0.09	1.1	1
ex8_1_5	2 / 2	0	0	Y / Y	131 / 131	0.81 / 0.82	1.0	0
ex8_1_6	2 / 2	0	0	Y / Y	36 / 36	0.34 / 0.35	1.0	0
ex8_1_7	5 / 3	1	4	N / N	100000 / 100000	3169 / 3488	1.1	2
ex8_1_8	6 / 2	4	1	Y / Y	101 / 101	4.55 / 4.52	1.0	4
ex9_2_4	8 / 4	7	0	N / N	100000 / 100000	3237 / 3284	1.0	4
ex9_2_5	8 / 5	7	0	N / N	44233 / 43768	3606 / 3606	1.0	3
ex9_2_8	3 / 0	2	0	Y / Y	8 / 8	0 / 0		3
house	8 / 3	4	4	N / N	82533 / 31269	3602 / 3602	1.0	5
least	3 / 2	0	0	Y / Y	1440 / 1440	61.42 / 60.78	1.0	1
mhw4d	5 / 3	3	0	Y / Y	393 / 240	7.47 / 4.29	0.6	2
nemhaus	5 / 0	0	0	Y / Y	0 / 0	0 / 0		5
rbrock	2 / 1	0	0	Y / Y	4 / 4	0.01 / 0	0.0	1
sample	4 / 0	0	2	Y / Y	29 / 187	2.64 / 6.65	2.5	4
wall	6 / 0	6	0	Y / Y	117 / 117	2.51 / 2.54	1.0	6

experimental version of GlobSol with the NAG Fortran 95 compiler, release 5.0. The first column of Table 7.1 gives the problem name[4] from the Library 1 set in [18]. The second column of Table 7.1 gives the dimension n followed by the dimension n_r of the last reduced space computed. (The reduced space dimension depends on the box $\boldsymbol{x}$ and thus varies throughout the computation process.) Columns 3 and 4 give the number of equality constraints and inequality constraints, respectively.

As in [12], we used adaptive approximation of convex functions, with a relative tolerance $\epsilon_{\text{LP}} = 10^{-1}$.

We allowed GlobSol to consider no more than 100,000 subboxes, and we allowed

[4] These problems include problems of dimension 10 or less from the Library 1 set in [18], excluding those for which translation from AMPL format within the COCONUT [18] did not succeed in producing Fortran input that could be compiled. These problems are to be fixed in a future version of the COCONUT environment.

no more than 3,600 seconds of execution (CPU) time. Column 5, labelled "success?" gives, first, whether or not the search process without the subspace analysis succeeded within these bounds and, second, whether or not the search process with the subspace analysis succeeded within these allocated bounds. ("Y" signifies success, while "N" signifies failure.) Subsequent columns, giving performance comparisons between the search in the full space and the search in subspaces, are most meaningful when both the full space algorithm and subspace algorithm succeeded: Column 6, labelled "# boxes," gives the total number of boxes processed for the full-space algorithm, followed by the total number of boxes processed with the subspace algorithm; column 7, labelled "CPU sec.," gives first the total processor time for the full-space algorithm, then the total processor time for the subspace algorithm. The column labelled "ratio" gives the ratio of processor times: {time for the subspace algorithm} / {time for the full-space algorithm}. The last column, labelled $n - n_r$, gives the difference between the full-space dimension and the last computed subspace dimension (for easy comparison with the performance ratio).

7.1. Conclusions. A perusal of Table 7.1 shows that, for most of the problems in the test set, the subspace analysis has little effect on the practicality of the overall algorithm. However, `ex14_1_1` and `ex7_3_3`, in which the full-space algorithm fails to complete but the subspace algorithm completes extremely efficiently, are notable exceptions. Both of these problems are relatively simple, of a form similar to the Lemaréchal formulation in section 6.

In addition to the extreme contrast between the search in the full space and search in the subspace for `ex14_1_1` and `ex7_3_3`, use of the subspace analysis resulted in some improvement in `ex5_4_2` and `mhw4d`, and the subspace analysis resulted in more time spent in `ex14_1_9` and `sample`. In "sample," the reduced space had dimension 0, so no bisections were done: The additional boxes were an artifact of the box complementation process.

During a review of our implementation, we have found very recently that the subspace algorithm may be hampered by the way we are handling validation of the variable bounds in the orthogonal complement of the space of variables being bisected, and that considerably better performance of the subspace algorithm is possible. We have ideas of how to improve our process, but this will take significant additional development.

8. Summary and future work. In [9], we detail some of the techniques we have used to provide machine-representable relaxations that are mathematically rigorous, while in [12] we describe our implementation of linear relaxations within GlobSol, and we give experimental results comparing use of linear relaxations within GlobSol to GlobSol without linear relaxations. However, in the experiments in [12], we worked in the full space, and not in the subspace.

Here, we have presented an analysis of nonlinear programming problems that leads to a way of automatically determining a measure of difficulty for the problem. This analysis leads to a method of determining a lower-dimensional subspace in which to branch. This method appears to give similar subspaces to the method in [3], but we have implemented the method from a different point of view. We have presented the subspace analysis process on a particular problem we previously found to be difficult within a validation context but without linear relaxations.

We also mention that the algorithm in Epperly and Pistikopoulos [3] solves the problems presented in [3] more efficiently than our validated algorithm, but we suspect that this is not due to the subspace selection method. Our current thinking is that we will see equivalent efficiency with better validated handling of the orthogonal complement of the space of reduced variables, and we are currently developing this idea. Evidence that an improved validated environment can be produced is in the successful validated codes of Lebbah et al. [13].

We have used the subspace analysis technique in GlobSol's validated branch-and-bound algorithm, testing the technique on a published low-dimensional test set. Those tests revealed that, for most problems, there was little difference, but a huge advantage was revealed for two problems whose solution was impractical without the subspace analysis method.

The tests, along with those in [12], reveal that GlobSol's validated algorithm, although more practical when validated linear relaxations are included, still does not handle problems as quickly as the BARON package described in [23]. The subspace analysis method described here, not implemented in BARON, does greatly help for some problems, and additional tuning (i.e., the setting of heuristic parameters) may further improve performance. However, these are not the entire answers to questions concerning performance differences. Nonetheless, we are convinced that the performance differences are not an inevitable consequence of insistence on validation, but are a result of how techniques, which can be modified to be validated, are used and combined in the overall algorithm, and how efficiently these techniques are implemented.

For instance, which LP solver is used to solve the linear relaxations may have a significant effect on the practicality of the overall branch-and-bound algorithm. Also, the "probing" technique in BARON, first described in [21] and later in [23], may be effective; we are presently developing a validated version of this technique.

Acknowledgments. We wish to thank Sven Leyffer and Jorge Moré for inviting the first author to participate in the Global Optimization Theory Workshop at Argonne Laboratories, Fall 2003. The interactions during that conference greatly clarified his thinking on the subject, and led to this work. We also wish to thank Vladik Kreinovich for pointing out various aspects of computational complexity associated with global optimization to us. We also wish to thank the referees, especially for their pointers to previous work on the subject.

REFERENCES

[1] M. Berz, C. Bischof, G. Corliss, and A. Griewank, *Computational Differentiation: Techniques, Applications, and Tools*, SIAM, Philadelphia, 1996.

[2] G. Corliss, Ch. Faure, A. Griewank, L. Hascoët, and U. Naumann, *Automatic Differentiation of Algorithms: From Simulation to Optimization*, Springer-Verlag, New York, 2000.

[3] T. G. W. Epperly and E. N. Pistikopoulos, *A reduced space branch and bound algorithm for global optimization*, J. Global Optim., 11 (1997), pp. 287–311.

[4] J. E. Falk and R. M. Soland, *An algorithm for separable nonconvex programming problems*, Management Sci., 11 (1968), pp. 550–569.

[5] C. A. Floudas, *Deterministic Global Optimization: Theory, Methods, and Applications*, Kluwer, Dordrecht, The Netherlands, 2000.

[6] E. P. Gatzke, J. E. Tolsma, and P. I. Barton, *Construction of convex relaxations using automated code generation techniques*, Optim. Eng., 3 (2002), pp. 305–326.

[7] A. Griewank and G. F. Corliss, *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*, SIAM, Philadelphia, 1991.

[8] E. R. Hansen, *Global Optimization Using Interval Analysis*, Marcel Dekker, New York, 1992.

[9] S. Hongthong and R. B. Kearfott, *Rigorous Linear Overestimators and Underestimators*, 2004, preprint, http://interval.louisiana.edu/preprints/estimates_of_powers.pdf.

[10] R. B. Kearfott, *Rigorous Global Search: Continuous Problems*, Kluwer, Dordrecht, The Netherlands, 1996.

[11] R. B. Kearfott, *Globsol: History, composition, and advice on use*, in Global Optimization and Constraint Satisfaction, Lecture Notes in Comput. Sci. 2861, Springer-Verlag, New York, 2003, pp. 17–31.

[12] R. B. Kearfott, *Empirical comparisons of linear relaxations and alternate techniques in validated deterministic global optimization*, Optim. Methods Softw., 2004, preprint, http://interval.louisiana.edu/preprints/validated_global_optimization_search_comparisons.pdf.

[13] Y. Lebbah, C. Michel, M. Rueher, D. Daney, and J.-P. Merlet, *Efficient and safe global constraints for handling numerical constraint systems*, SIAM J. Numer. Anal., 42 (2005), pp. 2076–2097.

[14] C. Lemaréchal, *Nondifferentiable optimization*, in Nonlinear Optimization, 1981, M. J. D. Powell, ed., Academic Press, London, 1982, pp. 85–89.

[15] G. P. McCormick, *Converting General Nonlinear Programming Problems to Separable Nonlinear Programming Problems*, Technical report T-267, George Washington University, Washington, D.C., 1972.

[16] G. P. McCormick, *Computability of global solutions to factorable nonconvex programs*, I. *Convex underestimating problems*, Math. Programming, 10 (1976), pp. 147–175.

[17] G. P. McCormick, *Nonlinear Programming, Theory, Algorithms, and Applications*, John Wiley and Sons, New York, 1983.

[18] A. Neumaier, O. Shcherbina, W. Huyer, and T. Vink'o, *A comparison of complete global optimization solvers*, Math. Program., 103 (2005), pp. 335–356.

[19] P. M. Pardalos and J. B. Rosen, *Constrained Global Optimization: Algorithms and Applications*, Lecture Notes in Comput. Sci. 268, Springer-Verlag, Berlin, 1987.

[20] L. B. Rall, *Automatic Differentiation: Techniques and Applications*, Lecture Notes in Comput. Sci. 120, Springer-Verlag, Berlin, 1981.

[21] H. S. Ryoo and N. V. Sahinidis, *Global optimization of nonconvex NLPS and MINLPS with applications in process design*, Computers and Chemical Engineering, 19 (1995), pp. 551–566.

[22] R. M. Soland, *An algorithm for separable nonconvex programming problems,* II. *Nonconvex constraints*, Management Sci., 17 (1970), pp. 759–773.

[23] M. Tawarmalani and N. V. Sahinidis, *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming Theory, Algorithms, Software, and Applications*, Kluwer, Dordrecht, The Netherlands, 2002.

[24] A. Wächter, *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, 2002, http://dynopt.cheme.cmu.edu/andreasw/thesis.pdf.

[25] J. L. Zhou and A. L. Titts, *An SQP algorithm for finely discretized continuous minimax problems and other minimax problems with many objective functions*, SIAM J. Optim., 6 (1996), pp. 461–487.

SIAM J. Optim.
Vol. 16, No. 2, pp. 434–451

AN EFFICIENT APPROXIMATE ALGORITHM FOR THE 1-MEDIAN PROBLEM IN METRIC SPACES*

D. CANTONE†, G. CINCOTTI†, A. FERRO†, AND A. PULVIRENTI†

Abstract. We propose a simple and natural linear randomized algorithm for the approximate 1-median selection problem in metric spaces. The 1-median of a finite subset S of a metric space is the element of S which minimizes the average distance from the remaining points in S. This problem is extremely important in most applications using clustering of metric spaces, but also in connection with several algorithms in bioinformatics. The only linear approximation algorithm for the 1-median problem, which provably works in any metric space without going through any Euclidean space, has been proposed by Indyk in [*Proceedings of the* 31*st Annual ACM Symposium on Theory of Computing*, Atlanta, 1999, pp. 428–432]. However, Indyk's algorithm, which is based on sufficiently large sampling, turns out not to be a practical solution. The same holds true even for its heuristic variants which use samplings of smaller size. The algorithm we propose has a simple and efficient implementation, which performs better than Indyk's algorithm in practice. On the other hand, while the performance of Indyk's algorithm is guaranteed by an approximation factor, in the case of our algorithm we are only able to produce experimental evidence of its precision. Extensive experimentation has been performed on both synthetic and real input datasets. Synthetic datasets were generated with uniform and skewed distributions, using various metrics. Real datasets have been extrapolated from real world official databases available on the web. Successful results of the proposed algorithm are reported for several applications in bioinformatics and various classes of approximate search queries.

Key words. 1-median problem, clustroid selection, algorithms for metric spaces, selection algorithms, approximation algorithms, randomized algorithms, Fermat–Weber problem

AMS subject classifications. 54E35, 74P05, 68W05, 68W20, 68W25

DOI. 10.1137/S1052623403424740

1. Introduction.

Let (M, d) be a metric space with distance function $d : M \times M \longrightarrow \mathbb{R}$ and let S be a finite subset of M. Given $k \in \mathbb{N}$, the *k-median problem* for S is the problem of finding k points $c_1, c_2, \ldots, c_k$ in S which minimize the sum

$$\sum_{s \in S} \min_{i=1,\ldots,k} d(s, c_i).$$

The k-median problem is $\mathcal{NP}$-complete [20] and several approximation algorithms have been developed for it [10, 18, 23, 24, 25, 21].

The 1-*median problem*, obtained for $k = 1$, is therefore the problem of selecting an element c in S which minimizes the sum

$$w(c) = \sum_{s \in S} d(s, c)$$

of the distances from c to the remaining points of the input set S.

In Euclidean spaces the search procedure is not restricted to elements of the input set. In such spaces, the 1-median problem is easily seen to constitute an elementary

*Received by the editors March 25, 2003; accepted for publication (in revised form) February 23, 2005; published electronically November 14, 2005.

http://www.siam.org/journals/siopt/16-2/42474.html

†Department of Mathematics and Computer Science, University of Catania, Viale A. Doria 6, I–95125 Catania, Italy (cantone@dmi.unict.it, cincotti@dmi.unict.it, ferro@dmi.unict.it, apulvirenti@dmi.unict.it).

subcase of the *Fermat–Weber* problem, which has been studied intensively and shown to be unsolvable by algebraic constructions [3, 9]. An efficient numeric solution for the Fermat–Weber problem, based on the gradient descent method, has been presented in [29]. Other interesting numerical methods can be found in the literature [1, 26, 27, 30].

In general metric spaces there is an obvious quadratic solution of the *exact* 1-median problem; however, when the input size is large, even a quadratic algorithm can be prohibitive, and faster algorithms become necessary. A possibility would then be to take advantage of the fact that in most applications it is enough to find approximate solutions which are sufficiently close to the exact 1-median.

Recently, Indyk [21] has proposed a provably correct $(1 + \delta)$-approximation algorithm for the 1-median problem, having running-time $\mathcal{O}(n/\delta^5)$. The main part of Indyk's algorithm is a probabilistic comparator which, given any two points p and q in the input set S, performs an *approximate* comparison of the summations $w(p)$ and $w(q)$ in $\mathcal{O}(1/\delta^5)$-time. Such a procedure, based on "sufficiently large" sampling operations, returns with high probability depending on the approximation factor, the point with the smallest summation. Then one can easily construct a binary tournament tree over S, in which each internal node recursively selects the child with the smallest summation, using Indyk's probabilistic comparator. It can be shown that the final winner of the above tournament, which is the output of Indyk's algorithm, approximates the 1-median of S within a factor of $(1 + \delta)$.

However, due to the high cost of sampling operations inside the probabilistic comparator, Indyk's algorithm is not practical. In addition, it turns out that reducing the required samples size causes a significant loss in precision (see section 6.8).

In this paper we propose a simple and natural algorithm which efficiently computes an *approximate* solution of the 1-median problem in generic metric spaces. Our algorithm turns out to be very precise for most of the synthetic and real world problems investigated in this work, though its performance cannot be bound by any approximation factor (cf. section 4.1 of [4]).

Our proposed algorithm is based on a randomly generated tournament which, roughly speaking, is played as follows. For the sake of simplicity, let us assume that the size of the input set is an exact power of a fixed integer $t \geq 3$. At each round, the winners of the previous round are randomly partitioned into subsets of size t. The winners of the current round are then obtained by selecting the *exact* 1-median in each of the partitioning subsets. Rounds are played until only one element, namely the final winner, is left. The final winner is the output of our procedure.

Experimental results show that our algorithm is very effective. It is more efficient and precise as compared to Indyk's algorithm and the best elementary heuristic variants of Indyk's algorithm.

The paper is organized as follows. In section 2 we illustrate the main application fields of the 1-median selection problem. Our approximate solution is then fully described in section 3, its tuning is discussed in section 4, and its runtime complexity is analyzed in section 5. In section 6 we analyze an extensive collection of experimental results, to give an empirical evaluation of the precision and efficiency of our algorithm and to compare it to Indyk's algorithm and some of its variants. Finally, section 7 concludes the paper, pointing at some directions for future research.

2. Applications of the 1-median selection.

The 1-median problem for generic metric spaces is sometimes also referred to as the "center selection problem." Its main application fields, such as molecular biology [19], network management [2, 7, 15], and information retrieval [17], deal with both Euclidean and non-Euclidean metric spaces. Among its most relevant applications we cite the following:

- *approximate queries with a given threshold* on very large databases of objects belonging to clustered metric spaces. In such a problem, one seeks clusters whose representatives have distance from the query which is bounded by the threshold. It turns out that if the 1-medians are selected as representatives of the clusters and the clusters diameters are comparable with the threshold, then the average error during the search is minimized with very high probability [13, 14];
- *k-clustering of metric spaces*, in which iterative computations of 1-medians are required [16];
- *multiple alignment*, in which the goal is to find a common alignment of a set of genetic sequences [19] (this is a basic problem in biological data engineering).

In the above first two application problems, which come from the database community, 1-median computations are generally performed via embeddings in suitable Euclidean spaces. More specifically, the 1-median is approximated by the so-called "clustroid," which is the inverse image of the point closest to the center of gravity of the input set image (via the embedding).[1] However, if for some reason it is difficult to produce an immersion of a given metric space into a suitable Euclidean space, then an efficient algorithm for the direct computation of the 1-median in the given metric space must be designed.

Algorithmic solutions of optimization and search problems on metric spaces [18, 22] are very much affected by the high computational cost of distance calculations among objects of the space. Our solution was properly developed to overcome this difficulty. Its good behavior in practice has been successfully reported for several applications in bioinformatics [12] and various classes of approximate search queries [8].

3. The algorithm. In this section we present our randomized algorithm for the approximate 1-median selection. It is based on a tournament played among the elements of an input set S belonging to a given metric space (M, d). At each round, the elements which passed the preceding turn are randomly partitioned into subsets, say $X_1, \dots, X_k$. Then, the exact 1-median x_i of each subset X_i is computed, for $i = 1, \dots, k$. The elements $x_1, \dots, x_k$ win the round and go to the next one. The tournament terminates when only a single element $\overline{x}$, the final winner, remains. This is chosen to approximate the exact 1-median in S.

One possible implementation of the above general method is to partition the elements at each round in subsets of the same size t, with the possible exception of only one subset, whose size lies between $(t+1)$ and $(2t-1)$. Plainly, the requirement that no round is played with less than t elements is useful to ensure statistical significance to the tournament. In addition, we can assume that the iteration stops when the number of elements falls below a given *threshold*, at which point the exact 1-median of the residual elements is computed. This is summarized in Figure 1, where the local optimization procedure WINNER (X) returns the exact 1-median in X.

It is easily seen that in the case of unidimensional Euclidean spaces, our proposed algorithm is essentially the one presented in [4], for the approximate median computation of a finite ordered set. In particular, each random partitioning phase can be simplified by introducing efficient pseudorandomization methods as in [4].

4. Tuning of the algorithm. In this section we discuss the two fundamental parameters present in the algorithm reported in Figure 1, namely the *splitting factor*

[1] We recall that the center of gravity of a given subset X of an Euclidean space is the point which minimizes the sum of squares of distances from any other point in the set X [17, 31].

```
The approximate 1-median selection algorithm
WINNER ($X$)
    Input: A set $X$ of elements in a metric space $(M, d)$.
    Output: The exact 1-median element of $X$.
      Begin
          for each $x \in X$ do
              $s_x \leftarrow \sum_{y \in X} d(x, y)$;
          Let $c \in X$ be an element such that $s_c = \min_{x \in X}(s_x)$;
          return $c$;
      End;

APPROX_1_MEDIAN ($S$)
    Input: A set $S$ of elements in a metric space $(M, d)$.
    Output: The approximate 1-median element of $S$.
      Begin
          while $|S| >$ threshold do
              $W \leftarrow \emptyset$;
              while $|S| \geq 2t$ do
                  Choose randomly a subset $T \subseteq S$, with $|T| = t$;
                  $S \leftarrow S \setminus T$;
                  $W \leftarrow W \cup \{$WINNER ($T$)$\}$;
              end while;
              $S \leftarrow W \cup \{$WINNER ($S$)$\}$;
          end while;
          return WINNER ($S$);
      End.
```

FIG. 1. *Pseudocode of the approximate* 1*-median selection algorithm.*

t (also referred to as the *tournament size*) and the parameter *threshold*.

The splitting factor t is used to set the size of each subset X processed by calls to procedure WINNER within the internal **while**-loop of procedure APPROX_1_MEDIAN. It is clear that larger values of t correspond to more precise outputs, but higher computational costs. In several cases, a satisfying output quality can be obtained even with small values for t. In particular, when t is set to its minimum value, i.e., $t = 3$, the WINNER procedure can be implemented in an extremely efficient way, since one needs to compute just 3 distances, and the local output can be determined with an average of only $\frac{8}{3}$ distance comparisons [4]. A good trade-off between output quality and computational cost is obtained by choosing as value for t one unit more than the dimension that characterizes the investigated metric space [11]. This suggestion lies on intuitive grounds developed in the case of a Euclidean metric space $\mathbb{R}^m$ and it is confirmed by the experiments reported in section 6.

The parameter *threshold* controls the termination of the tournament. Again, larger values for *threshold* ensure more precise outputs, but at increasing computational cost. Observe that the value $(t^2 - 1)$ for *threshold* forces the property that the last set of elements, where the final winner is selected, must contain at least t elements, provided that $|S| \geq t$. Notice also that in order to ensure a linear computational complexity of the algorithm, the *threshold* value needs to be $\mathcal{O}(\sqrt{|S|})$ (see section 5). Thus, a good choice is *threshold* $= \min\{t^2 - 1, \sqrt{|S|}\}$.

5. Complexity analysis. The algorithm APPROX_1_MEDIAN given in Figure 1 is characterized by its simplicity and hence it is expected to be very efficient from the computational point of view, at least in the case in which the parameters t and *threshold* are small enough. In fact, as will be shown below, our algorithm has a worst-case complexity of $\frac{t}{2}n + o(n)$ in the input size n, provided that the *threshold* is $o(\sqrt{n})$.

Plainly, the complexity of the algorithm APPROX_1_MEDIAN is dominated by the number of distances computed within calls to procedure WINNER. To estimate such a number, let $W(n, t, \vartheta)$ denote the number of calls to procedure WINNER made within the **while**-loops by APPROX_1_MEDIAN, with an input of size n and using the parameters $t \geq 3$ and threshold $\vartheta \geq 1$. Plainly, $W(n, t, \vartheta) \leq W(n, t, 1)$, for any $\vartheta \geq 1$, and thus it will suffice to find an upper bound for $W(n, t, 1)$.

For notational convenience, let us put $W_1(n) = W(n, t, 1)$, where t has been fixed. It can easily be seen that $W_1(n)$ satisfies the following recurrence relation:

$$W_1(n) = \begin{cases} 0 & \text{if } 0 \leq n \leq 1, \\ 1 & \text{if } 2 \leq n < 2t, \\ \left\lfloor \frac{n}{t} \right\rfloor + W_1\left(\left\lfloor \frac{n}{t} \right\rfloor\right) & \text{if } n \geq 2t. \end{cases}$$

By induction on n, we show next that $W_1(n) \leq \left\lceil \frac{n}{t-1} \right\rceil$. For $n < 2t$, our estimate is trivially true. Thus, let $n \geq 2t$. Then, by inductive hypothesis, we have

$$\begin{aligned} W_1(n) &= \left\lfloor \frac{n}{t} \right\rfloor + W_1\left(\left\lfloor \frac{n}{t} \right\rfloor\right) \\ &\leq \left\lfloor \frac{n}{t} \right\rfloor + \left\lceil \frac{\left\lfloor \frac{n}{t} \right\rfloor}{t-1} \right\rceil \\ &= \left\lceil \left\lfloor \frac{n}{t} \right\rfloor + \frac{\left\lfloor \frac{n}{t} \right\rfloor}{t-1} \right\rceil \\ &= \left\lceil \frac{\left\lfloor \frac{n}{t} \right\rfloor \cdot t}{t-1} \right\rceil \leq \left\lceil \frac{n}{t-1} \right\rceil . \end{aligned}$$

Hence $W(n, t, \vartheta) \leq \left\lceil \frac{n}{t-1} \right\rceil$.

The number of distance computations made by a call WINNER(X) is equal to $\sum_{i=1}^{|X|}(i-1) = \frac{|X|(|X|-1)}{2}$. At each round of the tournament, all the calls to procedure WINNER have an argument of size t, with the possible exception of the last call, which can have an argument of size between $(t+1)$ and $(2t-1)$. Since there are $\lceil \log_t n \rceil$ rounds, it follows that the total number of distances computed by a call APPROX_1_MEDIAN(S), with $|S| = n$, constant tournament size t, and threshold ϑ, is upper bounded by the expression

$$\begin{aligned} & W(n, t, \vartheta) \cdot \frac{t(t-1)}{2} + \lceil \log_t n \rceil \cdot \left[\frac{(2t-1)(2t-2)}{2} - \frac{t(t-1)}{2} \right] + \frac{\vartheta(\vartheta-1)}{2} \\ & = \frac{t}{2}n + \mathcal{O}(\log n + \vartheta^2), \qquad (5.1) \end{aligned}$$

since the last call to procedure WINNER made within the **return** instruction of APPROX_1_MEDIAN has an argument of size at most ϑ. By taking $\vartheta = o(\sqrt{n})$, the above expression is easily seen to be $\frac{t}{2}n + o(n)$.

Summing up, we have the following theorem.

THEOREM 5.1. *Given an input set of size $n \in \mathbb{N}$, a constant tournament size $t \geq 3$, and a threshold $\vartheta = o(\sqrt{n})$, the algorithm* APPROX_1_MEDIAN *performs $\frac{t}{2}n + o(n)$ distance computations.*

6. Experimental results.

In this section we evaluate the practical performance of the algorithm APPROX_1_MEDIAN and compare it with Indyk's algorithm, by presenting and commenting various experimental results. Our algorithm is very precise and efficient. It performs better than Indyk's algorithm in terms of both running time and output precision.

Our implementations have been done in standard C (GNU-gcc compiler v.2.96) and all the experiments have been carried out on a PC Pentium III 1GHz with the Linux operating system (Mandrake distribution v.8.1). The source code is available on the web at the URL address: http://lipari.dmi.unict.it/∼ferro/source/1-median.tgz.

We tested several samples belonging to metric spaces of various dimensions with different Euclidean and non-Euclidean metrics. All samples were generated using the 48-bit linear congruential pseudorandom number generator written by Birgmeier [5]. All synthetic datasets were generated with uniform or nonuniform (clustered) distributions, using various metrics. We also tested our algorithm on datasets extrapolated from real world databases.

In the following, we continue to use the notation introduced in the previous sections. Thus,

- S will be the input set, namely a finite subset of a metric space (M, d);
- $w : S \longrightarrow \mathbb{R}$ will be the weight function defined by

$$w(x) = \sum_{s \in S} d(s, x),$$

  for $x \in S$, which we intend to minimize;
- t and *threshold* will be the parameters used in APPROX_1_MEDIAN.

Moreover, unless otherwise specified, in each of the following test sessions, we will assume that the parameter *threshold* has been set to $\min\left\{t^2 - 1, \lfloor\sqrt{n}\rfloor\right\}$, where n is the size of the input set S.

6.1. Asymptotic behavior of the algorithm.

Our first group of experiments refers to $5,000$ independent executions of our algorithm on a *fixed* input set S of size $n = 10^i$, with $2 \leq i \leq 5$. The input set has been taken as a uniformly distributed set of random points in the unit square, i.e., $[0, 1]^2$-space, with the Euclidean metric L_2.

The relative frequencies histograms reported in Figure 2 show the empirical distribution of the algorithm outputs, obtained with a tournament size $t = 3$. In each histogram, the abscissa contains an initial segment of S nondecreasingly sorted w.r.t. the ordering induced by $w()$; therefore, the leftmost element represents the exact 1-median of S.

The histograms in Figure 2 give the output precision of our algorithm in terms of the relative position w.r.t. the exact 1-median. Nevertheless, in many applications, it is more convenient to define the output quality in terms of the weight function $w()$, by introducing the following quantities relative to a generic input set S:

- $m_S = \min_{x \in S} w(x)$, the minimum weight in S, i.e., the weight of the exact 1-*median*;
- $M_S = \max_{x \in S} w(x)$, the maximum weight in S;
- $\mu_S = E[w(x)]$ and $\sigma_S = \sigma[w(x)]$ (with $x \in S$), i.e., the average and the standard deviation of weights in S;

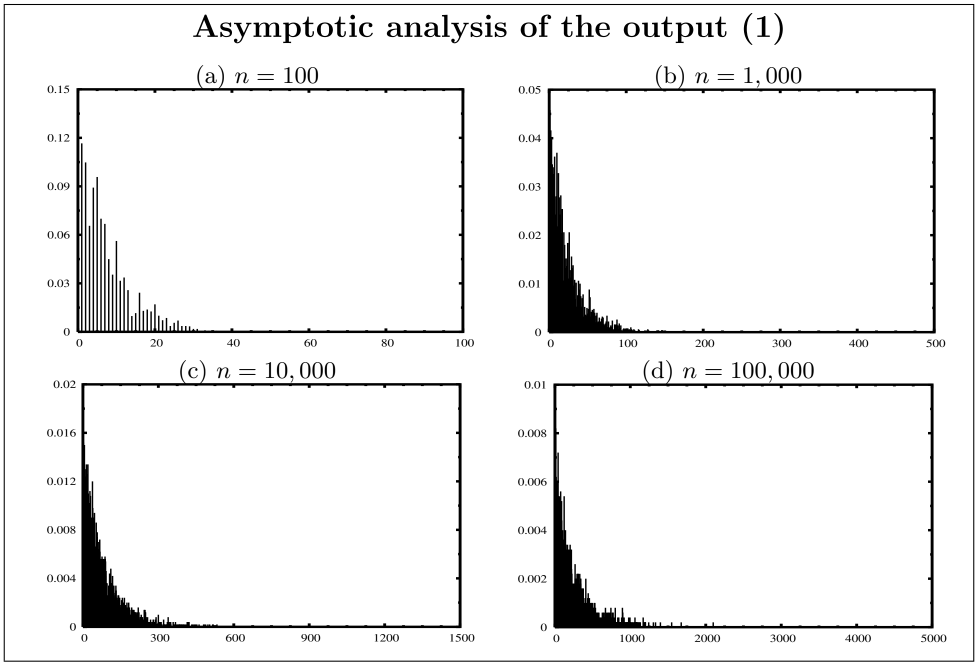


FIG. 2. *Relative frequencies histograms of the outputs. The abscissae refer to the elements in the input set with the smallest values for $w()$, nondecreasingly ordered. The ordinates show the corresponding frequencies.*

- $w_{out} = w(Output)$, the weight of the *Output* element returned by our algorithm on input S.

In the log-scale diagram in Figure 3, we report such statistical values, relative to a fixed random input set for each value of the size. Notice that two times the standard deviation is shown as a vertical straight-line centered on the mean value.

Statistics reported in Figure 3 can be further summarized by introducing the following value, relative to a single test on a random input set S:

- $\epsilon_{out} = \frac{w_{out} - m_S}{M_S - m_S} \cdot 100$, the percentage error distance defined w.r.t. the largest range of values $w(x)$, with $x \in S$; the extreme values assumed by ϵ_{out} are 0% and 100%, when the minimum- and maximum-weight elements in S are returned by the algorithm, respectively.

Figure 4(a) reports the average percentage error distance $E[\epsilon_{out}]$ and its standard deviation $\sigma[\epsilon_{out}]$ relative to our first experiment on a fixed set S. Moreover, Figure 4(b) refers to a similar experiment, but executed with *variable* input sets, namely, each time the algorithm is executed, a new input set S of size $n = 10^i$ is generated, for $2 \leq i \leq 5$.

It is to be noticed that, in both experiments, the percentage error can be approximated by $2^{4-\log_{10} n}$; in particular, an error smaller than 1% is obtained whenever the input size is large enough, for instance $n \geq 10,000$.

6.2. Threshold value analysis. We tested how the *threshold* value influences the quality of the output. By suitably tuning the value of the threshold, the precision of the algorithm can sensibly be improved, without affecting the efficiency (see section 4). Results refer to 5,000 iterations with a fixed input set of $n = 10^i$ random points, with $2 \leq i \leq 5$, chosen in the $[0,1]^2$-space with metric L_2, and where the tour-

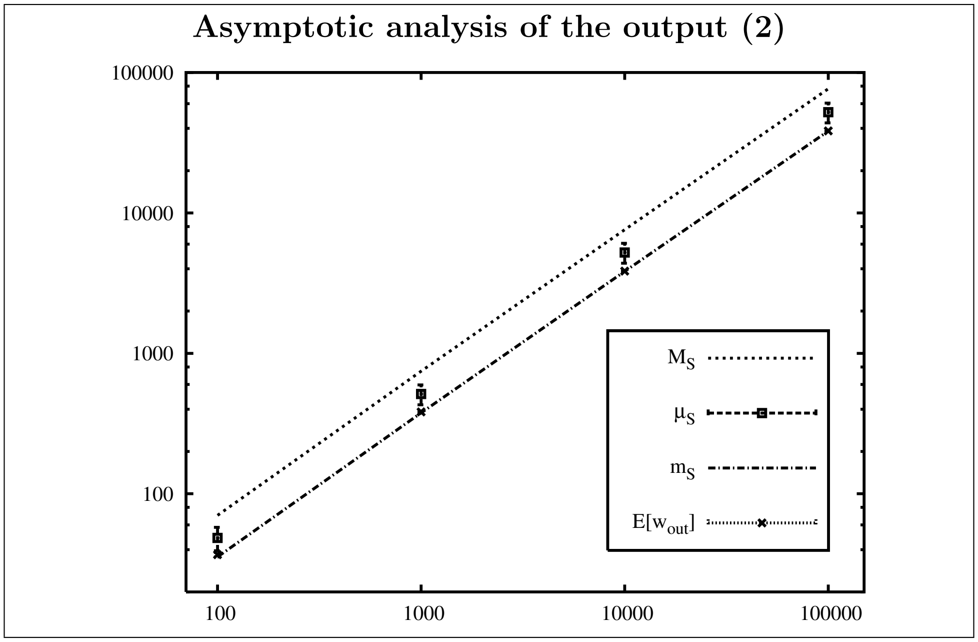


FIG. 3. *Diagram of the values m_S, M_S, μ_S, $E(w_{out})$, relative to a fixed random set S for each assigned value of the size.*

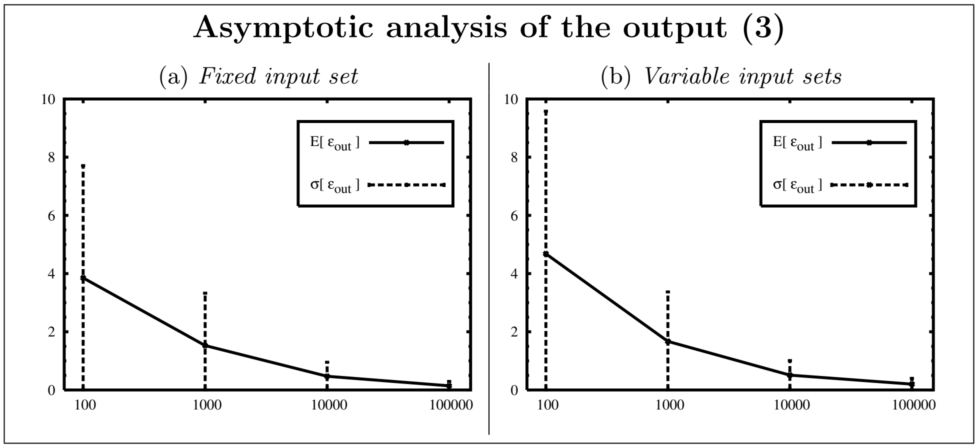


FIG. 4. *Average percentage error $E[\epsilon_{out}]$ and standard deviation $\sigma[\epsilon_{out}]$, w.r.t. the input size, on* (a) *fixed input set and* (b) *variable input sets (each time the algorithm is executed, a new input set S of size $n = 10^i$ is generated, for $2 \leq i \leq 5$.)*

nament size is $t = 3$. In Figure 5, we collected the average percentage error $E[\epsilon_{out}]$ for the following threshold values: $Thr = 8,\ 26,\ \lfloor\sqrt{n}\rfloor,\ 2\cdot\lfloor\sqrt{n}\rfloor$.

As expected, larger threshold values provide smaller average percentage errors, whenever the input size value is significant.

6.3. Distribution type analysis. We have analyzed some types of *nonuniform* distributions for the input sets. More specifically, we have examined uniform and skewed distributions of c clustered input sets containing $n = 10^i/c$ points each, with

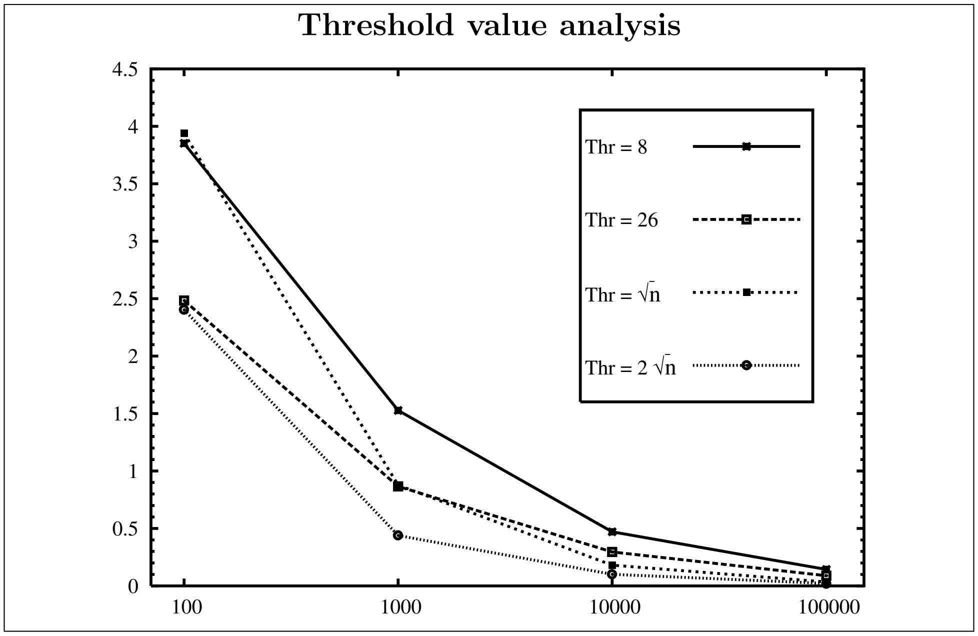


FIG. 5. *Average percentage error $E[\epsilon_{out}]$, w.r.t. the input size, for different threshold values.*

$i = 3, 4, 5$, and where $c = \lfloor \log_{10} n \rfloor$, 10, $\lfloor \sqrt{n} \rfloor$, $2 \cdot \lfloor \sqrt{n} \rfloor$. In the case of the uniform clustered distribution, we have generated c random clusters in $[0, 1]^2$, with uniform distribution in each cluster. Concerning the skewed clustered distribution, we have generated c random clusters with skewed distribution, i.e., clusters formed by very close points with no symmetry.

In both cases, clusters have been characterized by a parameter $\rho \in \mathbb{R}$, with $0 < \rho < \frac{1}{2}$, that determines the wideness of clusters. Larger values of ρ correspond to wider clusters with a high overlapping degree and, vice versa, smaller values of ρ correspond to nonoverlapping dense clusters. Such clusters have been generated using the same source code implemented for the experimental session reported in [6].

In our tests, the Euclidean metric L_2 was adopted, with a tournament size $t = 3$. The average percentage errors $E[\epsilon_{out}]$ are shown in Figure 6 for both types of distribution considered, with wideness factors $\rho = 0.2$ and $\rho = 0.4$.

Concerning the local inversions about skewed distribution in Figure 6(c), these are minimal and are due to the fact that each component of a tuple is very close to zero.

In general, for sufficiently large values of n, from the plots in Figure 6 it follows that (1) small cluster sizes are more difficult to treat than large ones, and (2) skewed distributions are more difficult than uniform ones.

6.4. Tournament size analysis. In the metric space $[0, 1]^2$, the choice of deriving local winners from triplets has shown to be convenient, but other values of the tournament size t could be used as well (see section 4). In general, computing local winners for larger subsets produces higher precision results, at increasing cost, as given by (5.1). In Figure 7, we report some results relative to experimental tests which used triplets, quadruples, quintets, and sextets. The experiment settings were similar to the ones reported in section 6.2, but with a fixed threshold value of 8.

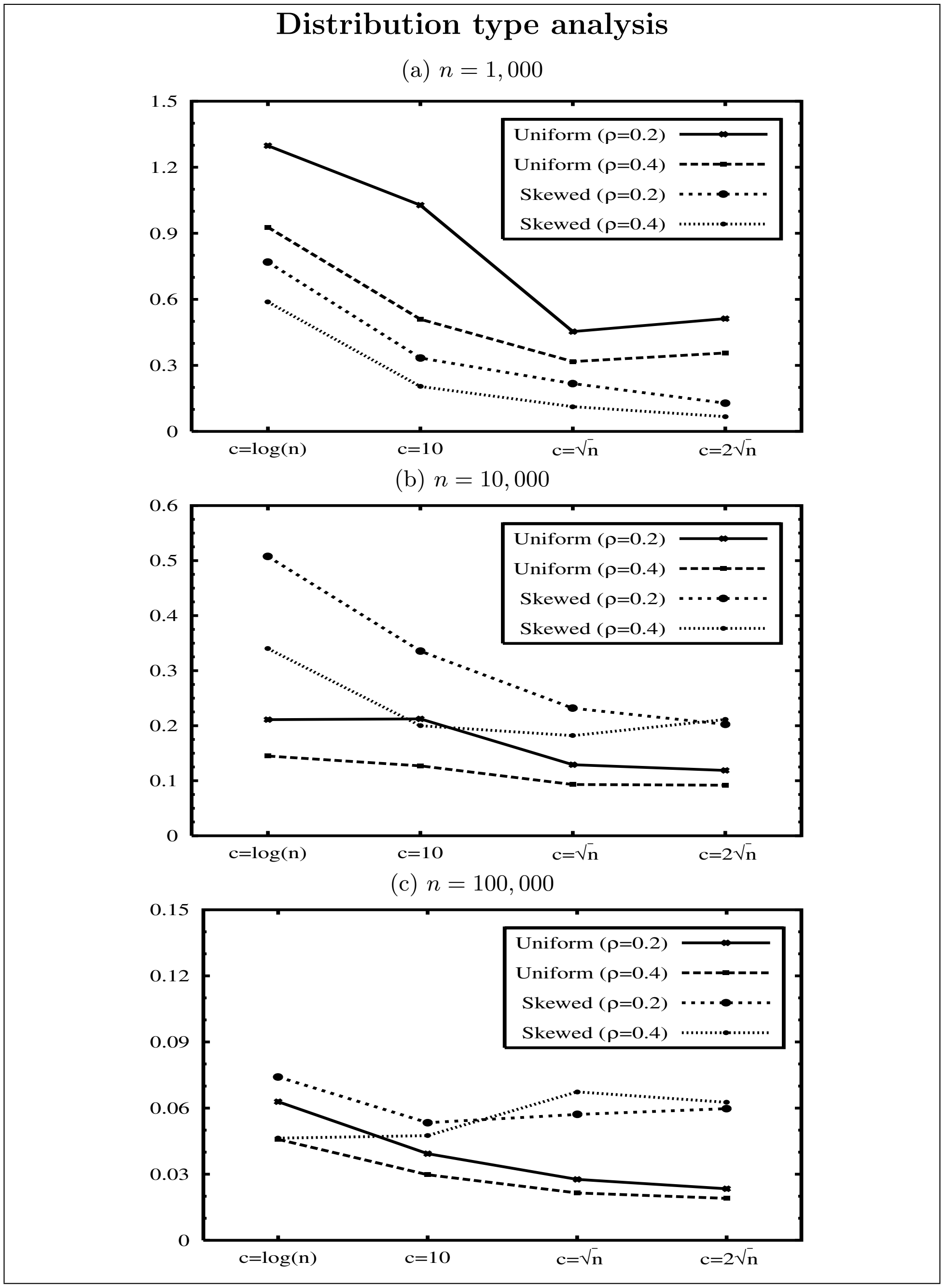


FIG. 6. *Average percentage error $E[\epsilon_{out}]$ for different types of clustered distribution w.r.t. the number of clusters.*

6.5. Space dimensionality analysis. Experimental results reported in Figure 8 allow one to evaluate the performance of our algorithm in the case of a $[0, 1]^m$ metric space equipped with the metric L_2, for $m = 5, 10, 20, 30$. Experiments have been performed with $5,000$ iterations over a fixed input set of $n = 10^i$ random uniformly distributed points, with $i = 3, 4, 5$.

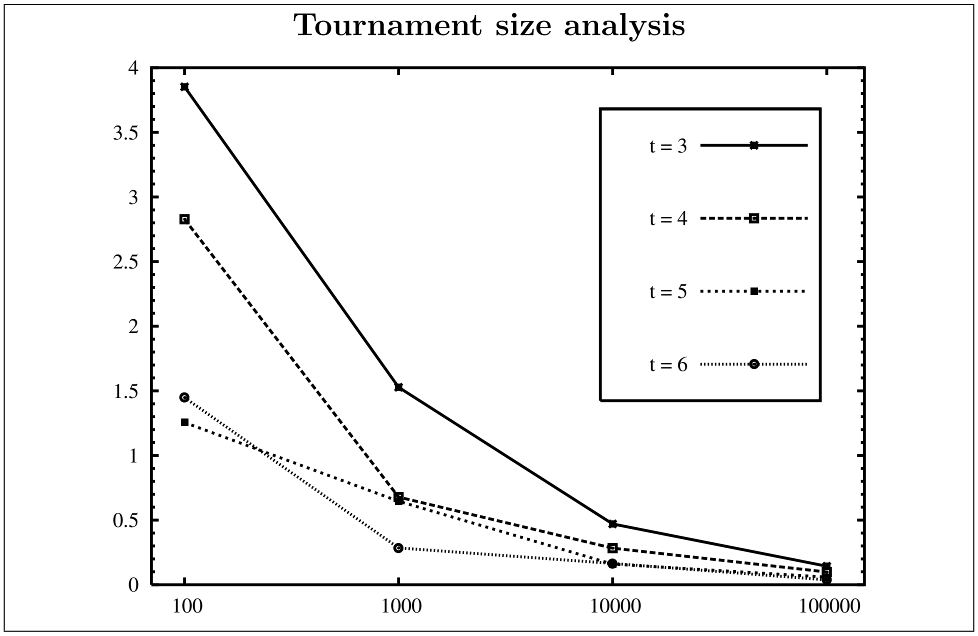


FIG. 7. *Average percentage error $E[\epsilon_{out}]$, w.r.t. the input size, for different tournament sizes.*

The average percentage errors $E[\epsilon_{out}]$ collected in Figure 8(a) and (b) refer, respectively, to tournament sizes of 3 and $m+1$ (see section 4).

By examining plot 8(a), we can argue that, in general, the error of our algorithm increases with the dimension of the metric space, when the tournament size is fixed. On the other hand, the plot of 8(b) shows that a tournament size equal to $m+1$ reduces the average percentage error $E[\epsilon_{out}]$ by a factor of 10 w.r.t. the value $t=3$; more specifically, an error smaller than 2% is obtained whenever the input size is large enough. We observe that it is not particularly significant to compare data series reported in the diagram of 8(b), as the tournament size is not constant for them.

6.6. Metric analysis. The following *metrics* in $[0,1]^m$, with $m = 5, 10, 20, 30$, have been tested: the Manhattan distance L_1, the Euclidean metric L_2, the Euclidean squared metric L_2^2, the Chebyshev distance L_∞, the Minkowski distance L_p, for $p = 10$. Experimental results refer to $5,000$ iterations over a fixed input set of $n = 10^i$ random uniformly distributed points, with $i = 3, 4, 5$. Data series collected in Figure 9 refer to the different metrics used; the average percentage error $E[\epsilon_{out}]$ has been obtained with a tournament size equal to $m+1$.

From the diagrams of Figure 9(b) and (c), one can see that the Chebyshev metric L_∞ and the Minkowski distance L_p are asymptotically harder than the remaining ones; moreover, excluding such two metrics, the percentage error is always less than 2%.

Concerning Figure 9(a), we can notice a different behavior with respect to (b) and (c). This is due to the tournament size which in the examined cases is too big relative to the size of the data set.

6.7. Real world datasets analysis. In this section we analyze the average percentage error $E[\epsilon_{out}]$ of our algorithm on input datasets extrapolated from real world databases.

Our first test has been run on input sets taken, with random sampling, from

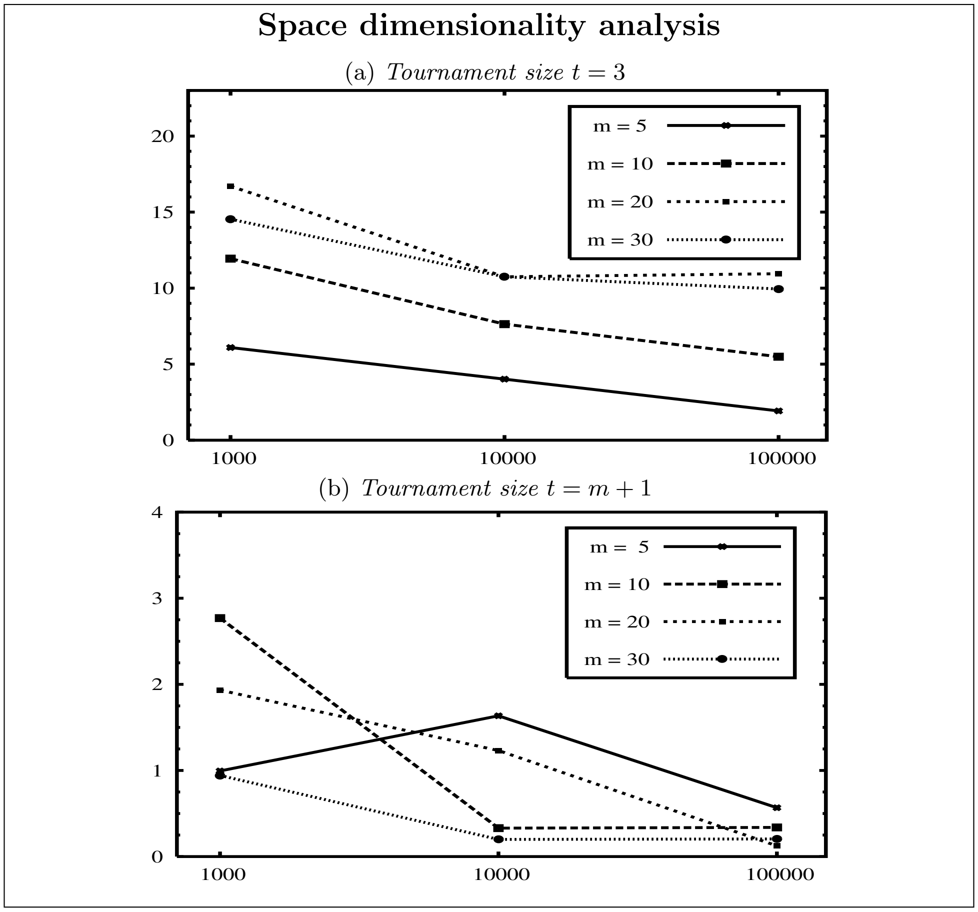


FIG. 8. *Average percentage error* $E[\epsilon_{out}]$*, w.r.t. the input size, for different space dimensions and tournament sizes.*

the metric space of strings, equipped with the minimum edit distance metric. More specifically, the experiments reported in Figure 10(a) refer to 5,000 iterations over a fixed input set of $n = 10^i$, with $i = 2, 3, 4$, randomly chosen strings from the *Linux dictionary*,[2] using tournament sizes $t = 3, 6, 9, 12$.

Our second test has been based on a set of $n = 10^i$ randomly selected images from the *Corel* images database, with $i = 2, 3, 4.$[3]. Each image has been characterized by its colors histograms, represented in the Euclidean metric space $\mathbb{R}^{32}$. The plot reported in Figure 10(b) refers to 5,000 iterations with tournament sizes $t = 3$, 33.

Our third and final test has been based on a set of biosequences drawn from the NCBI database.[4]. More precisely, we considered all the Alpha-Globins and the Beta-Globins listed in the database, for a total amount of about 2,500 biosequences, using the classical pairwise alignment metric computed by the linear space algorithm of Myers and Miller [28]. For this experiment, built with 5,000 iterations, we considered

[2] The dictionary is contained in the text file /usr/share/dict/linux.words, under the Linux Mandrake v.8.1.

[3] The *Corel* images database can be downloaded from the UCI Knowledge Discovery in Databases Archive at http://kdd.ics.uci.edu/

[4] The NCBI database can be downloaded from the Nucleotide database at http://www.ncbi.nlm.nih.gov/

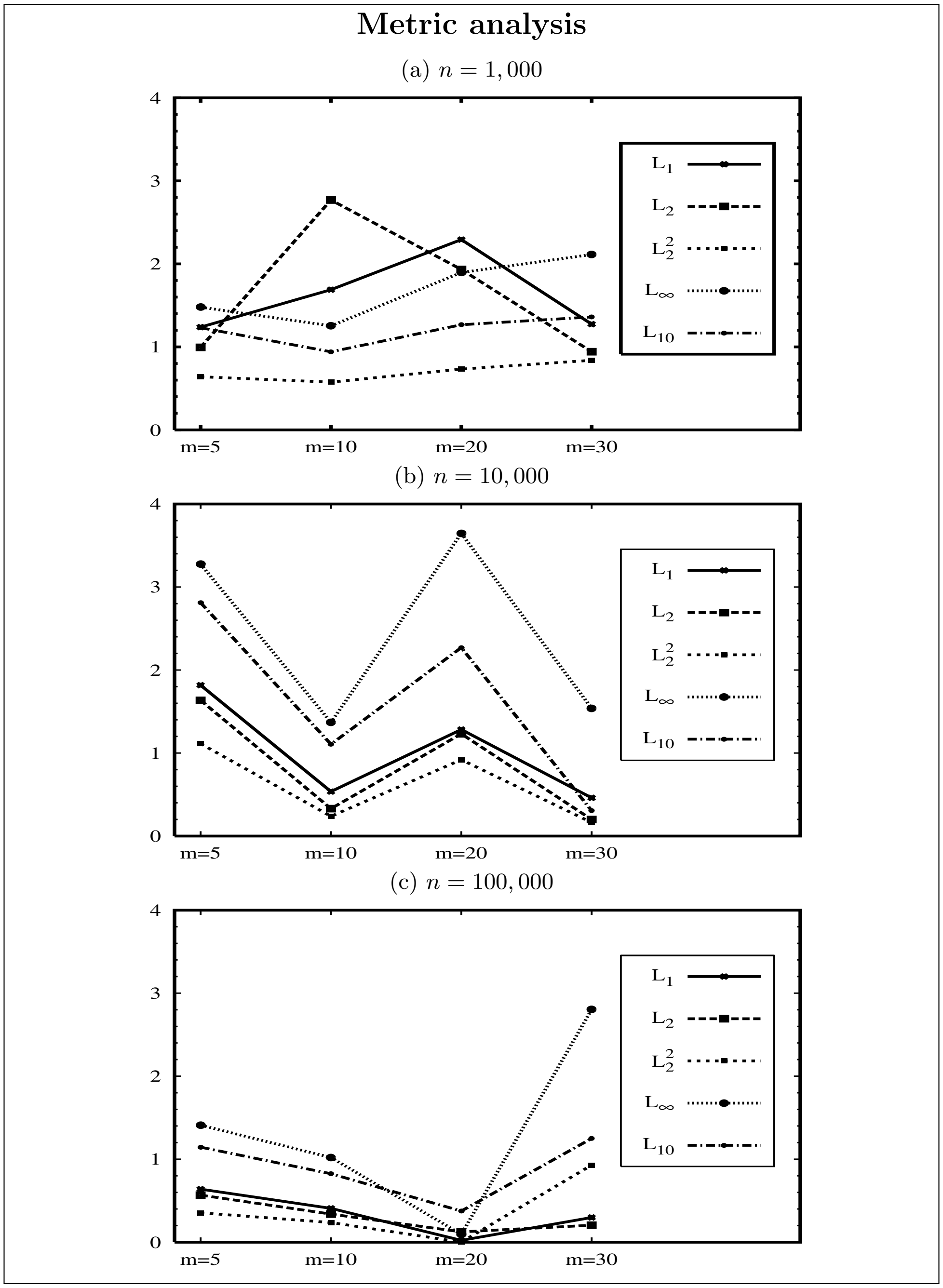


FIG. 9. *Average percentage error $E[\epsilon_{out}]$ for different space dimensions and metrics.*

sets of $n = 500 \cdot i$ biosequences randomly selected from the database, with $i = 1, 2, 3, 4$. Results reported in Figure 10(c) refer to a tournament size $t = 3$ and $t = 5$, which showed to be a good trade-off in the case of biosequences.

6.8. Comparison with Indyk's algorithm. In this section we report and comment various results relative to experimental data related to the APPROX_1_MEDIAN algorithm and Indyk's algorithm [21]. Such experiments show that our algorithm

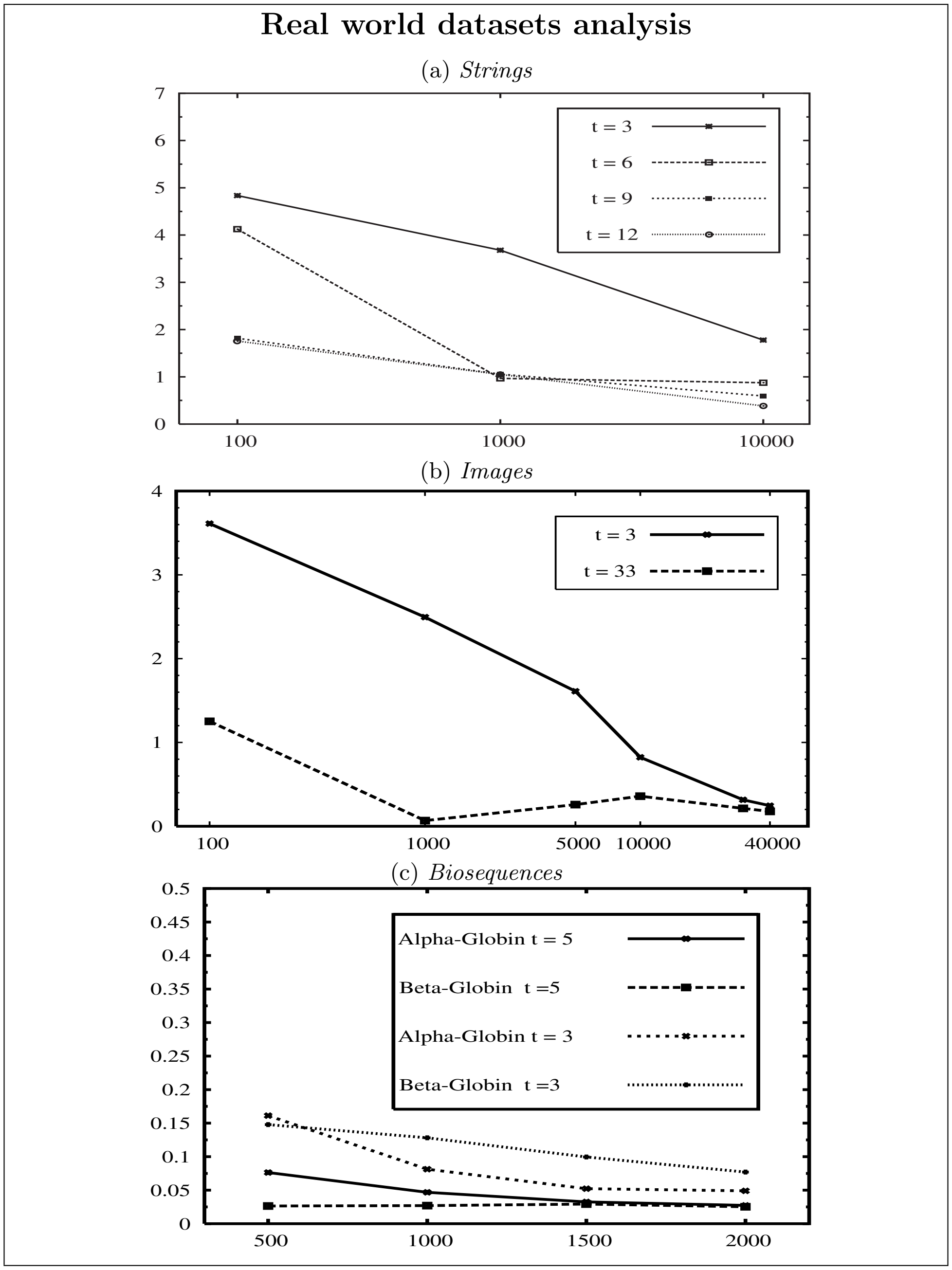


FIG. 10. *Average percentage error* $E[\epsilon_{out}]$ *for* (a) *the strings metric space with the minimum edit distance,* (b) *the images metric space with Euclidean distance, and* (c) *the biosequences metric space with pairwise distance alignment.*

outperforms Indyk's one w.r.t. both the number of computed distances and output precision. We recall that Indyk's algorithm, briefly described in section 1, has a $\mathcal{O}(n/\delta^5)$ running time and approximation factor $(1+\delta)$.

In Figures 11 and 12 the pseudocode of the Indyk's algorithm is sketched.

To obtain a coherent comparison, we considered input sizes n which are exact

```
                              The Indyk algorithm
INDYK $(S, \delta)$
    Input: A set $S$ of elements in a metric space $(M, d)$.
    Input: The maximum error allowed $\delta$.
    Output: The approximate 1-median element of $S$.
      Begin
          $S_t \leftarrow S$;
          $T \leftarrow \emptyset$;
          while $|S_t| > 1$ do
                while $|S_t| \geq 2$ do
                      Choose randomly $p, q \in S_t$;
                      $S_t \leftarrow S_t \setminus \{p, q\}$;
                      $T \leftarrow T \cup \{$PROBABILISTIC_WINNER $(p, q, S, \delta)\}$;
                end while;
                $S_t \leftarrow T$;
          end while;
          return $(S_t)$;
      End.
```

FIG. 11. *Pseudocode of the Indyk algorithm. For the sake of simplicity the pseudocode given refers to the case in which the size of the input set S is a power of 2.*

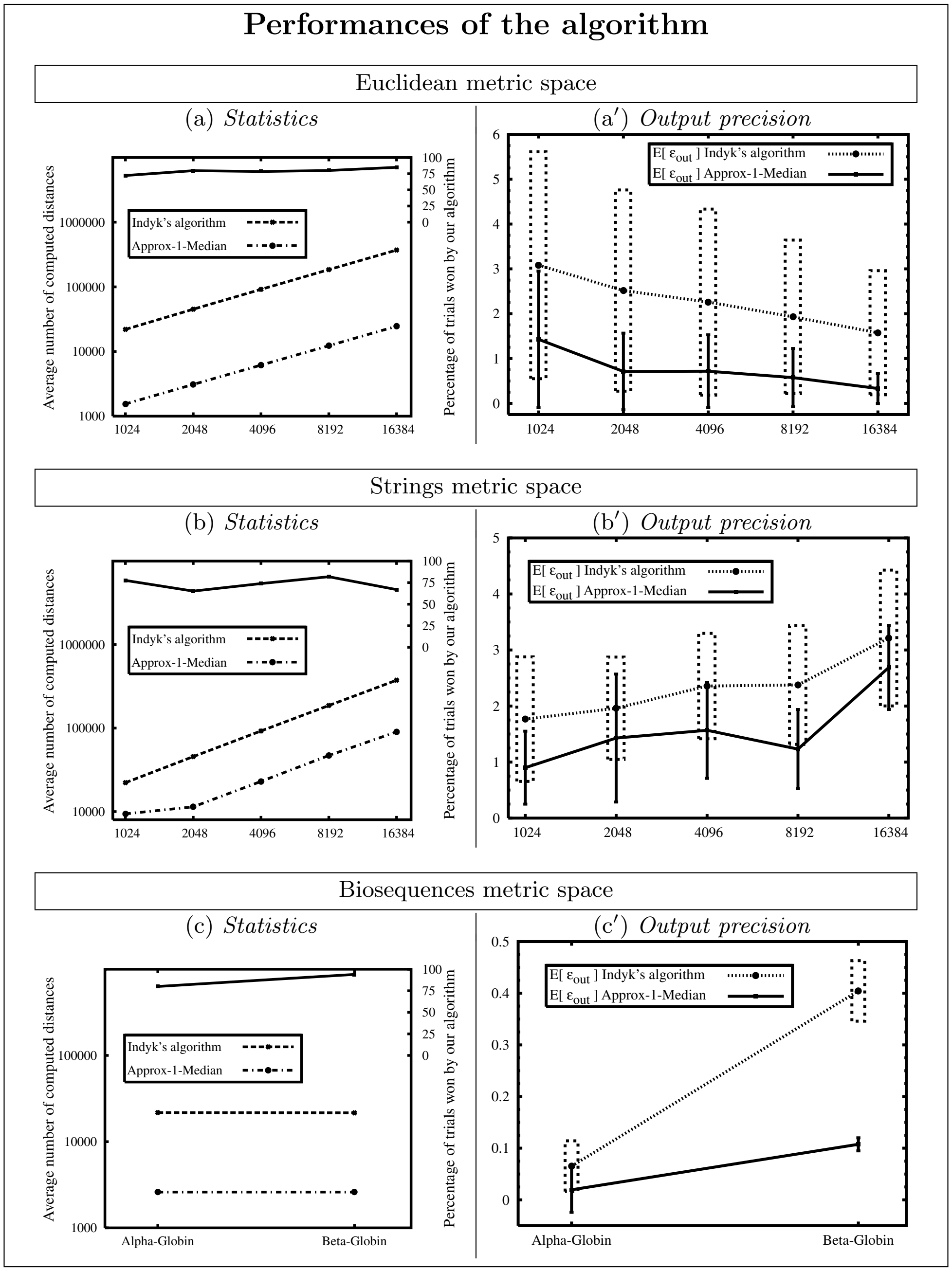
```
PROBABILISTIC_WINNER $(p, q, X, \delta)$
    Input: A set $X$ of elements in a metric space $(M, d)$.
    Input: $p, q$ extracted from $X$, $\delta$.
    Output: $p$ or $q$.
      Begin
          Let $R$ be a random sample of points in $X$ whose size
          is directly proportional to $d(p, q)$ and inversely
          proportional to $\delta^4$;
          Let $w_R(x) = \sum_{s \in R} d(s, x)$;
          Statistically estimate the number $\gamma$ of points in $X$ whose
          distance from $p$ is less than $4 \times d(p, q)/\delta + d(p, q)$;
          if $\gamma < \delta/4$, then
               return one random element in $\{p, q\}$;
          else
               return the element in $\{p, q\}$ which minimizes $w_R$;
      End.
```

FIG. 12. *Pseudocode of the probabilistic winner.*

powers of 2, because of the *binary* tournament generated by Indyk's algorithm. In particular, we executed $1,000$ iterations with input sets of size $n = 2^i$, for $10 \leq i \leq 14$, with the only exception of the biosequences session, in which we executed only 100 iterations with input sets of size $n = 1,024$. In addition, Indyk's approximation factor δ was appropriately chosen among different values in order to optimize its performance in terms of both precision and running time.

Tests were carried out with input sets of the following nature:

- uniformly distributed subsets of $[0, 1]^2$, with the Euclidean metric and tournament size $t = 3$;
- subsets of strings from the *Linux dictionary* with the minimum edit distance and tournament size $t = 9$ (see section 6.7);
- subsets of biosequences from the NCBI database with the Myers and Miller



Performances of the algorithm

Euclidean metric space

(a) *Statistics*

(a′) *Output precision*

Strings metric space

(b) *Statistics*

(b′) *Output precision*

Biosequences metric space

(c) *Statistics*

(c′) *Output precision*

FIG. 13. *Comparison between Indyk's algorithm and the* APPROX_1_MEDIAN *algorithm w.r.t. the number of computed distances and output precision.*

pairwise alignment metric [28] and tournament size $t = 5$ (see section 6.7).

The statistics in Figure 13(a), (b), and (c) report the results of comparing the two above-mentioned algorithms in terms of number of distances computed (left-ordinate) and percentage of tests won by the APPROX_1_MEDIAN algorithm w.r.t. precision (right-ordinate). Finally, Figure 13(a′), (b′), and (c′) report the average percentage error $E[\epsilon_{out}]$ of the two algorithms.

The above experimental results show that our algorithm is more precise and has a more stable behavior than Indyk's algorithm, as can also be deduced from the standard deviation $\sigma[\epsilon_{out}]$ reported as vertical bars (two times the value of $\sigma[\epsilon_{out}]$) in Figure 13(a′), (b′), and (c′). Results can be summarized by saying that, given a fixed input set, our algorithm

A. computes about 1/10 of the distances computed by Indyk's algorithm;

B. produces an output whose error is approximately one half of the output error produced by Indyk's algorithm.

6.9. Final remarks about experiments. Results collected here show that, with high probability, our algorithm computes elements contained in a very small neighborhood of the optimal solution, thus gaining strong support to its effectiveness. Specifically, in the cases of the synthetic and real datasets investigated, the average percentage error $E[\epsilon_{out}]$ was always less than 3%, for sufficiently large input sizes and when the algorithm's parameters were suitably tuned as suggested in section 4.

7. Conclusions. We proposed an approximate algorithm for the 1-median selection problem in general metric spaces which does not use immersion in Euclidean spaces. We also discussed experimental evidence of its efficiency and precision. In particular, we have successfully compared it with a performance guaranteed algorithm proposed in [21].

We believe that similar techniques can be applied to other optimization problems on metric spaces. In particular, we are currently investigating various clustering problems, the furthest pair problem, and some network management problems.

Acknowledgment. The authors are grateful to the anonymous referees for useful suggestions and comments.

REFERENCES

[1] K. D. Andersen, E. Christiansen, A. R. Conn, and M. L. Overton, *An efficient primal-dual interior-point method for minimizing a sum of Euclidean norms*, SIAM J. Sci. Comput., 22 (2000), pp. 243–262.

[2] V. Auletta, D. Parente, and G. Persiano, *Dynamic and static algorithms for optimal placement of resources in a tree*, Theoret. Comput. Sci., 165 (1996), pp. 441–461.

[3] C. Bajaj, *The algebraic degree of geometric optimization problems*, Discrete Comput. Geom., 3 (1988), pp. 177–191.

[4] S. Battiato, D. Cantone, D. Catalano, G. Cincotti, and M. Hofri, *An efficient algorithm for the approximate median selection problem*, in Proceedings of the 4th Italian Conference on Algorithms and Complexity (CIAC 2000), Lecture Notes in Comput. Sci. 1767, Springer-Verlag, New York, 2000, pp. 226–238.

[5] M. Birgmeier, *The* 48-*Bit Linear Congruential Pseudo-random Number Generator*, available online at http://www.ics.uci.edu/∼eppstein/projects/pairs/Source/testbed/rand48/.

[6] T. Bozkaya and M. Ozsoyoglu, *Indexing large metric spaces for similarity search queries*, ACM Trans. Database Systems, 24 (1999), pp. 361–404.

[7] R. E. Burkard, and J. Krarup, *A linear algorithm for the pos/neg-weighted* 1*-median problem on a cactus*, Computing, 60 (1998), pp. 193–216.

[8] D. Cantone, A. Ferro, A. Pulvirenti, D. Reforgiato, and D. Shasha, *Antipole tree indexing to support range search and k-nearest neighbor search in metric spaces*, IEEE Transaction on Knowledge and Data Engineering, 17 (2005), pp. 535–550.

[9] R. Chandrasekaran and A. Tamir, *Algebraic optimization: The Fermat–Weber location problem*, Math. Programming, 46 (1990), pp. 219–224.

[10] M. Charikar, S. Guha, E. Tardos, and D. Shmoys, *A constant-factor approximation for the k-median problem*, in Proceedings of the the 31st Annual ACM Symposium on Theory of Computing, Atlanta, GA, 1999, pp. 1–10.

[11] E. Chavez, G. Navarro, R. Baeza-Yates, and J. L. Marroquin, *Searching in metric spaces*, ACM Computing Surveys, 33 (2001), pp. 273–321.

[12] C. Di Pietro, V. Di Pietro, G. Emmanuele, A. Ferro, T. Maugeri, E. Modica, G. Pigola, A. Pulvirenti, M. Purrello, M. Ragusa, M. Scalia, D. Shasha, S. Travali, and V. Zimmitti, *ANTICLUSTAL: Multiple sequences alignment by antipole clustering and linear approximate* 1*-median computation*, in Proceedings of the IEEE Computer Society Bioinformatics Conference 2003 (CSB2003), Stanford, CA, 2003.

[13] C. Faloutsos, *Searching Multimedia Databases by Content*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[14] W. B. Frakes and R. Baeza-Yates, *Information Retrieval—Data Structures and Algorithms*, Prentice-Hall, Englewood Cliffs, NJ, 1992.

[15] G. N. Frederickson, *Parametric search and locating supply centers in trees*, in Algorithms and Data Structures, Lecture Notes in Comput. Sci. 519, F. Dehne, J.-R. Sack, and N. Santoro, eds., 1991, Springer, Berlin, pp. 299–319.

[16] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed., Academic Press, Boston, 1990.

[17] V. Ganti, R. Ramakrishnan, J. Gehrke, A. Powell, and J. French, *Clustering large datasets in arbitrary metric spaces*, in Proceedings of the 15th International Conference on Data Engineering (ICDE), Sydney, 1999, pp. 502–511.

[18] A. Goel, P. Indyk, and K. Varadarajan, *Reductions among high dimensional proximity problems*, in Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms, Washington, 2001, pp. 769–778.

[19] D. Gusfield, *Efficient methods for multiple sequence alignments with guaranteed error bounds*, Bull. Math. Biol., 55 (1993), pp. 141–154.

[20] D. S. Hochbaum, ed., *Approximation Algorithms for NP-Hard Problems*, PWS, Boston, 1997.

[21] P. Indyk, *Sublinear time algorithms for metric space problems*, in Proceedings of the 31st Annual ACM Symposium on Theory of Computing, Atlanta, 1999, pp. 428–432.

[22] P. Indyk, *Dimensionality reduction techniques for proximity problems*, in Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, 2000, pp. 371–378.

[23] K. Jain and V. Vazirani, *Primal-dual approximation algorithms for metric facility location and k-Median Problems*, in Proceedings of the FOCS '99, New York, 1999, pp. 2–13.

[24] K. Jain and V. Vazirani, *Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation*, J. ACM, 48 (2001), pp. 274–296.

[25] P. N. Klein and N. E. Young, *Approximation algorithms for NP-hard optimization problems*, in Algorithms and Theory of Computation Handbook, CRC Press, Boca Raton, 1999, pp. 34-1–34-19.

[26] Y. Li, *A Newton Acceleration of the Weiszfeld Algorithm for Minimizing the Sum of Euclidean Distances*, Technical report CTC95TR224, Computer Science Department of Cornell University, Cornell University, Ithaca, NY, November 1995.

[27] J. Lin and J. S. Vitter, *Approximation algorithms for geometric median problems*, Inform. Process Lett., 44 (1992), pp. 245–249.

[28] E. W. Myers and W. Miller, *Optimal alignments in linear space*, CABIOS, 4 (1988), pp. 11–17.

[29] E. Weiszfeld, *Sur le Point per lequel le somme des distances de n points est minimum*, Tohoku Math. J., 43 (1937), pp. 355–386.

[30] G. Xue and Y. Ye, *An efficient algorithm for minimizing a sum of Euclidean norms with applications*, SIAM J. Optim., 7 (1997), pp. 1017–1036.

[31] T. Zhang, R. Ramakrishnan, and M. Livny, *BIRCH: An efficient data clustering method for very large databases*, in Proceedings of the ACM SIGMOD Conference on Management of Data, Montreal, Canada, 1996, pp. 103–114.

SIAM J. Optim.
Vol. 16, No. 2, pp. 452–470

AN ALGEBRAIC CONDITION EQUIVALENT TO STRONG STABILITY OF STATIONARY SOLUTIONS OF NONLINEAR POSITIVE SEMIDEFINITE PROGRAMS[*]

TOSHIHIRO MATSUMOTO[†]

Abstract. This paper addresses strong stability, in the sense of Kojima, of stationary solutions of nonlinear positive semidefinite programs (NSDP). First, we give a characterization of stability from the point of view of one-to-one maps under an LICQ condition generalized to these programs. Second, under the same condition we construct a method for NSDP that is analogous to Kojima's method for classical nonlinear programs (NLP) treated in his famous paper. From this construction we make clear the essential difference between NSDP and NLP, and we deduce an algebraic condition equivalent to strong stability for those NSDP to which there does not exist this difference.

Key words. nonlinear programming, positive semidefinite programming, stationary solution, strong stability, stationary index, Lipschitz continuous map, implicit function theorem, nonsmooth analysis

AMS subject classifications. 90C25, 90C30, 90C31

DOI. 10.1137/S1052623402416797

1. Introduction.

In this section, we introduce nonlinear positive semidefinite programs that are nonlinear versions of linear positive semidefinite programs, and their stationary solutions and the concept of strong stability. Linear positive semidefinite programs (LSDP) are defined as

$$\left\| \begin{array}{ll} \text{minimize} & C \bullet X \\ \text{subject to} & X \in S_+(n), \\ & A_i \bullet X = b_i \quad (i = 1, \dots, \ell) \end{array} \right\},$$

where $S_+(n)$ is the set of $n \times n$ real symmetric positive semidefinite matrices, and $C,\ A_i,\ (1 \leq i \leq \ell)$, are $n \times n$ real symmetric matrices and $b = (b_1, \dots, b_\ell)$ is a vector of ℓ-dimensional Euclidean space $\boldsymbol{R}^\ell$ and $\bullet$ denotes the trace form. LSDP has been studied intensively this decade; for details, we recommend [10]. We refer to the following programs as nonlinear positive semidefinite programs (NSDP):

$$\text{Pro}(f, h) \qquad \left\| \begin{array}{ll} \text{minimize} & f(X) \\ \text{subject to} & X \in S_+(n), \\ & h_i(X) = 0 \quad (i = 1, \dots, \ell) \end{array} \right\},$$

where $f,\ h_i,\ (i = 1, \dots, \ell)$, are C^2 functions on the space $S(n)$ of $n \times n$ real symmetric matrices, i.e., $(f, h) \in \mathcal{F}$ in usage of notations defined in the section 2. Then, $X \in S_+(n) \bigcap \mathcal{N}(h)$ is called a stationary solution of program $\text{Pro}(f, h)$ if $-D_X f(X) \in \boldsymbol{R} D_X h(X) + \sigma(X)$ holds. Here, $\mathcal{N}(h) = \{X \in S(n) : h_i(X) = 0\ (\forall i)\}$ and $\boldsymbol{R} D_X h(X)$ denotes the affine space spanned by $\{D_X h_i(X) : i = 1, \dots, \ell\}$, and $\sigma(X)$ the normal cone of $S_+(n)$ at X, i.e., $\sigma(X) = \{G \in S(n) : (Y - X) \bullet G \leq 0\ (\forall Y \in S_+(n))\}$. The stationary solution X is defined to be strongly stable if there exist $\delta > 0$ and $\alpha > 0$

[*] Received by the editors October 31, 2002; accepted for publication (in revised form) March 8, 2005; published electronically November 14, 2005.

http://www.siam.org/journals/siopt/16-2/41679.html

[†] Department of Media Science, Teikyo University of Science and Technology, Yatsuzawa 2525, Uenohara-machi, Kitatsuru-gun, Yamanashi 409-0193, Japan (matsu@ntu.ac.jp).

such that, for any $(f, h) \in \mathcal{F}$ satisfying $\|f - f'\| < \alpha$ and $\|h - h'\| < \alpha$, there exists a unique stationary solution $X(f', h') \in \{X' \in S_+(n) : \|X - X'\| \leq \delta\}$ of $\mathrm{Pro}(f', h')$ and the correspondence $(f', h') \mapsto X(f', h')$ is continuous at (f, h). We state the norm used in this definition and the definition itself more precisely in section 2.

Similar programs are treated by Bonnans and Shapiro [1]. Given a Banach space U and a map $G : \boldsymbol{R}^n \times U \to S_+(p)$, they investigated sensitivity of parameterized semidefinite programs (SDP) problems in the form

$$\left\| \begin{array}{l} \text{minimize } f(x, u) \\ \text{subject to } (x, u) \in \boldsymbol{R}^n \times U : \ G(x, u) \in S_+(p) \end{array} \right\}$$

and gave a sufficient condition for directional Lipschitz stability. They also treated more general programs and showed that strong stability is equivalent to the second order growth condition when local minimum solutions are considered. However, no criterion of strong stability has yet been found for these programs in general cases.

In [9], Kojima introduced, for the first time, the concept of strong stability of stationary solution for nonlinear programs which have finite equality constraints and finite inequality constraints of C^2 class satisfying the so-called Mangasarian–Fromovitz condition. We refer to programs of this type as NLP; Kojima also gave an algebraic condition that is necessary and sufficient for the stability by means of Jacobian and Hessian matrices. Since then, strong stability for programs of this type has been intensively studied and it is known that various kinds of regularities are equivalent to that stability (see [7], [8]). However, since it is not known that LSDP and NSDP have finite inequality constraints of C^2 class satisfying the Mangasarian–Fromovitz condition, we cannot apply Kojima's theory directly to LSDP and NSDP. The purpose of this paper is to present a condition equivalent to strong stability from the point of view of one-to-one maps under an LICQ condition generalized to NSDP, and to make clear the essential difference between NSDP and NLP by construction of an analogous method for NSDP to the method exploited by Kojima in his famous paper [9]. For those NSDP to which there does not exist this difference, we can deduce the similar result as in [9].

In section 2,

- we define stationary solutions and strong stability, and we prepare a series of elementary results and facts.

In section 3,

- we characterize strong stability from the point of view of one-to-one maps under an LICQ condition.

In section 4,

- we calculate the generalized Jacobians of $\rho^+(X) = X^+$ and $\rho^-(X) = X^-$, where $X^+ = \arg \text{minimize } \|X - Y\|$ subject to $Y \in S_+(n)$ and $X^- = X - X^+$, and
- we prove that all eigenvalues of the generalized Jacobian of ρ^+ and ρ^- lie between 0 and 1 on the real line.

In section 5, under an LICQ condition,

- we prove that all eigenvalues of A are real for any $A \in \partial_{(X,\lambda)}\psi(X, \lambda; f, h)$, where ψ is defined associated to program $\mathrm{Pro}(f, h)$ in section 2,
- we construct a method for NSDP analogous to Kojima's method for NLP in [9],
- we make clear the essential difference between NSDP and NLP, and
- we deduce an algebraic condition equivalent to stability for those NSDP to which there does not exist the above difference.

Before ending the introduction we refer the reader to the paper [16] of Pang, Sun, and Sun. Using different methods they treat NSDP with no equality constraints and characterize strong stability by B-subdifferential. Our manuscript of this paper was finished before their paper has appeared.

2. Preliminaries. In this section, we define strong stability in the sense of Kojima and we prepare a series of elementary results and facts. For their preparation, we list the following notations used in this paper:

$\boldsymbol{R}$: the field of all real numbers,

$\boldsymbol{R}^\ell$: the ℓ-dimensional Euclidean space,

$\mathcal{M}(m,n)$: the set of all $m \times n$ real matrices,

$End_{\boldsymbol{R}}(V)$: the set of all $\boldsymbol{R}$ linear maps from V to V for a linear space V,

$S(n)$: the set of all $n \times n$ symmetric real matrices,

$S_+(n)$: the set of all $n \times n$ positive semidefinite symmetric real matrices,

$S_{++}(n)$: the set of all $n \times n$ positive definite symmetric real matrices,

$S_-(n)$: the set of all $n \times n$ negative semidefinite symmetric real matrices,

$S_{r,s}(n)$: the set of all $n \times n$ symmetric real matrices with r positive eigenvalues and s negative eigenvalues,

$S^*(n)$: the set of all $n \times n$ nonsingular symmetric real matrices,

$D(n)$: the set of all $n \times n$ diagonal real matrices,

$\mathrm{Diag}(\gamma_1, \ldots, \gamma_n)$: an $n \times n$ diagonal matrix whose (i,i) component is $\gamma_i (1 \le i \le n)$,

$O(n)$: the set of all $n \times n$ orthogonal real matrices,

I_r : the $r \times r$ identity matrix, i.e., the identity map on $\boldsymbol{R}^r$,

I_A : the identity map on A for any set A,

I : the identity matrix of an appropriate size,

O_r : the $r \times r$ zero matrix,

O : the zero matrix of an appropriate size,

E_{ij} : the elementary matrix whose (i,j) entry is 1 and other entries are 0's,

X^T : the transposition of the matrix X,

$A \bullet B$: the trace form of $m \times n$ matrices $A = (a_{ij})$ and $B = (b_{ij})$, i.e.,

$$A \bullet B = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} b_{ij},$$

$t^+ = \max\{t, 0\}$ for a real number t,

$t^- = \min\{t, 0\}$ for a real number t,

$$sgn\ t = \begin{cases} 1 & (t > 0), \\ 0 & (t = 0), \\ -1 & (t < 0), \end{cases}$$

$A \setminus B = \{x \in A : x \notin B\}$,

$\mathrm{int}(A)$: the interior of a subset A of a topological space X,

$c\ell(A)$: the closure of a subset A of a topological space X,

$$
\begin{aligned}
\mathrm{conv}A\ &:\ \text{the convex hull of a subset } A \text{ of a vector space } V,\\
ex(K)\ &:\ \text{the set of extremal points of a convex set } K,\\
A+B &= \{a+b : a\in A, b\in B\} \text{ for subsets } A \text{ and } B \text{ of a vector space } V,\\
d(A,B) &= \inf\{d(a,b) : (a,b)\in A\times B\} \text{ for subsets } A, B\subset X, \text{ where } X \text{ is}\\
&\quad \text{a metric space with its metric } d : X\times X\to \boldsymbol{R},\\
\mathcal{F} &= \{(f,h) = (f,h_1,\ldots,h_\ell) : f,h_1,\ldots,h_\ell \in C^2(S(n))\},\\
&\quad \text{where } C^2(S(n)) \text{ is the set of all functions on } S(n) \text{ of } C^2 \text{ class},\\
\mathcal{N}(h) &= \{X\in S(n) : h(X) = \mathbf{0}\} \text{ for } (f,h)\in\mathcal{F},\\
F|A\ &:\ \text{the restriction of a map } F \text{ to a subset } A \text{ of its domain},\\
T_X\mathcal{M}\ &:\ \text{the tangent space of a } C^1\text{-manifold } \mathcal{M} \text{ at } X\in\mathcal{M}.
\end{aligned}
$$

Since program $\mathrm{Pro}(f,h)$ has a domain constraint, i.e., $X \in S_+(n)$, we need the normal cone $\sigma(X)$ of $S_+(n)$ at $X \in S_+(n)$.

DEFINITION 2.1. *Let $X \in S_+(n)$. The normal cone $\sigma(X)$ of $S_+(n)$ at X is defined by $\sigma(X) = \{G \in S(n) : (Y - X) \bullet G \le 0\ (\forall Y \in S_+(n))\}$; $\boldsymbol{R}\sigma(X)$ denotes the affine space spanned by $\sigma(X)$.*

For $\mathcal{S} \subset S(r)$ and $\mathcal{T} \subset S(n-r)$, we define a set $\mathcal{S}\times\mathcal{T} = \left\{\begin{pmatrix} A & O \\ O & B\end{pmatrix} : A\in\mathcal{S}, B\in\mathcal{T}\right\}$. We abbreviate $\{O_r\}\times\mathcal{T}$ to $O_r\times\mathcal{T}$. The next fact is readily proved.

FACT 2.2 (see [14]). *Let $X \in S_+(n)$. Then* (i), (ii), *and* (iii) *hold.*

(i) $\sigma(X) = \{G \in S_-(n) : G \bullet X = 0\}$.

(ii) $\sigma(PXP^T) = P\sigma(X)P^T = \{PGP^T : G \in \sigma(X)\}$ *holds for any $P \in O(n)$.*

(iii) *Let $X \in S_{r,0}(n)$. Suppose $X \in P(D(r)\times O_{n-r})P^T$ with $P \in O(n)$. Then $\sigma(X) = P(O_r \times S_-(n-r))P^T$.*

Taking a local coordinate system shows that $S_{r,s}(n)$ is a $\frac{(r+s)(2n-r-s+1)}{2}$ dimensional analytic submanifold of $S(n)$. Its tangent space $T_XS_{r,s}(n)$ is explicitly stated as in Fact 2.3. We denote by $(T_XS_{r,s}(n))^\perp = \{Z \in S(n) : Z \bullet Y = 0\ (\forall Y \in T_XS_{r,s}(n))\}$ the orthogonal complementary space of $T_XS_{r,s}(n)$ in $S(n)$ with respect to the inner product defined by the trace form. Since $S(n) = \boldsymbol{R}^{\frac{n(n+1)}{2}}$ as affine spaces, in this paper we consider that $T_XS(n) = S(n)$ for any $X \in S(n)$. The following fact can be proved directly.

FACT 2.3 (see [14]). *Let $X \in S_{r,0}(n)$ and suppose that $X = P\begin{pmatrix}\Gamma_{11} & O\\ O & O\end{pmatrix}P^T$, where $P \in O(n)$ and $\Gamma_{11} \in S_{r,0}(r)$. Then the following* (i) *and* (ii) *hold.*

(i) $T_XS_{r,0}$= $\left\{P\begin{pmatrix}\dot Y_{11} & \dot Y_{21}^T\\ \dot Y_{21} & O\end{pmatrix}P^T : \dot Y_{11} \in S(r)\ \textit{and}\ \dot Y_{21} \in \mathcal{M}(n-r,r)\right\}$.

(ii) $\boldsymbol{R}\sigma(X) = (T_XS_{r,0})^\perp = \left\{P\begin{pmatrix}O & O\\ O & \dot Y_{22}\end{pmatrix}P^T : \dot Y_{22} \in S(n-r)\right\} = P(O_r\times S(n-r))P^T$.

The positive part Z^+ and negative part Z^- of a matrix Z are useful concepts in what follows. In the next definition, $\|\cdot\|$ denotes a norm on $S(n)$ by $\|X\| = \sqrt{X \bullet X}$ for $X \in S(n)$.

DEFINITION 2.4.

(i) *For $Z \in S(n)$, it follows from Fact* 4.1 *in section* 4 *that there exists a unique matrix $Y \in S_+(n)$ minimizing $\|Y - Z\|$. We denote it by Z^+. Similarly, there exists a unique matrix $Y \in S_-(n)$ minimizing $\|Y - Z\|$ and we denote it by Z^-. From their definition, Z^+ and Z^- are continuous with respect to Z. It is well known* (*see* [14]) *that*

(a) $Z = Z^+ + Z^-$,

(b) *$Z^+ \bullet Z^- = 0$,*

(c) *$(PZP^T)^+ = PZ^+P^T$ and $(PZP^T)^- = PZ^-P^T$ hold for any $P \in O(n)$. We define $\rho^+, \rho^- : S(n) \to S(n)$ by $\rho^+(Z) = Z^+$ and $\rho^-(Z) = Z^-$.*

(ii) *Let $\mathcal{H} = \{(X, G) \in S(n) \times S(n) : X \in S_+(n)$ and $G \in \sigma(X)\}$,* i.e., *$\mathcal{H}$ is the set of complementary matrices. We define $\eta : \mathcal{H} \to S(n)$, $\rho : S(n) \to \mathcal{H}$ by $\eta(X, G) = X + G$ and $\rho(Z) = (\rho^+(Z), \rho^-(Z)) = (Z^+, Z^-)$. Notice that ρ and η are homeomorphisms because of $\rho = \eta^{-1}$.*

Let $\mathcal{M}$ be a C^1 manifold and $\mathcal{N} \subset \mathcal{M}$ be a C^1-submanifold of $\mathcal{M}$ and $\bar{x} \in \mathcal{N}$ and U be a neighborhood of $\bar{x}$ in $\mathcal{M}$. Consider the coordinate system x of $\mathcal{M}$ around $\bar{x}$ and the coordinate system y of $\mathcal{N}$ around $\bar{x}$. Then, the natural immersion $\mathcal{N} \subset \mathcal{M}$ is represented by a unique C^1-map $x = \nu(y)$. Let $f : U \to \boldsymbol{R}^n$ be a C^1 map. Then, we use the notation $D_y f(\bar{x})$, whose meaning we define to be $D_y f(\bar{x}) = D_x f(\bar{x}) D_y \nu(\bar{x})$. We identify functions on $S(n)$ with those on $\mathcal{M}(n)$ satisfying $f(X) = f(X^T)$, $(\forall X \in \mathcal{M}(n))$. In this situation, it is readily apparent that $D_X f(X) \in S(n)$.

The gradient matrix $D_X f(X)$ and the Hessian tensor $D_X^2 f(X)$ of $f \in C^2(S(n))$ can be represented explicitly as

$$\begin{cases} D_X f(X) = \sum_{i=1}^n \sum_{j=1}^n D_{x_{ij}} f(X) E_{ij} \in S(n), \\ D_X^2 f(X) = \sum_{p=1}^n \sum_{q=1}^n \sum_{i=1}^n \sum_{j=1}^n D_{x_{pq}} D_{x_{ij}} f(X) E_{pq} \otimes E_{ij} \in S(n) \times S(n), \end{cases}$$

where $\otimes$ denotes the Kronecker product (see [4]). In general, for subsystems $X_1 = (x_{pq})_{(p,q)\in\Lambda_1}$ and $X_2 = (x_{ij})_{(i,j)\in\Lambda_2}$ of X, we define

$$D_{X_1} D_{X_2} f(X) = \sum_{(p,q)\in\Lambda_1} \sum_{(i,j)\in\Lambda_2} D_{x_{pq}} D_{x_{ij}} f(X) E_{pq} \otimes E_{ij} \in S(n) \otimes S(n).$$

The norms $\|D_X f(X)\|$ and $\|D_X^2 f(X)\|$ are induced by the trace form, i.e.,

$$\begin{cases} \|D_X f(X)\| = \sqrt{\sum_{i=1}^n \sum_{j=1}^n |D_{x_{ij}} f(X)|^2}, \\ \|D_X^2 f(X)\| = \sqrt{\sum_{p=1}^n \sum_{q=1}^n \sum_{i=1}^n \sum_{j=1}^n |D_{x_{pq}} D_{x_{ij}} f(X)|^2} \end{cases}.$$

For $B \subset S(n)$ we use the following norms:

$$\begin{cases} \|f\|_B = \sup\{|f(X)|, \|D_X f(X)\|, \|D_X^2 f(X)\| : X \in B\} \text{ for } f \in C^2(S(n)), \\ \|(f,h)\|_B = \max\{\|f\|_B\ ,\ \|h_i\|_B : 1 \le i \le \ell\} \text{ for } (f,h) \in \mathcal{F} \end{cases}$$

and denote by $\mathcal{F}_B$ the space $\mathcal{F}$ with $\|\cdot\|_B$-topology. In general, given a normed vector space V with its norm $\|\cdot\|$, we define a closed ball and an open ball by $B_\delta(x) = \{y \in V : \|y - x\| \le \delta\}$ and $\mathrm{int}(B_\delta(x)) = \{y \in V : \|y - x\| < \delta\}$ for $x \in V$ and a positive real number $\delta > 0$.

Definition 2.5. *Let $(f, h) \in \mathcal{F}$. $\boldsymbol{R} D_X h(X) = \sum_{i=1}^\ell \boldsymbol{R} D_X h_i(X)$ denotes the affine space spanned by $\{D_X h_i(X) : i = 1, \ldots, \ell\}$. Then $\bar{X} \in S_+(n)$ is called a stationary solution of program* Pro(f, h) *if $-D_X f(\bar{X}) \in \boldsymbol{R} D_X h(\bar{X}) + \sigma(\bar{X})$ holds. Also, $(\bar{X}, \bar{G}, \bar{\lambda}) \in \mathcal{H} \times \boldsymbol{R}^\ell$ is called a stationary point of program* Pro(f, h) *if $D_X f(\bar{X}) + \sum_{i=1}^\ell \bar{\lambda}_i D_X h_i(\bar{X}) + \bar{G} = O$ holds. Identifying $\mathcal{H} \times \mathcal{F}$ with $S(n) \times \mathcal{F}$ by $\rho \times I_{\mathcal{F}}$, $(\bar{Z}, \bar{\lambda}) \in S(n) \times \boldsymbol{R}^\ell$ is also called a stationary point of program* Pro(f, h) *if $(\rho(\bar{Z}), \bar{\lambda})$ is a stationary point of program* Pro(f, h)*, i.e., if both $D_X f(Z^+) + \sum_{i=1}^\ell \lambda_i D_X h_i(Z^+) + Z^- = O$ and $h(Z^+) = \mathbf{0}$ hold, where $\mathbf{0}$ denotes the zero vector of $\boldsymbol{R}^\ell$.*

The following are some notations for the remainder of this paper. For $(f,h) \in \mathcal{F}$, we define $L(X,\lambda;f,h) : S(n) \times \boldsymbol{R}^\ell \to \boldsymbol{R}$, $\psi(\cdot,\cdot;f,h) : S(n) \times \boldsymbol{R}^\ell \to S(n) \times \boldsymbol{R}^\ell$, $\Omega \subset S(n) \times \boldsymbol{R}^\ell \times \mathcal{F}$, $\Xi \subset S(n) \times \mathcal{F}$, and $\chi : \Omega \to \Xi$ as follows:

$$\begin{aligned}
L(X,\lambda;f,h) &= f(X) + \sum_{i=1}^{\ell} \lambda_i h_i(X),\\
\psi(Z,\lambda;f,h) &= (D_X L(Z^+,\lambda;f,h) + Z^-, D_\lambda L(Z^+,\lambda;f,h))\\
&= \left(D_X f(Z^+) + \sum_{i=1}^{\ell} \lambda_i D_X h_i(Z^+) + Z^-, h(Z^+) \right),\\
\Omega &= \{(Z,\lambda,f,h) \in S(n) \times \boldsymbol{R}^\ell \times \mathcal{F} : (Z,\lambda) \text{ is a stationary point of } \mathrm{Pro}(f,h)\}\\
&= \{(Z,\lambda,f,h) \in S(n) \times \boldsymbol{R}^\ell \times \mathcal{F} : \psi(Z,\lambda,f,h) = (O,\mathbf{0})\},\\
\Xi &= \{(X,f,h) \in S(n) \times \mathcal{F} : X \text{ is a stationary solution of } \mathrm{Pro}(f,h)\},\\
\chi(Z,\lambda,f,h) &= (Z^+,f,h), \text{i.e.}, \chi : \Omega \to \Xi \text{ is a natural projection.}
\end{aligned}$$

Definition 2.6 (see [7], [9]). *Let $\bar{X} \in S_+(n)$ be a stationary solution of* $\mathrm{Pro}(\bar{f},\bar{h})$. *$\bar{X}$ is said to be strongly stable if there exist neighborhoods $U = B_\delta(\bar{X})$ of $\bar{X}$ in $S(n)$ and V of $(\bar{f},\bar{h})$ in $\mathcal{F}_U$ such that the natural projection $pr : \Xi \bigcap (U \times V) \to V$ is bijective and $pr^{-1} : V \to \Xi \bigcap (U \times V)$ is continuous at $(\bar{f},\bar{h})$.*

We refer to the following condition as LICQ condition 2.7 since, under this condition, each stationary solution corresponds to a unique stationary point and this condition takes a role in programs $\mathrm{Pro}(f,h)$ just as LICQ condition does in the setting of [9].

Condition 2.7.

(i) $D_X h_i(X)$ $(1 \le i \le \ell)$ *are linearly independent.*

(ii) $T_X \mathcal{N}(h) + T_X S_{r,0}(n) = T_X S(n)$ *for* $X \in S_{r,0}(n) \bigcap \mathcal{N}(h)$.

Remark 2.8 (see [14]). Under LICQ condition 2.7, it is known that $\bar{X}$ is strongly stable if and only if there exist neighborhoods $U = B_\delta(\bar{X})$ of $\bar{X}$ in $S(n)$ and V of $(\bar{f},\bar{h})$ in $\mathcal{F}_U$ such that the natural projection $pr : \Xi \bigcap (U \times V) \to V$ is a homeomorphism. In fact, this is true for a more general condition that is called the Mangasarian–Fromovitz condition.

Definition 2.9. *Under LICQ condition* 2.7, *it is readily inferred that $\chi : \Omega \bigcap ((\rho^+)^{-1}(U) \times \boldsymbol{R}^\ell \times \mathcal{F}_U) \to \Xi \bigcap (U \times \mathcal{F}_U)$ is a homeomorphism for any subset $U \subset S(n)$. We refer to (Z,λ) as a strongly stable stationary point of* $\mathrm{Pro}(f,h)$ *if and only if Z^+ is a strongly stable stationary solution of* $\mathrm{Pro}(f,h)$.

We assume LICQ condition 2.7 throughout the remainder of this paper.

3. Map theoretic characterization of strong stability. This section reports investigation of strong stability for programs $\mathrm{Pro}(f,h)$. Under LICQ condition 2.7, we will characterize it from the point of view of one-to-one maps. We prepare two lemmas for the proof of Theorem 3.4. In this section we use the notation $V_\delta(f,h;U) = \{(f',h') \in \mathcal{F} : \|(f',h') - (f,h)\|_U < \delta\}$ for $\delta > 0$ and $U \subset S(n)$ and $(f,h) \in \mathcal{F}$.

Lemma 3.1. *Suppose that LICQ condition* 2.7 *holds. Let $\bar{X}^+ \in S_+(n)$ and U be a compact neighborhood of $\bar{X}$ in $S(n)$ and $(\bar{f},\bar{h}) \in \mathcal{F}$. Suppose $(\bar{X},\bar{\lambda}) \in S(n) \times \boldsymbol{R}^\ell$ is a unique stationary point of* $\mathrm{Pro}(\bar{f},\bar{h})$ *on $(\rho^+)^{-1}(U) \times \boldsymbol{R}^\ell$; then,* (*) *holds.*

(*) *For any $\epsilon > 0$, there exists $\delta > 0$ such that*

$$d(\psi((\rho^+)^{-1}(U) \times \boldsymbol{R}^\ell \backslash B_\epsilon((\bar{X},\bar{\lambda})); f,h), (O,\mathbf{0})) > \delta \text{ for any } (f,h) \in V_\delta(\bar{f},\bar{h};U).$$

Proof. Suppose that statement (*) does not hold, which implies that there exist $\epsilon > 0$ and a sequence $(f^{(k)}, h^{(k)}) \in \mathcal{F}$ such that

$$\begin{cases} \lim_{k\to\infty}(f^{(k)}, h^{(k)}) = (\bar{f}, \bar{h}) \in \mathcal{F}_U, \text{ and} \\ \lim_{k\to\infty} d(\psi((\rho^+)^{-1}(U) \times \boldsymbol{R}^\ell \setminus B_\epsilon((\bar{X}, \bar{\lambda})); f^{(k)}, h^{(k)}), (O, \mathbf{0})) = 0. \end{cases}$$

Therefore, there exists a sequence $(X^{(k)}, \lambda^{(k)}) \in (\rho^+)^{-1}(U) \times \boldsymbol{R}^\ell \setminus B_\epsilon((\bar{X}, \bar{\lambda})), (k = 1, 2, \ldots)$, such that

$$\lim_{k\to\infty} \psi(X^{(k)}, \lambda^{(k)}; f^{(k)}, h^{(k)}) = (O, \mathbf{0}).$$

We first consider the case that $\{(X^{(k)}, \lambda^{(k)}) : k = 1, 2, \ldots\}$ is bounded. We may assume that $\lim_{k\to\infty}(X^{(k)}, \lambda^{(k)})$ converges, so let $\lim_{k\to\infty}(X^{(k)}, \lambda^{(k)}) = (A, \lambda)$. Then we can readily show $(A, \lambda) \in (\rho^+)^{-1}(U) \times \boldsymbol{R}^\ell \setminus \mathrm{int}(B_\epsilon((\bar{X}, \bar{\lambda})))$ and $\psi(A, \lambda; \bar{f}, \bar{h}) = (O, \mathbf{0})$, which implies that (A, λ) is a stationary point of $\mathrm{Pro}(\bar{f}, \bar{h})$ and $\|(A, \lambda) - (\bar{X}, \bar{\lambda})\| \geq \epsilon$. Since $(\bar{X}, \bar{\lambda}) \in S(n) \times \boldsymbol{R}^\ell$ is a unique stationary point of $\mathrm{Pro}(\bar{f}, \bar{h})$ on $(\rho^+)^{-1}(U) \times \boldsymbol{R}^\ell$, we can deduce that $(A, \lambda) = (\bar{X}, \bar{\lambda})$. This contradicts $\|(A, \lambda) - (\bar{X}, \bar{\lambda})\| \geq \epsilon$.

Next, we consider the remaining case that $\{(X^{(k)}, \lambda^{(k)}) : k = 1, 2, \ldots\}$ is not bounded. In this remaining case one can derive a contradiction to LICQ condition 2.7 without difficulties. □

The following corollary follows directly from Definition 2.6 and Lemma 3.1.

COROLLARY 3.2. *Let $(\bar{X}, \bar{\lambda}) \in S(n)$ be a stationary point of* $\mathrm{Pro}(\bar{f}, \bar{h})$. *Under LICQ condition* 2.7, (i) *and* (ii) *are equivalent.*

(i) $(\bar{X}, \bar{\lambda})$ *is strongly stable.*

(ii) *There exist neighborhoods $U = B_\delta(\bar{X}^+)$ of $\bar{X}^+$ in $S(n)$ and V of $(\bar{f}, \bar{h})$ in $\mathcal{F}_U$ such that the natural projection $\pi : \Omega \bigcap ((\rho^+)^{-1}(U) \times \boldsymbol{R}^\ell \times V) \to V$ is bijective.*

Proof. (i)⇒(ii): Since $\bar{X}^+$ is strongly stable, it follows from Definition 2.6 that there exist real numbers $\delta > 0$ and $\alpha > 0$ such that $\mathrm{Pro}(f, h)$ has a unique stationary solution in $U = B_\delta(\bar{X}^+)$ for all $(f, h) \in V_\alpha(\bar{f}, \bar{h}; U)$. With the notice in Definition 2.9 in mind, this implies statement (ii).

(ii)⇒(i): Since $\pi : \Omega \bigcap ((\rho^+)^{-1}(U) \times \boldsymbol{R}^\ell \times V) \to V$ is bijective, we can represent $\pi^{-1}(f, h)$ as $\pi^{-1}(f, h) = (X(f, h), \lambda(f, h), f, h)$ for any $(f, h) \in V$. Apparently, $(X(f, h), \lambda(f, h))$ is a unique stationary point of $\mathrm{Pro}(f, h)$ on $(\rho^+)^{-1}(U) \times \boldsymbol{R}^\ell$ for $(f, h) \in V$. Therefore, from Lemma 3.1, for any ϵ with $0 < \epsilon \leq \delta$, there exists $\delta_\epsilon > 0$ such that

$$d(\psi((\rho^+)^{-1}(U) \times \boldsymbol{R}^\ell \setminus B_\epsilon((\bar{X}, \bar{\lambda})); f, h), (O, \mathbf{0})) > \delta_\epsilon \text{ for any } (f, h) \in V_{\delta_\epsilon}(\bar{f}, \bar{h}; U) \subset V,$$

which implies that $(X(f, h), \lambda(f, h)) \in B_\epsilon((\bar{X}, \bar{\lambda}))$ for any $(f, h) \in V_{\delta_\epsilon}$. This directly implies that $(\bar{X}, \bar{\lambda})$ is strongly stable. □

In proofs of the following lemma and theorem, we use the Brouwer's invariance theorem of domain (see [5]) which is now a standard method in deduction of equivalent conditions for strong stability as treated in, for example, [11], [8], and [17] since Kojima [9] successfully used it for the first time.

LEMMA 3.3. *Suppose that LICQ condition* 2.7 *holds. Let $(\bar{f}, \bar{h}) \in \mathcal{F}$ and $(\bar{X}, \bar{\lambda}) \in S(n) \times \boldsymbol{R}^\ell$ be a stationary point of* $\mathrm{Pro}(\bar{f}, \bar{h})$. *Let U be a neighborhood of $\bar{X}^+$ in $S(n)$*

and W be a neighborhood of $(\bar{X},\bar{\lambda})$ in $(\rho^+)^{-1}(U)\times\boldsymbol{R}^\ell$. Suppose $V=\{(f,h)\in\mathcal{F}:\psi(\cdot,\cdot;f,h)$ is one-to-one on $W\}$ is a neighborhood of $(\bar{f},\bar{h})$ in $\mathcal{F}_U$. Then, there exists $\delta>0$ such that, for any $(f,h)\in V_\delta(\bar{f},\bar{h};U)$, the following (i) *and* (ii) *hold.*

(i) $\mathrm{Pro}(f,h)$ *has a unique stationary point $(X(f,h),\lambda(f,h))$ in W.*

(ii) $(X(f,h),\lambda(f,h))\in B_\delta((\bar{X},\bar{\lambda}))\subset W$.

Proof. Since $\psi(\cdot,\cdot;\bar{f},\bar{h})$ is one-to-one on $W_0=\mathrm{int}(W)$, it follows from Brouwer's invariance theorem of domain that $\psi(W_0;\bar{f},\bar{h})$ is an open set in $S(n)\times\boldsymbol{R}^\ell$ which is homeomorphic to W_0. Therefore, there exists $\delta_0>0$ such that $K=B_{\delta_0}((\bar{X},\bar{\lambda}))\subset W_0$. It is clear that $\delta_0=d(\psi(Bd(K);\bar{f},\bar{h}),(O,\mathbf{0}))>0$. Define $r(f,h)=d(\psi(K;f,h),(O,\mathbf{0}))$ for $(f,h)\in V$. Since $\psi(X,\lambda;f,h)$ is continuous with respect to $(X,\lambda,f,h)\in(\rho^+)^{-1}(U)\times\boldsymbol{R}^\ell\times\mathcal{F}_U$ and K is compact, $r(f,h)$ is continuous with respect to $(f,h)\in V_{\delta_0}=V_{\delta_0}(f_0,h_0;U)$ in $\mathcal{F}_U$.

In fact, this can be proved as follows. Since K is compact, there exists $\alpha(f,h)\in K$ such that $\|\psi(\alpha(f,h);f,h)\|=r(f,h)$. It can readily be proved that there exists a constant $M>0$ such that $\|\psi(X,\lambda;f,h)-\psi(X,\lambda;f',h')\|\le M\|(f,h)-(f',h')\|$ for any $(X,\lambda)\in K$ and any $(f,h)\in V_{\delta_0}$. Therefore, $r(f',h')\le r(f,h)+M\|(f,h)-(f',h')\|$ for any $(f,h),(f',h')\in V_{\delta_0}$. Similarly, $r(f,h)\le r(f',h')+M\|(f,h)-(f',h')\|$ holds and we have proved $|r(f',h')-r(f,h)|\le M\|(f,h)-(f',h')\|$ for any $(f,h),(f',h')\in V_{\delta_0}$.

Define $r_0(f,h)=d(\psi(Bd(K);f,h),(O,\mathbf{0}))$ for $(f,h)\in V$. Similarly, $r_0(f,h)$ is continuous with respect to $(f,h)\in\mathcal{F}_U$. $r(\bar{f},\bar{h})=0$ follows from $\psi(\bar{X},\bar{\lambda};\bar{f},\bar{h})=(O,\mathbf{0})$ and $r_0(\bar{f},\bar{h})=\delta_0>0$. From the continuity of $r(\cdot,\cdot)$ and $r_0(\cdot,\cdot)$, there exists $\delta>0$ satisfying $0<\delta\le\delta_0$ such that $r(f,h)<r_0(f,h)$ holds $\forall(f,h)\in V_\delta=V_\delta(\bar{f},\bar{h};U)$. Therefore, $\alpha(f,h)\in\mathrm{int}(K)\subset W_0$ and $\psi(\alpha(f,h);f,h)\in\mathrm{int}(\psi(K;f,h))=\psi(\mathrm{int}(K);f,h))$ holds for any $(f,g)\in V_\delta$. Since it follows from $\psi(\alpha(f,h);f,h)\in\mathrm{int}(\psi(K;f,h))$ that $r(f,h)=0$, it is implied that $\psi(\alpha(f,h);f,h)=(O,\mathbf{0})$ for any $(f,h)\in V_\delta$. Therefore, $\alpha(f,h)$ is a unique stationary point of $\mathrm{Pro}(f,h)$ in K for any $(f,h)\in V_\delta$. Since $\psi(\cdot,\cdot;f,h)$ is one-to-one on W, $\alpha(f,h)$ is a unique stationary point of $\mathrm{Pro}(f,h)$ in W for any $(f,h)\in V_\delta$. □

The next theorem gives an equivalent condition for strong stability.

THEOREM 3.4. *Suppose that LICQ condition* 2.7 *holds. Let $(\bar{f},\bar{h})\in\mathcal{F}$ and $(\bar{X},\bar{\lambda})\in S(n)\times\boldsymbol{R}^\ell$ be a stationary point of* $\mathrm{Pro}(\bar{f},\bar{h})$. *Then* (i) *and* (ii) *are equivalent.*

(i) *$(\bar{X},\bar{\lambda})$ is strongly stable.*

(ii) *There exist neighborhoods $U=B_{\delta^*}(\bar{X}^+)$ of $\bar{X}^+$ in $S(n)$ and $W=B_\delta((\bar{X},\bar{\lambda}))$ of $(\bar{X},\bar{\lambda})$ with $W\subset(\rho^+)^{-1}(U)\times\boldsymbol{R}^\ell$ satisfying the following two conditions.*

(a) *$\bar{X}^+$ is a unique stationary solution in U for* $\mathrm{Pro}(\bar{f},\bar{h})$.

(b) *$V=\{(f,h)\in\mathcal{F}:\psi(\cdot,\cdot;f,h)$ is one-to-one on $W\}$ is a neighborhood of $(\bar{f},\bar{h})$ in $\mathcal{F}_U$.*

Proof. (i)⇒(ii): From Definition 2.9 of strong stability, there exist neighborhoods $U=B_{\delta^*}(\bar{X}^+)$ of $\bar{X}^+$ in $S(n)$ and V_0 of $(\bar{f},\bar{h})$ in $\mathcal{F}_U$ satisfying the natural projection $\pi:\Omega\bigcap((\rho^+)^{-1}(U)\times\boldsymbol{R}^\ell\times V_0)\to V_0$ is a homeomorphism. Therefore, $\mathrm{Pro}(f,h)$ has a unique stationary point on $(\rho^+)^{-1}(U)\times\boldsymbol{R}^\ell$ for any $(f,h)\in V_0$. It is clear that there exist $\delta_1>0$ and a neighborhood $V_1\subset V_0$ of $(\bar{f},\bar{h})$ in $\mathcal{F}_U$ satisfying $(\bar{f}(X^+)-A\bullet X^+,\bar{h}(X^+)-\mu)\in V_0$ for any $(A,\mu,f,h)\in B_{\delta_1}((O,\mathbf{0}))\times V_1$. Hence, $\psi(\cdot,\cdot;f(X^+)-A\bullet X^+,h(X^+)-\mu)$ has a unique stationary point on $(\rho^+)^{-1}(U)\times\boldsymbol{R}^\ell$ for any $(A,\mu,f,h)\in B_{\delta_1}((O,\mathbf{0}))\times V_1$. Therefore, there exists a unique $(X,\lambda)\in(\rho^+)^{-1}(U)\times\boldsymbol{R}^\ell$ such that $\psi(X,\lambda;f(X^+)-A\bullet X^+,h(X^+)-\mu)=(O,\mathbf{0})$. This implies that, for any $(A,\mu,f,h)\in B_{\delta_1}((O,\mathbf{0}))\times V_1$, there exists a unique $(X,\lambda)\in(\rho^+)^{-1}(U)\times\boldsymbol{R}^\ell$ such that $\psi(X,\lambda;f,h)=(A,\mu)$. Since $\psi(\cdot,\cdot;f,h)$ is continuous and bijective from $D(f,h)=\{(X,\lambda)\in(\rho^+)^{-1}(U)\times\boldsymbol{R}^\ell:\psi(X,\lambda;f,h)\in\operatorname{int}(B_{\delta_1}((O,\boldsymbol{0}))\}$ to $\operatorname{int}(B_{\delta_1}((O,\boldsymbol{0})))$, $D(f,h)$ is homeomorphic to $\operatorname{int}(B_{\delta_1}((O,\boldsymbol{0})))$ by Brouwer's invariance theorem of domain. Therefore, $K(f,h)=\{(X,\lambda)\in(\rho^+)^{-1}(U)\times\boldsymbol{R}^\ell:\psi(X,\lambda;f,h)\in B_{\frac{1}{2}\delta_1}((O,\boldsymbol{0}))\}$ is compact and homeomorphic to $B_{\frac{1}{2}\delta_1}((O,\boldsymbol{0}))$ and $Bd(K(f,h))=\{(X,\lambda)\in(\rho^+)^{-1}(U)\times\boldsymbol{R}^\ell:\|\psi(X,\lambda;f,h)\|=\frac{1}{2}\delta_1\}$. We will prove that there exists a neighborhood W of $(\bar{X},\bar{\lambda})$ in $S(n)\times\boldsymbol{R}^\ell$ such that $W\subset\bigcap\{K(f,h):(f,h)\in V_1\}$ in what follows. In fact, suppose the contrary, i.e., there exists a sequence $(X^{(k)},\lambda^{(k)},f^{(k)},h^{(k)})$ satisfying $\lim_{k\to\infty}(X^{(k)},\lambda^{(k)},f^{(k)},h^{(k)})=(\bar{X},\bar{\lambda},\bar{f},\bar{h})$ and $\|\psi(X^{(k)},\lambda^{(k)};f^{(k)},h^{(k)})\|>\frac{1}{2}\delta_1$. Taking the limit $\lim_{k\to\infty}\|\psi(X^{(k)},\lambda^{(k)};f^{(k)},h^{(k)})\|>\frac{1}{2}\delta_1$, make $\|\psi(\bar{X},\bar{\lambda};\bar{f},\bar{h})\|\geq\frac{1}{2}\delta_1$, which leads to a contradiction, since $\lim_{k\to\infty}\psi(X^{(k)},\lambda^{(k)};f^{(k)},h^{(k)})=\psi(\bar{X},\bar{\lambda};\bar{f},\bar{h})$.

(ii)⇒(i): Since $V=\{(f,h)\in\mathcal{F}:\psi(\cdot,\cdot;f,h)\text{ is one-to-one on }W\}$ is a neighborhood of $(\bar{f},\bar{h})$ in $\mathcal{F}_U$, it follows from Lemma 3.3 that there exists $\delta_1>0$ such that

$$\begin{cases}\text{Pro}(f,h)\text{ has a unique stationary point }(X(f,h),\lambda(f,h))\text{ in }W=B_\delta((\bar{X},\bar{\lambda}))\\ \qquad\text{for any }(f,h)\in V_{\delta_1}(\bar{f},\bar{h};U).\end{cases}$$

From Lemma 3.1, there exists $\delta_2>0$ such that

$$\begin{cases}d(\psi((\rho^+)^{-1}(U)\times\boldsymbol{R}^\ell\setminus B_\delta((\bar{X},\bar{\lambda}));f,h),(O,\boldsymbol{0}))>\delta_2\\ \qquad\text{for any }(f,h)\in V_{\delta_2}(\bar{f},\bar{h};U).\end{cases}$$

Therefore, $\text{Pro}(\bar{f},\bar{h})$ has no stationary point in $(\rho^+)^{-1}(U)\times\boldsymbol{R}^\ell\setminus B_\delta((\bar{X},\bar{\lambda}))=(\rho^+)^{-1}(U)\times\boldsymbol{R}^\ell\setminus W$ for any $(f,h)\in V_{\delta_2}(\bar{f},\bar{h};U)$. Let $\delta_3=\min\{\delta_1,\delta_2\}$. Then $\text{Pro}(\bar{f},\bar{h})$ has a unique stationary point in $(\rho^+)^{-1}(U)\times\boldsymbol{R}^\ell$ for any $(f,h)\in V_{\delta_3}(\bar{f},\bar{h};U)$, which implies that $(\bar{X},\bar{\lambda})$ is strongly stable from Corollary 3.2. □

Remark 3.5. Since all proofs in this paper to deduce Theorem 3.4 are applicable to those programs satisfying that $\chi:\Omega\to\Xi$ is bijective, Theorem 3.4 holds for such programs.

4. Generalized Jacobian $\partial_X\rho^+(\bar{X})$ and $\partial_X\rho^-(\bar{X})$ and their eigenvalues.

In this section we prove that all eigenvalues of $C_+\in\partial_X\rho^+(\bar{X})$ and $C_-\in\partial_X\rho^-(\bar{X})$ lie in the interval between 0 and 1 on the real line, and we deduce an inclusion stated in Lemma 4.15. We need both these facts to construct an analogous method for NSDP to that of Kojima's for NLP. The next fact is well known (see [18]).

FACT 4.1. *Define* $\|x\|=\sqrt{\sum_{i=1}^n|x_i|^2}$ *for* $x=(x_1,\dots,x_n)\in\boldsymbol{R}^n$. *Let* C *be a convex closed subset in* $\boldsymbol{R}^n$. *In that case,* (i) *and* (ii) *hold.*

(i) *There exists a unique* $a\in C$, *which is denoted by* $a(x)$, *satisfying* $\|x-a\|=\inf\{\|x-a\|:a\in C\}$.

(ii) $\|a(x)-a(y)\|\leq\|x-y\|$ *holds for any* $x,y\in\boldsymbol{R}^n$.

DEFINITION 4.2. *Let* V_1 *and* V_2 *be normed vector spaces with their norms denoted by* $\|\cdot\|$ *and* U *be an open subset of* V_1. *Then, a mapping* $f:U\to V_2$ *is called Lipschitz continuous with its modulus* M *if there exists a constant* M *such that* $\|f(x)-f(y)\|\leq M\|x-y\|$ *for any* x, $y\in U$.

Fact 4.1 shows that $\rho(X)=(\rho^+(X),\rho^-(X))$ is Lipschitz continuous. Before we state the next definition, we remark that any Lipschitz continuous map is differentiable almost everywhere in the sense of Lebesgue measure by Rademacher's theorem (see [3]).

DEFINITION 4.3 (see [2], [6]). *Let* U *be an open set of* $\boldsymbol{R}^n$ *and* f *be a Lipschitz continuous map from* U *to* $\boldsymbol{R}^m$. *Let* S *be any set of Lebesgue measure* 0 *in* $\boldsymbol{R}^n$ *and*

E_f be the set of all points $x \in U$ for which the Jacobian $D_x f$ exists. Then, for $\bar{x} \in U$, the generalized Jacobian $\partial_x f(\bar{x})$ of f at $\bar{x}$ is defined by

$$\partial_x f(\bar{x}) = conv\Big\{\lim_{k\to\infty} D_x f(x_k) : x_k \in E_f \setminus S, (k = 1, 2, \ldots),\ such\ that\ \lim_{k\to\infty} x_k = \bar{x}\Big\}.$$

It is known that this definition of the generalized Jacobian is independent of choice of S. In case $m = n$, f is called nonsingular at $\bar{x}$ if $\operatorname{rank} A = n$ *for any $A \in \partial_x f(\bar{x})$, and f is called singular at $\bar{x}$ if f is not nonsingular at $\bar{x}$.*

Definition 4.4. *Let A be an $n \times n$ real matrix whose eigenvalues are all real. We denote the number of positive* (*zero, negative*) *eigenvalues of A by posi(A)* (*resp., zero(A), nega(A)*). *We define Type(A) = (posi(A), zero(A), nega(A)).*

The proof of the next lemma is elementary, so we omit it.

Lemma 4.5. *Let U be an open set of $\boldsymbol{R}^n$ and f be a Lipschitz continuous map from U to $\boldsymbol{R}^n$ and $\bar{x} \in U$. Then, the following* (i) *and* (ii) *are equivalent.*

(i) *f is nonsingular at $\bar{x}$.*

(ii) *sgn* $\det A$ *is nonzero and constant for $A \in \partial_x f(\bar{x})$.*

Moreover, in the case that all eigenvalues of A are real for all $A \in \partial_x f(\bar{x})$, the above (i), (ii), *and the following* (iii) *are equivalent.*

(iii) *Type(A) = (posi(A), zero(A), nega(A)) is constant and zero$(A) = 0$ for $A \in \partial_x f(\bar{x})$.*

Remark 4.6. It is clear that $S_{r,n-r}(n)$ is an open subset of $S(n)$ and $S^*(n) = \bigcup_{r=0}^n S_{r,n-r}(n)$ is a disjoint union and $\rho^+(S_{r,n-r}(n)) = S_{r,0}(n)$. Since $S_{r,0}(n)$ is an analytic submanifold of $S(n)$ and $\rho^+|S_{r,n-r}(n) : S_{r,n-r}(n) \to S_{r,0}(n)$ is the orthogonal projection, ρ^+ is analytic on $S_{r,n-r}(n)$ and therefore ρ^+ is analytic on $S^*(n)$. Similarly, ρ^- is also analytic on $S^*(n)$.

Definition 4.7. *We define the following notations for Lemma* 4.8*:*

$$J^+(\alpha,\beta) = \begin{cases} 1, & (\alpha = \beta > 0), \\ \frac{\alpha^+ - \beta^+}{\alpha - \beta}, & (\alpha \neq \beta) \\ 0, & (\alpha = \beta < 0), \end{cases} \quad and\ J^-(\alpha,\beta) = \begin{cases} 0, & (\alpha = \beta > 0), \\ \frac{\alpha^- - \beta^-}{\alpha - \beta}, & (\alpha \neq \beta), \\ 1, & (\alpha = \beta < 0). \end{cases}$$

We have $D_X \rho^+(X)$ of more explicit form in the next lemma. It shows that $D_X \rho^+(X)$ and $D_X \rho^-(X)$ are nonnegative diagonal tensors at their differentiable points.

Lemma 4.8. *Let $X \in S^*(n)$ and represent $X = P\Gamma P^T$, where $P \in O(n)$ and $\Gamma = \operatorname{Diag}(\gamma_1, \ldots, \gamma_n) \in D^*(n)$. Then*

$$\begin{aligned}(1)\ D_X \rho^+(X) = {} & \textstyle\sum_{k:\gamma_k>0} PE_{kk}P^T \otimes PE_{kk}P^T \\ & + \textstyle\sum_{1\le i<j\le n} J^+(\gamma_i,\gamma_j) P\left(\dfrac{E_{ij}+E_{ji}}{\sqrt{2}}\right)P^T \otimes P\left(\dfrac{E_{ij}+E_{ji}}{\sqrt{2}}\right)P^T.\end{aligned}$$

$$\begin{aligned}(2)\ D_X \rho^-(X) = {} & \textstyle\sum_{k:\gamma_k<0} PE_{kk}P^T \otimes PE_{kk}P^T \\ & + \textstyle\sum_{1\le i<j\le n} J^-(\gamma_i,\gamma_j) P\left(\dfrac{E_{ij}+E_{ji}}{\sqrt{2}}\right)P^T \otimes P\left(\dfrac{E_{ij}+E_{ji}}{\sqrt{2}}\right)P^T.\end{aligned}$$

Moreover, $D_X \rho^+(X) + D_X \rho^-(X) = I_{S(n)}$ holds.

Proof. Before addressing the proof, we remark that $E_{ij}E_{pq} = \begin{cases} E_{iq}, & (j = p), \\ O, & (j \neq p). \end{cases}$ Let $X = P\Gamma P^T$ with $P \in O(n)$ and $\Gamma = \operatorname{Diag}(\gamma_1, \ldots, \gamma_n) \in D(n)$. For $T = (t_{ij}) = \sum_{i=1}^n \sum_{j=1}^n t_{ij}E_{ij} \in S(n)$, define

$$\begin{cases} \langle T\rangle = \sum_{1\le i<j\le n} t_{ij}(E_{ij}-E_{ji}), \\ [T] = \sum_{k=1}^{n} t_{kk}E_{kk}, \\ \Theta(T) = Pe^{-\langle T\rangle}(\Gamma+[T])e^{\langle T\rangle}P^T. \end{cases}$$

By a simple calculation, we have

$$\begin{cases} D_{t_{ij}}\Theta(O) = P(-(E_{ij}-E_{ji})\Gamma+\Gamma(E_{ij}-E_{ji}))P^T \\ \qquad\qquad = (\gamma_i-\gamma_j)P(E_{ij}+E_{ji})P^T,\ (1\le i<j\le n), \\ D_{t_{kk}}\Theta(O) = PE_{kk}P^T,\ (k=1,2,\dots,n). \end{cases}$$

Therefore, since $\left\{E_{kk}, \frac{E_{ij}+E_{ji}}{\sqrt{2}} : 1\le k\le n,\ 1\le i<j\le n\right\}$ is an orthonormal basis of $S(n)$, we have

$$D_{\scriptscriptstyle T}\Theta(O) = \sum_{k=1}^{n} PE_{kk}P^T\otimes E_{kk} + \sum_{1\le i<j\le n}(\gamma_i-\gamma_j)P\left(\frac{E_{ij}+E_{ji}}{\sqrt{2}}\right)P^T\otimes\left(\frac{E_{ij}+E_{ji}}{\sqrt{2}}\right).$$

This equation shows that $D_{\scriptscriptstyle T}\Theta(O)$ is a diagonal tensor in $S(n)\otimes S(n)$; also, when $\gamma_i\ne\gamma_j, (\forall i\ne\forall j)$, $D_{\scriptscriptstyle T}\Theta(O)$ is nonsingular. In fact,

$$\begin{aligned}(D_{\scriptscriptstyle T}(O))^{-1} = &\sum_{k=1}^{n} E_{kk}\otimes PE_{kk}P^T + \sum_{1\le i<j\le n}(\gamma_i-\gamma_j)^{-1} \\ &\times\left(\frac{E_{ij}+E_{ji}}{\sqrt{2}}\right)\otimes P\left(\frac{E_{ij}+E_{ji}}{\sqrt{2}}\right)P^T.\end{aligned}$$

Therefore, $\Theta : S(n)\ni T\mapsto\Theta(T)\in S(n)$ is a diffeomorphism from a neighborhood of O to a neighborhood of X in the case $\gamma_i\ne\gamma_j, (\forall i\ne\forall j)$. Since $\langle T\rangle$ is skew symmetric, $e^{\langle T\rangle}$ is an orthogonal matrix. Define $\Theta^+(T)$ as the positive semidefinite part of $\Theta(T)$; it is calculated explicitly as

$$\Theta^+(T) = \Theta(T)^+ = (Pe^{-\langle T\rangle}(\Gamma+[T])e^{\langle T\rangle}P^T)^+ = Pe^{-\langle T\rangle}(\Gamma+[T])^+e^{\langle T\rangle}P^T.$$

When X is nonsingular, i.e., $\gamma_k\ne 0, (\forall k)$, it is readily shown that there exists a $\delta>0$ such that $(\Gamma+[T])^+ = \sum_{k:\gamma_k>0}(\gamma_{kk}+t_{kk})E_{kk}$ for any T with $\|T\|<\delta$. From this equation, we can easily show that $(\Gamma+[T])^+$ is differentiable with respect to T at O and that its derivative is

$$D_{t_{kk}}(\Gamma+[T])^+ = \begin{cases} E_{kk}, & (\gamma_k>0), \\ O, & (\gamma_k<0). \end{cases}$$

Therefore, when X is nonsingular, $\Theta^+(T)$ is differentiable with respect to T at O, and

$$\begin{aligned}D_{\scriptscriptstyle T}\Theta^+(\mathbf{0}) = &\sum_{k:\gamma_k>0} PE_{kk}P^T\otimes E_{kk} + \sum_{1\le i<j\le n}(\gamma_i^+-\gamma_j^+) \\ &\times P\left(\frac{E_{ij}+E_{ji}}{\sqrt{2}}\right)\ P^T\otimes\left(\frac{E_{ij}+E_{ji}}{\sqrt{2}}\right).\end{aligned}$$

From these equations above, we can deduce that $\rho^+(X)$ is differentiable in the case $\gamma_i\ne\gamma_j, (\forall i\ne\forall j)$ and that its derivative is calculated as

$$\begin{aligned} D_X\rho^+(X) &= D_T\Theta^+(O)(D_T\Theta(O))^{-1} \\ &= \sum_{k:\gamma_k>0} PE_{kk}P^T \otimes PE_{kk}P^T \\ &\quad + \sum_{1\le i<j\le n} \frac{\gamma_i^+ - \gamma_j^+}{\gamma_i - \gamma_j} P\left(\frac{E_{ij}+E_{ji}}{\sqrt{2}}\right)P^T \otimes P\left(\frac{E_{ij}+E_{ji}}{\sqrt{2}}\right)P^T. \end{aligned}$$

From Remark 4.6, we can directly deduce the equation for $D_X\rho^+(X)$ in this lemma.

Similarly, we can deduce the equation for $D_X\rho^-(X)$. The relation $D_X\rho^+(X) + D_X\rho^-(X) = I_{S(n)}$ holds clearly since $\rho^+(X) + \rho^-(X) = X$. □

DEFINITION 4.9. *Let $X \in S(n)$. Then, we define $\mathcal{G}(X) = \{P \in O(n) : X = PXP^T\}$.*

We omit proof of the next lemma since it is rudimentary.

LEMMA 4.10. *Let $\alpha,\ \beta \in \boldsymbol{R}$. Define $J(\alpha,\beta) = \{(\lim_{k\to\infty} \frac{\alpha_k^+ - \beta_k^+}{\alpha_k - \beta_k}, \lim_{k\to\infty} \frac{\alpha_k^- - \beta_k^-}{\alpha_k - \beta_k}) \in \boldsymbol{R}^2 : \alpha_k, \beta_k, (k = 1,2,\ldots)$ with $\alpha_k \neq \beta_k, (k = 1,2,\ldots)$ and $\lim_{k\to\infty}\alpha_k = \alpha, \lim_{k\to\infty} \beta_k = \beta\}$. Then $J(\alpha,\beta) = \begin{cases} \{(s,t) : s \ge 0, t \ge 0, s+t = 1\}, & (\alpha = \beta = 0), \\ \{(J^+(\alpha,\beta), J^-(\alpha,\beta))\}, & (otherwise). \end{cases}$*

Since the next lemma follows readily from Lemma 4.8 and Definition 4.3, we omit its proof.

LEMMA 4.11. *$\bar{X} = \bar{P}\mathrm{Diag}(\bar{\gamma}_1, \ldots, \bar{\gamma}_n)\bar{P}^T \in S_{r,s}(n)$ with $\bar{P} \in O(n)$ and $\bar{\gamma}_1 \ge \cdots \ge \bar{\gamma}_n$. Let $X^{(k)} = P^{(k)}\Gamma^{(k)}P^{(k)T} \in S_{d,n-d}(n), (k = 1,2,\ldots)$, where $P^{(k)} \in O(n)$ and $\Gamma^{(k)} = \mathrm{Diag}(\gamma_1^{(k)}, \ldots, \gamma_n^{(k)})$ with $\gamma_1^{(k)} > \cdots > \gamma_n^{(k)}$, i.e., $\gamma_1^{(k)} > \cdots > \gamma_d^{(k)} > 0 > \gamma_{d+1}^{(k)} > \cdots > \gamma_n^{(k)}$. Suppose that*

$$\begin{cases} \lim_{k\to\infty} X^{(k)} & = \bar{X}, \\ \lim_{k\to\infty} P^{(k)} & = Q, \\ \lim_{k\to\infty} D_X\rho^+(X^{(k)}) & = C_+, \\ \lim_{k\to\infty} D_X\rho^-(X^{(k)}) & = C_-. \end{cases}$$

Then

$$\begin{cases} Q\bar{P}^T \in \mathcal{G}(\bar{X}), \\ C_+ \quad = \sum_{k=1}^n s_{kk} QE_{kk}Q^T \otimes QE_{kk}Q^T \\ \qquad\quad + \sum_{1\le i<j\le n} s_{ij} Q\left(\frac{E_{ij}+E_{ji}}{\sqrt{2}}\right)Q^T \otimes Q\left(\frac{E_{ij}+E_{ji}}{\sqrt{2}}\right)Q^T, \\ C_- \quad = \sum_{k=1}^n t_{kk} QE_{kk}Q^T \otimes QE_{kk}Q^T \\ \qquad\quad + \sum_{1\le i<j\le n} t_{ij} Q\left(\frac{E_{ij}+E_{ji}}{\sqrt{2}}\right)Q^T \otimes Q\left(\frac{E_{ij}+E_{ji}}{\sqrt{2}}\right)Q^T, \end{cases}$$

where

$$\begin{cases} (s_{kk}, t_{kk}) = (1,0),\ (1 \le \forall i \le d), \\ (s_{kk}, t_{kk}) = (0,1),\ (d+1 \le \forall i \le n), \\ (s_{ij}, t_{ij}) \ \ \in J(\bar{\gamma}_i, \bar{\gamma}_j),\ (\forall i, \forall j : i \neq j). \end{cases}$$

DEFINITION 4.12. *Let $A \in S(n)$. Then we denote by $V(A; > 0)$ (resp., $V(A; \ge 0)$, $V(A; < 0)$, $V(A; \le 0)$, $V(A; = 0)$) the space spanned by the eigenvectors of A whose eigenvalues are positive (resp., nonnegative, negative, nonpositive, zero). By inspection, $\boldsymbol{R}^n = V(A; > 0) \oplus V(A; = 0) \oplus V(A; < 0)$ holds. We also use these conventions for symmetric real tensors, for example, C_+ and C_- of $C \in \partial_X\rho(\bar{X})$.*

The next proposition is proved directly, and immediately from it we can deduce that all eigenvalues of $C_+ \in \partial_X \rho^+(\bar{X})$ and $C_- \in \partial_X \rho^-(\bar{X})$ lie in the interval between 0 and 1 on the real line.

PROPOSITION 4.13. *Let $\bar{X} = \bar{P}\mathrm{Diag}(\bar{\gamma}_1, \ldots, \bar{\gamma}_n)\bar{P}^T$ with $\bar{P} \in O(n)$ and $\bar{\gamma} = (\bar{\gamma}_1, \ldots, \bar{\gamma}_n)$. Then, the following statements hold.*

(i) *C_+ and C_- are positive semidefinite symmetric real tensors of $End_{\boldsymbol{R}}(S(n)) = S(n) \otimes S(n)$ for any $C = (C_+, C_-) \in \partial_X \rho(\bar{X})$.*

(ii) *$S(n) = V(C_+; > 0) \oplus V(C_+; = 0)$ holds for any $C = (C_+, C_-) \in \partial_X \rho(\bar{X})$.*

(iii) *$C_+ + C_- = I_{S(n)}$ holds for any $C \in \partial_X \rho(\bar{X})$.*

(iv) *C_+ and C_- are commutative, i.e., $C_+C_- = C_-C_+$ for any $C = (C_+, C_-) \in \partial_X \rho(\bar{X})$.*

(v) *C_+ and C_- are simultaneously diagonalized for any $C = (C_+, C_-) \in \partial_X \rho(\bar{X})$.*

Proof. Parts (i), (ii), and (iii) are inferred without difficulties. We will prove (iv) first. Since $C_+ + C_- = I_{S(n)}$, C_- is a polynomial of C_+, which implies that C_+ and C_- commute to each other, i.e., that $C_+C_- = C_-C_+$ holds. In fact, $C_+C_- = C_+(I_{S(n)} - C_+) = (I_{S(n)} - C_+)C_+ = C_-C_+$. Since C_+ and C_- are commutative, (v) follows immediately. □

Remark 4.14. Through the remainder of this paper, we often identify an element of $S(n) \otimes S(n)$ with one of $End_{\boldsymbol{R}}(S(n))$ by the canonical isomorphism from $S(n) \otimes S(n)$ to $End_{\boldsymbol{R}}(S(n))$, $A \otimes B \mapsto (X \mapsto A(B \bullet X))$. Hence, we often consider $C_+, C_- \in End_{\boldsymbol{R}}(S(n))$. From this viewpoint, the commutativity of C_+ and C_- implies that any eigenspace of C_+ is an invariant space of both C_+ and C_-. For example, $C_+(V(C_+; > 0)) = V(C_+; > 0)$, $C_+(V(C_+; 0)) = V(C_+; = 0)$, $C_-(V(C_+; > 0)) = V(C_+; > 0)$, $C_-(V(C_+; = 0)) = V(C_+; = 0)$.

As stated in section 2, we consider $T_{\bar{X}}S(n) = T_{\bar{X}^+}S(n) = S(n)$ canonically in the next lemma. The inclusion $T_{\bar{X}^+}S_{r,0}(n) \subset V(C_+; > 0)$ of this lemma takes an important role in deducing part (i) of Lemma 5.6.

LEMMA 4.15. *Let $\bar{X} \in S(n)$ and $\bar{X}^+ \in S_{r,0}(n)$. Then, $V(C_+; > 0) \supset T_{\bar{X}^+}S_{r,0}(n)$ for any $C = (C_+, C_-) \in \partial_X \rho(\bar{X})$.*

Proof. Suppose that $\bar{X} = \bar{P}\,\mathrm{Diag}(\bar{\gamma}_1, \ldots, \bar{\gamma}_n)\bar{P}^T$ with $\bar{\gamma}_1 = \cdots = \bar{\gamma}_{k_1} > \bar{\gamma}_{k_1+1} = \cdots = \bar{\gamma}_{k_2} > \bar{\gamma}_{k_2+1} = \cdots, \text{etc.}, \cdots = \gamma_{k_{\nu-1}} > \bar{\gamma}_{k_{\nu-1}+1} (= 0) = \cdots = \bar{\gamma}_{k_\nu} > \bar{\gamma}_{k_\nu+1} = \cdots, \text{etc.}, \cdots = \bar{\gamma}_{k_{s-1}} > \bar{\gamma}_{k_{s-1}+1} = \cdots = \bar{\gamma}_n$. Then

$$\begin{aligned}\mathcal{G}(\bar{X}) &= \bar{P}(O(k_1) \times O(k_2 - k_1) \times \cdots \times O(k_{s-1} - k_{s-2}) \times O(n - k_{s-1}))\bar{P}^T \\ &\subset \bar{P}(O(r) \times O(n - r))\bar{P}^T.\end{aligned}$$

Since

$$T_{\bar{X}^+}S_{r,0}(n) = \bar{P}\left\{\begin{pmatrix}\dot{Y}_{11} & \dot{Y}_{21}^T \\ \dot{Y}_{21} & O\end{pmatrix} : \dot{Y}_{11} \in S(r), \dot{Y}_{21} \in \mathcal{M}(n - r, r)\right\}\bar{P}^T$$

from Fact 2.3, we can infer

$$G(T_{\bar{X}^+}S_{r,0}(n))G^T = T_{\bar{X}^+}S_{r,0}(n) \text{ for all } G \in \mathcal{G}(\bar{X}) \subset \bar{P}(O(r) \times O(n - r))\bar{P}^T.$$

Suppose that $\lim_{k\to\infty} X_k = \bar{X}$ and $C_+ = \lim_{k\to\infty} D_X\rho^+(X_k) \in \partial_X \rho^+(\bar{X})$. Then Lemma 4.11 shows that

$$\begin{aligned}C_+ = &\sum_{k=1}^{n} s_{kk} Q E_{kk} Q^T \otimes Q E_{kk} Q^T \\ &+ \sum_{1 \le i < j \le n} s_{ij} Q \left(\frac{E_{ij} + E_{ji}}{\sqrt{2}}\right) Q^T \otimes Q \left(\frac{E_{ij} + E_{ji}}{\sqrt{2}}\right) Q^T,\end{aligned}$$

with

$$QP^T \quad \in \quad \mathcal{G}(\bar{X}) \qquad \text{and} \qquad \begin{cases} s_{kk} > 0, & (\bar{\gamma}_k > 0), \\ s_{ij} > 0, & (i \neq j \text{ and } (\bar{\gamma}_i > 0 \textit{ or } \bar{\gamma}_j > 0)). \end{cases}$$

Therefore,

$$\begin{aligned} V(C_+; > 0) &\supset Q \left\{ \begin{pmatrix} \dot{Y}_{11} & \dot{Y}_{21}^T \\ \dot{Y}_{21} & O \end{pmatrix} : \dot{Y}_{11} \in S(r), \dot{Y}_{21} \in \mathcal{M}(n-r, r) \right\} Q^T \\ &= \bar{P} \left\{ \begin{pmatrix} \dot{Y}_{11} & \dot{Y}_{21}^T \\ \dot{Y}_{21} & O \end{pmatrix} : \dot{Y}_{11} \in S(r), \dot{Y}_{21} \in \mathcal{M}(n-r, r) \right\} \bar{P}^T \\ &= \ T_{\bar{X}^+} S_{r,0}(n). \end{aligned}$$

Since $\partial_X \rho^+(X)$ is the convex hull of the set of such C_+'s of above type, Lemma 4.15 follows. □

5. Construction of an analogue of Kojima's method and the essential difference between NSDP and NLP. In this section we construct an analogue for NSDP to Kojima's method for NLP and we make clear the essential difference between NSDP and NLP. For those NSDP that do not have this difference, we can deduce an algebraic condition equivalent to strong stability for them under LICQ condition 2.7 exactly similar to classical cases of [9].

DEFINITION 5.1. *Let $C = (C_+, C_-) \in \partial\rho(X)$. Define $C_{++} : V(C_+; > 0) \to V(C_+; > 0)$ to be the restriction C_+ to $V(C_+; > 0)$. $C_{+-} : V(C_+; = 0) \to V(C_+; = 0)$ to be the restriction C_+ to $V(C_+; = 0)$. Define $C_{-+} : V(C_+; > 0) \to V(C_+; > 0)$ to be the restriction C_- to $V(C_+; > 0)$. Define $C_{--} : V(C_+; = 0) \to V(C_+; = 0)$ to be the restriction C_- to $V(C_+; = 0)$.*

Remark 5.2.

(i) It follows from $C_{+-} = O$ that $C_{--} = I_{V(C_+; > 0)}$. Then, from Remark 4.14, we can write C_+ and C_- as

$$\begin{cases} C_+ = \begin{pmatrix} C_{++} & O \\ O & C_{+-} \end{pmatrix} = \begin{pmatrix} C_{++} & O \\ O & O \end{pmatrix}, \\ C_- = \begin{pmatrix} C_{-+} & O \\ O & C_{--} \end{pmatrix} = \begin{pmatrix} C_{-+} & O \\ O & I \end{pmatrix} \end{cases}$$

and we have

$$\begin{cases} D_X^2 L(X^+, \lambda; f, h) = \begin{pmatrix} D_{X_1}^2 L(X^+, \lambda; f, h) & D_{X_1} D_{X_2} L(X^+, \lambda; f, h) \\ D_{X_2} D_{X_1} L(X^+, \lambda; f, h) & D_{X_2}^2 L(X^+, \lambda; f, h) \end{pmatrix}, \\ D_X^2 L(X^+, \lambda; f, h) C_+ + C_- = \begin{pmatrix} D_{X_1}^2 L(X^+, \lambda; f, h) C_{++} + C_{-+} & O & (D_{X_1} h(X^+))^T \\ D_{X_2} D_{X_1} L(X^+, \lambda; f, h) C_{++} & I & (D_{X_2} h(X^+))^T \end{pmatrix}, \\ D_X h(X^+) C_+ = \begin{pmatrix} D_{X_1} h(X^+) C_{++} & O \end{pmatrix}. \end{cases}$$

(ii) By chain rule of generalized Jacobian (see [8]), we have

$$\partial_{(X,\lambda)} \psi(X, \lambda; f, h) = \left\{ \begin{pmatrix} D_X^2 L(X^+, \lambda; f, h) C_+ + C_- & (D_X h(X^+))^T \\ D_X h(X^+) C_+ & O \end{pmatrix} : C \in \partial_X \rho(X) \right\}.$$

LEMMA 5.3. *The following* (i), (ii), *and* (iii) *hold.*

(i) *Let $A \in S(n)$ and $A = (a_1, a_2, \dots, a_n)$ be the representation by column vectors $a_1, a_2, \dots, a_n$ of A. Let $t_1, \dots, t_n \geq 0$ and $t = (t_1, t_2, \dots, t_n)$ and $A(t) = (t_1 a_1, t_2 a_2, \dots, t_n a_n)$. Then all eigenvalues of $A(t)$ are real. Moreover, if $t_1, \dots, t_n > 0$, then $Type(A(t)) = Type(A)$.*

(ii) *Let $A \in S(n)$ and $B \in S_+(n)$. Then all eigenvalues of AB are real. Moreover, if $B \in S_{++}(n)$, then $Type(A) = Type(AB)$.*

(iii) *Let $A, C \in S(n)$ and $B \in S_+(n)$ with $B + C = I$. Then all eigenvalues of $AB + C$ are real.*

Proof. (i): Let $T = \text{Diag}(t) = \text{Diag}(t_1, \dots, t_n)$. Then $A(t) = AT$ holds. Let $S = \sqrt{T} = \text{Diag}(\sqrt{t_1}, \sqrt{t_2}, \dots, \sqrt{t_n})$. Suppose that $t_1, \dots, t_n > 0$. Then S is nonsingular and $A(t) = S^{-1}(SAS^T)S$. Since SAS^T is a symmetric real matrix and since $A(t) = S^{-1}(SAS^T)S$ and SAS^T are similar, their characteristic polynomials are consistent; therefore, their eigenvalues are all real and $Type(A(t)) = Type(SAS^T) = Type(A)$.

In case $t_1, \dots, t_n \geq 0$, let $\epsilon > 0$ and $t_\epsilon = (t_1 + \epsilon, \dots, t_n + \epsilon)$ and $T_\epsilon = \text{Diag}(t_\epsilon) = \text{Diag}(t_1 + \epsilon, \dots, t_n + \epsilon)$ and $A(t_\epsilon) = AT_\epsilon$. It follows from the former result that all eigenvalues of $A(t_\epsilon)$ are real. Therefore, all eigenvalues of $A(t)$ are real since $A(t) = \lim_{\epsilon \to +0} A(t_\epsilon)$.

(ii): We represent B as $b = P\Gamma P^T$ with $P \in O(n)$ and $\Gamma = \text{Diag}(\gamma_1, \dots, \gamma_n) \in D(n)$. From $B \in S_+(n)$, it follows that $\gamma_i \geq 0, (\forall i)$, holds. Since $\det(tI - (AB)) = \det(tI - (P^T AP)\Gamma)$, part (ii) follows from part (i) of this lemma.

(iii): By part (ii), $(A - I)B$ has real eigenvalues, and hence, so does $B + C = (A - I)B + I$. □

The next proposition indicates that ψ has an advantageous property that Kojima function does not have.

Proposition 5.4. *All eigenvalues of A are real for any $A \in \partial_{(X,\lambda)}\psi(X, \lambda; f, h)$.*

Proof. We can represent A as

$$A = \begin{pmatrix} D_X^2 L(X^+, \lambda; f, h) & (D_X h(X^+))^T \\ D_X h(X^+) & O \end{pmatrix} \begin{pmatrix} C_+ & O \\ O & I \end{pmatrix} + \begin{pmatrix} C_- & O \\ O & O \end{pmatrix}.$$

Therefore, this proposition follows from Proposition 4.13 and Lemma 5.3. □

For the remainder of this paper we treat the case that $C = (C_+, C_-) \in \partial\rho(\bar{X})$ and $(\bar{X}, \bar{\lambda}) \in S(n) \times \boldsymbol{R}^\ell$ is a stationary point of $\text{Pro}(\bar{f}, \bar{h})$.

Definition 5.5. *Let $W_1(C_+, \bar{h}) = V(C_+; > 0) \bigcap T_{\bar{X}^+}\mathcal{N}(\bar{h})$ and $W_2(C_+, \bar{h})$ be the vector subspace of $V(C_+; > 0)$ such that $V(C_+; > 0) = W_1(C_+, \bar{h}) \oplus W_2(C_+, \bar{h})$ and $W_1(C_+, \bar{h}) \perp W_2(C_+, \bar{h})$, i.e., $w_1 \bullet w_2 = 0, (\forall w_1 \in W_1(C_+, \bar{h}), \forall w_2 \in W_2(C_+, \bar{h}))$. Then, $S(n) = V(C_+; > 0) \oplus V(C_+; = 0) = W_1(C_+, \bar{h}) \oplus W_2(C_+, \bar{h}) \oplus V(C_+; = 0)$. Let Y_{11} and Y_{12} be bases of $W_1(C_+, \bar{h})$ and $W_2(C_+, \bar{h})$, respectively. Let $Y_1 = (Y_{11}, Y_{12})$ and let Y_2 be a basis of $V(C_+; = 0)$ as a vector subspace of $S(n)$. Therefore, $Y = (Y_1, Y_2) = (Y_{11}, Y_{12}, Y_2)$ is a basis of $S(n)$ as a vector space. With respect to the basis $Y_1 = (Y_{11}, Y_{12})$ of $V(C_+; > 0) = W_1(C_+, \bar{h}) \oplus W_2(C_+, \bar{h})$, we represent*

$$\begin{cases} C_{++} &= \begin{pmatrix} C_{++11} & C_{++12} \\ C_{++21} & C_{++22} \end{pmatrix}, \\ C_{-+} &= \begin{pmatrix} C_{-+11} & C_{-+12} \\ C_{-+21} & C_{-+22} \end{pmatrix}, \\ C_{-+}C_{++}^{-1} &= \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix}. \end{cases}$$

By identification of $T_{\bar{X}^+}S(n) = S(n)$, we assume that $X = \bar{X}^+ + Y$ is a coordinate system of $S(n)$ around $\bar{X}^+$. In the remainder of this paper, we use the following *notations of coordinate systems around $\bar{X}^+$:*

$$\begin{cases} X_{11} = \bar{X}^+ + Y_{11}, \\ X_{12} = \bar{X}^+ + Y_{12}, \\ X_1 \;\; = (X_{11}, X_{12}) = \bar{X}^+ + Y_1, \\ X_2 \;\; = \bar{X}^+ + Y_2 \end{cases}$$

and the notations of derivatives

$$\begin{cases} T \;\;\; = T(C; \bar{X}, \bar{\lambda}, \bar{f}, \bar{h}) \\ \quad\;\;\; = D^2_{X_1} L(\bar{X}^+, \bar{\lambda}; \bar{f}, \bar{h}) + C_{-+}C_{++}^{-1}, \\ T_{ij} = T_{ij}(C; \bar{X}, \bar{\lambda}, \bar{f}, \bar{h}) \\ \quad\;\;\; = D_{X_{1i}} D_{X_{1j}} L(\bar{X}^+, \bar{\lambda}; \bar{f}, \bar{h}) + M_{ij}, \;\; (\forall i, \forall j = 1, 2). \end{cases}$$

LEMMA 5.6. *Suppose that LICQ condition* 2.7 *holds. Let $(\bar{X}, \bar{\lambda}, \bar{f}, \bar{h}) \in S(n) \times \boldsymbol{R}^\ell \times \mathcal{F}$. Let $A \in \partial_{(X,\lambda)}\psi(\bar{X}, \bar{\lambda}; \bar{f}, \bar{h})$ and represent A as*

$$A = \begin{pmatrix} D^2_X L(\bar{X}^+, \bar{\lambda}; \bar{f}, \bar{h}) C_+ + C_- & (D_X \bar{h}(\bar{X}^+))^T \\ D_X \bar{h}(\bar{X}^+) C_+ & O \end{pmatrix}$$

with $C = (C_+, C_-) \in \partial_X \rho(\bar{X})$.

Then, (i) *and* (ii) *hold.*

(i) $\mathrm{rank} D_{X_1} \bar{h}(\bar{X}^+) = \ell$.

(ii) *sgn* $\det A = (-1)^\ell$*sgn* $\det T_{11}(C; \bar{X}, \bar{\lambda}, \bar{f}, \bar{h})$.

Proof. (i): Let $\bar{X}^+ \in S_{r,0}(n)$. Let Y_3 be a basis of $T_{\bar{X}^+} S_{r,0}(n)$ as a vector subspace of $S(n)$ and define $X_3 = \bar{X}^+ + Y_3$. It is readily deduced that $\mathrm{rank} D_{X_3} \bar{h}(\bar{X}^+) = \ell$ from (ii) of LICQ condition 2.7. Since $V(C_+; > 0) \supset T_{\bar{X}^+} S_{r,0}(n)$ from Lemma 4.15, it is directly proved that $\mathrm{rank} D_{X_1} \bar{h}(\bar{X}^+) = \ell$.

(ii): Let $W_1(C_+, \bar{h})$ and $W_2(C_+, \bar{h})$ be the vector subspaces of $S(n)$ defined in Definition 5.5. From $\mathrm{rank}_{X_1} \bar{h}(\bar{X}^+) = \ell$, it follows that $\dim W_2(C_+, \bar{h}) = \ell$. It is also readily inferred that $D_{X_{11}} \bar{h}(\bar{X}^+) = O$ and $G = D_{X_{12}} \bar{h}(\bar{X}^+)$ is a nonsingular matrix of degree ℓ. With Remark 5.2 in mind, a simple calculation leads to $\det A = (-1)^\ell (\det G)^2 \det(C_{++})^{-1} \det T_{11}$ and we have *sgn* $\det A = (-1)^\ell$*sgn* $\det T_{11}$. □

DEFINITION 5.7. *Let U be an open subset of $\boldsymbol{R}^n$ and $F : U \to \boldsymbol{R}^n$ be a continuous map with $\bar{x} \in U$. Take $\delta > 0$ such that $B_\delta(\bar{x}) \subset U$. It is well known from the homology theory that there exists a canonical identification $H_n(B_\delta(\bar{x}; Z), B_\delta(\bar{x}; Z) \backslash \{\bar{x}\}; Z) = Z$, where Z denotes the ring of integers. The theory asserts that F induces the morphism of homology groups: $F_* : Z = H_n(B_\delta(\bar{x}; Z)) \to H_n(B_\delta(F(\bar{x}); Z)) = Z$. It is well known that this morphism F_* is independent on the choice of $\delta > 0$. Then, the Brouwer's degree $deg(\bar{x}; F)$ of the map F around $\bar{x}$ is defined as $deg(\bar{x}; F) = F_*(1) \in Z$.*

Remark 5.8 ([15]). We use the following properties of $deg(\cdot; \cdot)$.

(1) When F is a local homeomorphism around $\bar{x}$, $deg(\bar{x}, F) = F_*(1) = \pm 1$ since $F_* : Z \to Z$ is an isomorphism of the abelian group Z. For example, when F is one-to-one around $\bar{x}$, $deg(\bar{x}; F) = \pm 1$ holds since F is a local homeomorphism around $\bar{x}$ by the Brouwer's invariance theorem of domain.

(2) *Homotopy property:* Let $I = \{t \in \boldsymbol{R} : 0 \le t \le 1\}$ and $F_t : U \times I \to \boldsymbol{R}^n; (x, t) \mapsto F_t(x)$ be continuous. Then $deg(\bar{x}; F_0) = deg(\bar{x}; F_1)$ holds.

(3) It follows from (2) that $deg(x;F)$ is locally constant as the function of x.

(4) Suppose that F is differentiable around $\bar{x}$. Then, $deg(\bar{x};F) = sgn\ \det D_x F(\bar{x})$ holds.

We prove a necessary condition for strong stability in the following proposition, where we denote $deg(X,\lambda;f,h) = deg((X,\lambda);\psi(\cdot,\cdot;f,h))$.

Proposition 5.9. *Suppose that LICQ condition* 2.7 *holds. Let $(\bar{X},\bar{\lambda})$ be a stationary point for* $\mathrm{Pro}(\bar{f},\bar{h})$ *and that $(\bar{X},\bar{\lambda})$ is a strongly stable stationary point for* $\mathrm{Pro}(\bar{f},\bar{h})$. *Then $sgn\ \det(A) = deg(\bar{X},\bar{\lambda};\bar{f},\bar{h})$ for any $A \in ex(\partial_{(X,\lambda)}\psi(\bar{X},\bar{\lambda};\bar{f},\bar{h}))$.*

Proof. Theorem 3.4 asserts that there exist neighborhoods $U = B_{\delta^*}(\bar{X}^+)$ of $\bar{X}^+$ in $S(n)$ and $W = B_\delta((\bar{X},\bar{\lambda}))$ of $(\bar{X},\bar{\lambda})$ with $W \subset (\rho^+)^{-1}(U) \times \boldsymbol{R}^\ell$ such that $V = \{(f,h) \in \mathcal{F} : \psi(\cdot,\cdot;f,h)$ is one-to-one on $W\}$ is a neighborhood of $(\bar{f},\bar{h})$ in $\mathcal{F}_U$. Therefore, from Remark 5.8, we may assume that

$$(1) \qquad s = deg(X,\lambda;f,h) \text{ is nonzero and constant for } (X,\lambda,f,h) \in W \times V.$$

We can deduce a contradiction against value s of degree through exactly the same procedure used by Kojima in [9]. Let $s = deg(\bar{X},\bar{\lambda};\bar{f},\bar{h})$ and $\bar{s} = \left\{\begin{smallmatrix} -1, & (s=1). \\ 1, & (s=-1). \end{smallmatrix}\right.$ Suppose that there exists an element $A \in ex(\partial_{(X,\lambda)}\psi(\bar{X},\bar{\lambda};\bar{f},\bar{h}))$ such that $sgn\ \det(A) = t$ with $t \neq s$, i.e., $t = 0$ or $t = \bar{s}$. Represent A as $A = \left(\begin{smallmatrix} D_X^2 L(\bar{X}^+,\bar{\lambda};\bar{f},\bar{h})C_+ + C_- & (D_X\bar{h}(\bar{X}^+))^T \\ D_X\bar{h}(\bar{X}^+)C_+ & O \end{smallmatrix}\right)$ for some $C \in \partial_{(X,\lambda)}\rho(\bar{X})$. Then $C = (C_+,C_-) \in ex(\partial_X\rho(\bar{X}))$ follows from $A \in ex(\partial_{(X,\lambda)}\psi(\bar{X},\bar{\lambda};\bar{f},\bar{h}))$. We use the same symbols T_{11}, M_{11}, $W_1 = W_1(C_+,\bar{h})$, and $W_2 = W_2(C_+,\bar{h})$ as in the proof of Lemma 5.6, which implies that $sgn\ \det A = (-1)^\ell sgn\ \det T_{11}$ holds. Therefore, $sgn\ \det T_{11} = (-1)^\ell t$. It is readily inferred from definitions of X_{11} and X_{12} that $D_{X_{11}}\bar{h}(\bar{X}^+) = O$ and that $D_{X_{12}}\bar{h}(\bar{X}^+)$ is a nonsingular matrix of degree ℓ. Without difficulties, it can be proved that there exist $\epsilon_0 > 0$ and $B_{11} \in End_{\boldsymbol{R}}(W_1) = W_1 \otimes W_1$ such that $T_{11}(\epsilon) = T_{11} + \epsilon B_{11}$ satisfies that $sgn\ \det T_{11}(\epsilon) = (-1)^\ell \bar{s}$ for any $0 < \forall\epsilon < \epsilon_0$. Let $f_\epsilon(X) = \bar{f}(X) + \epsilon X_{11}^T B_{11} X_{11}$. Simple calculation shows that $A(\epsilon) = \left(\begin{smallmatrix} D_X^2 L(\bar{X}^+,\bar{\lambda};f_\epsilon,\bar{h})C_+ + C_- & (D_X\bar{h}(\bar{X}^+))^T \\ D_X\bar{h}(\bar{X}^+)C_+ & O \end{smallmatrix}\right) \in \partial_{(X,\lambda)}\psi(\bar{x},\bar{\lambda};f_\epsilon,\bar{h})$. It is readily inferred that

$$\begin{cases} T_{11} & = D_{X_{11}}^2 L(\bar{X}^+,\bar{\lambda};\bar{f},\bar{h}) + M_{11}, \\ T_{11}(\epsilon) & = D_{X_{11}}^2 L(\bar{X}^+,\bar{\lambda};f_\epsilon,\bar{h}) + M_{11}. \end{cases}$$

Therefore, it follows from Lemma 5.6 that $sgn\ \det A(\epsilon) = (-1)^\ell sgn\ \det T_{11}(\epsilon) = \bar{s} \neq 0$. Since $C = (C_+,C_-)$ is an extremal element of $\partial_X\rho(\bar{X})$, there exists a sequence $X^{(k)}$, $(k = 1,2,\ldots)$, such that $\lim_{k\to\infty} X^{(k)} = \bar{X}$ and $\lim_{k\to\infty} D_X\rho(X^{(k)}) = C$. Since $\lim_{k\to\infty} D_X\psi(X^{(k)},\bar{\lambda};f_\epsilon,\bar{h}) = A(\epsilon)$ and $sgn\ \det A(\epsilon) = \bar{s} \neq 0$, we have $\lim_{k\to\infty} sgn\ \det D_X\psi(X^{(k)},\bar{\lambda};f_\epsilon,\bar{h}) = \bar{s}$. Especially for large k, we may assume that $sgn\ \det D_X\psi(X^{(k)},\bar{\lambda};f_\epsilon,\bar{h}) = \bar{s}$, which implies that $deg(X^{(k)},\bar{\lambda};f_\epsilon,\bar{h}) = \bar{s}$ by Remark 5.8. This result contradicts (1). □

We prepare some notations for the implicit function theorem 5.11.

Definition 5.10 (see [6]). *Let U be a bounded open subset of $\boldsymbol{R}^n$. We denote by $Lip(U)$ the set of all Lipschitz continuous map from U to $\boldsymbol{R}^n$. For $F \in Lip(U)$, we define*

$$\begin{cases} Lip(F) = \inf\{c : \|F(x) - F(y)\| \leq c\|x - y\| \textit{ for all } x,y \in U\}, \\ \|F\|_{Lip} = \sup_{x\in U} \max\{\|F(x)\|,\ Lip(F)\}, \end{cases}$$

where $\|\cdot\|$ *denotes the Euclidean norm on* $\boldsymbol{R}^n$*. We consider* $Lip(U)$ *to be a normed vector space with its norm* $\|\cdot\|_{Lip}$*. For example,* $\mathrm{int}(B_\mu(F)) = \{G \in Lip(U) : \|F-G\|_{Lip} < \mu\}$.

The next implicit function theorem proved by Jongen, Klatte, and Tammer is important, which we need for the proof of implication (i) from (ii) in Theorem 5.14.

Theorem 5.11 (implicit function theorem, see [6]). *Let* U *be a nonempty open bounded subset of* $\boldsymbol{R}^n$*, let* $F \in Lip(U)$*, and let* $\bar{x} \in U$ *be a point satisfying* $F(\bar{x}) = 0$*. Suppose that* $\partial_x F(\bar{x})$ *is nonsingular. Then there exist positive real numbers* ν, μ *and* γ *such that the following holds:*

(i) *For each* $G \in \mathrm{int}(B_\mu(F))$*,* $B_{\frac{1}{2}\nu}(\bar{x})$ *contains a solution* $x(G)$ *of* $G(x) = 0$ *which is unique in* $\mathrm{int}(B_\nu(\bar{x}))$*.*

(ii) *The map* $G \mapsto x(G)$ *satisfies a Lipschitz condition on* $\mathrm{int}(B_\mu(F))$ *with its modulus* γ*, i.e., if* $G_1,\ G_2 \in \mathrm{int}(B_\mu(F))$*, then* $\|x(G_1) - x(G_2)\| \le \gamma\|G_1 - G_2\|_{Lip}$*.*

We introduce the following condition. This condition always holds for classical programs NLP as showed in Theorem 3.1 of [6]. For NSDP, this condition seems not to hold in general as inferred by Remark 5.13.

Condition 5.12. *If* sgn $\det A = s \neq 0$ *for any* $A \in ex(\partial_{(X,\lambda)}\psi(\bar{X}, \bar{\lambda}; \bar{f}, \bar{h}))$*, then* sgn $\det A = s$ *for any* $A \in \partial_{(X,\lambda)}\psi(\bar{X}, \bar{\lambda}; \bar{f}, \bar{h})$*.*

Remark 5.13. We consider the example 2.2 of [11] with concrete vectors of a_i and b_i. Let $u(\theta) = (\cos\theta, \sin\theta) \in \boldsymbol{R}^2$ and $a_1 = u(0)$, $a_2 = u\left(\frac{\pi}{4}\right)$, $a_3 = u\left(\frac{\pi}{2}\right)$, $a_4 = u\left(\frac{2}{3}\pi\right)$, $a_5 = u\left(\frac{3}{4}\pi\right)$, $a_6 = u\left(\frac{3}{2}\pi\right)$, $b_i = a_i$ for $i = 1,\ 2$ $b_i = -a_i$ for $i = 4,\ 5$, and $b_6 = u\left(\frac{11}{6}\pi\right)$. Let $F : \boldsymbol{R}^2 \to \boldsymbol{R}^2$ be a homeomorphic piecewise linear map that is defined by condition $F(a_i) = b_i$, $(i = 1, 2, \ldots, 6)$. Then, for the zero o of $\boldsymbol{R}^2$, $ex(\partial_x F(\mathbf{0})) = \Big\{G_1 = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$, $G_2 = \left(\begin{smallmatrix} 2 & -1 \\ 1 & 0 \end{smallmatrix}\right)$, $G_3 = \left(\begin{smallmatrix} -1-\sqrt{3} & -1 \\ \sqrt{3} & 0 \end{smallmatrix}\right)$, $G_4 = \left(\begin{smallmatrix} -1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$, $G_5 = \left(\begin{smallmatrix} -1-\frac{\sqrt{3}}{2} & -\frac{3}{2} \\ \frac{3}{2} & \frac{1}{2} \end{smallmatrix}\right)$, $G_6 = \left(\begin{smallmatrix} 1 & -\frac{\sqrt{3}}{2} \\ 0 & \frac{1}{2} \end{smallmatrix}\right)\Big\}$. Simple calculation shows that $\det(G_i) > 0$, $(i = 1, 2, \ldots, 6)$ and $\det\left(\frac{2}{5}G_4 + \frac{3}{5}G_6\right) = -\frac{1}{50} < 0$.

For those NSDP to which Condition 5.12 holds, the following theorem proposes an algebraic criterion for strong stability in terms of Jacobian $D_X h(\bar{X}^+)$ and Hessian $D_X^2 L(\bar{X}^+, \bar{\lambda}; \bar{f}, \bar{h})$.

Theorem 5.14. *Suppose that LICQ condition* 2.7 *holds. Let* $(\bar{X}, \bar{\lambda})$ *be a stationary point for* $\mathrm{Pro}(\bar{f}, \bar{h})$*. Suppose that Condition* 5.12 *holds for* ψ *at* $(\bar{X}, \bar{\lambda})$*. Then the following* (i)–(v) *are equivalent.*

(i) $\bar{X}^+$ *is a strongly stable stationary solution for* $\mathrm{Pro}(\bar{f}, \bar{h})$*.*

(ii) $\psi(X, \lambda; \bar{f}, \bar{h})$ *is nonsingular at* $(\bar{X}, \bar{\lambda})$*.*

(iii) sgn $\det A$ *is nonzero and constant for any* $A \in \partial_{(X,\lambda)}\psi(\bar{X}, \bar{\lambda}; \bar{f}, \bar{h})$*.*

(iv) $Type(A)$ *is constant and* $zero(A) = 0$ *for any* $A \in \partial_{(X,\lambda)}\psi(\bar{X}, \bar{\lambda}; \bar{f}, \bar{h})$*.*

(v) sgn $\det T_{11}(C; \bar{X}, \bar{\lambda}, \bar{f}, \bar{h})$ *is nonzero and constant for any* $C \in \partial_X \rho(\bar{X})$*.*

Proof. Equivalence between (ii), (iii), and (iv) is readily deduced by Lemma 4.5; equivalence between (iii) and (v) is directly deduced from Lemma 5.6. The implication of (i) from (ii) is clear from the implicit function theorem 5.11. The implication of (iii) from (i) is deduced from Proposition 5.9 and the assumption that Condition 5.12 holds at $(\bar{X}, \bar{\lambda})$. □

Definition 5.15. *Suppose that LICQ condition* 2.7 *holds. Let* $(\bar{X}, \bar{\lambda})$ *be a stationary point for* $\mathrm{Pro}(\bar{f}, \bar{h})$*. Suppose that Condition* 5.12 *holds at* $(\bar{X}, \bar{\lambda})$*. Then we can define the stationary index* $s.index(\bar{X}^+; \bar{f}, \bar{h})$ *after Kojima* [9] *by* $s.index(\bar{X}^+; \bar{f}, \bar{h}) = nega(T_{11}(C; \bar{X}, \bar{\lambda}, \bar{f}, \bar{h}))$*. We remark that this definition is independent of choice of*

$C \in \partial_X \rho(\bar{X})$; therefore, it is also independent of $A \in \partial_{(X,\lambda)} \psi(\bar{X}, \bar{\lambda}; \bar{f}, \bar{h})$ from Lemma 4.5.

Remark 5.16. Theorem 5.14 for NSDP corresponds to Corollary 4.3 in [9] for NLP. It shows that we have similar results as in [9] for those NSDP to which Condition 5.12 holds. Therefore, we could understand that the essential difference between NSDP and NLP lies in Condition 5.12.

6. Conclusions. We investigated strong stability, in the sense of Kojima, of stationary solutions of nonlinear positive semidefinite programs NSDP. In Theorem 3.4 we give a characterization of stability from the point of view of one-to-one maps under LICQ condition 2.7. In sections 4 and 5 we construct a method for NSDP analogous to Kojima's method for classical nonlinear programs NLP, and from the construction we make clear the essential difference between NSDP and NLP. In Theorem 5.14 we also deduce an algebraic condition equivalent to strong stability for those NSDP to which there does not exist the above difference.

Acknowledgment. The author is much obliged to the editor and the anonymous referees for many reexaminations, and would like to thank them for all of their valuable suggestions and remarks.

REFERENCES

[1] J. F. Bonnans and A. Shapiro, *Perturbation Analysis of Optimization Problems*, Springer-Verlag, New York, 2000.

[2] F. M. Clarke, *Optimization and Nonsmooth Analysis*, Wiley-Interscience, New York, 1983.

[3] H. Federer, *Geometric Measure Theory*, Springer-Verlag, New York, 1969.

[4] A. Graham, *Kronecker Products and Matrix Calculus: With Applications*, J. Wiley and Sons, New York, 1981.

[5] B. Iversen, *Cohomology of Sheaves*, Springer-Verlag, Berlin, 1986.

[6] H. Th. Jongen, D. Klatte, and K. Tammer, *Implicit functions and sensitivity of stationary points*, Math. Programming, 49 (1990/91), pp. 123–138.

[7] D. Klatte and B. Kummer, *Strong stability in nonlinear programming revised*, J. Austral. Math. Soc. Ser. B, 40 (1999), pp. 336–352.

[8] D. Klatte and B. Kummer, *Nonsmooth Equations in Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.

[9] M. Kojima, *Strongly stable stationary solutions in nonlinear programs*, in Analysis and Computation of Fixed Points, S. M. Robinson, ed., Academic Press, New York, 1980, pp. 93–138.

[10] M. Kojima, *Semidefinite programming and interior-point methods*, Bulletin of the Japan Society for Industrial and Applied Mathematics, 6 (1996), pp. 16–25 (in Japanese).

[11] B. Kummer, *Lipschitzian inverse functions, directional derivatives and application in $C^{1,1}$ optimization*, J. Optim. Theory Appl., 70 (1991), pp. 561–581.

[12] B. Kummer, *An implicit-function theorem for $C^{0,1}$-equations and parametric $C^{1,1}$-optimization*, J. Math. Anal. Appl., 158 (1991), pp. 35–46.

[13] A. Levy, *Solution sensitivity from general principles*, SIAM J. Control Optim., 40 (2001), pp. 1–38.

[14] T. Matsumoto, *On the stability of stationary solutions of nonlinear positive semidefinite programs*, J. Oper. Res. Soc. Japan, 46 (2003), pp. 22–34.

[15] J. M. Ortega and W. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[16] J.-S. Pang, D. Sun, and J. Sun, *Semismooth homeomorphisms and strong stability of semidefinite and Lorentz complementarity problems*, Math. Oper. Res., 28 (2003), pp. 39–63.

[17] R. Rockafellar and R. Wets, *Variational Analysis*, Springer-Verlag, Berlin, 1998.

[18] E. H. Zarantonello, *Projections on convex sets in Hilbert space and spectral theory*, in Contributions to Nonlinear Functional Analysis, E. H. Zarantonello, ed., Academic Press, New York, pp. 237–341.

SIAM J. OPTIM.
Vol. 16, No. 2, pp. 471–489

ON THE CONVERGENCE OF SUCCESSIVE LINEAR-QUADRATIC PROGRAMMING ALGORITHMS[*]

RICHARD H. BYRD[†], NICHOLAS I. M. GOULD[‡], JORGE NOCEDAL[§], AND RICHARD A. WALTZ[§]

Abstract. The global convergence properties of a class of penalty methods for nonlinear programming are analyzed. These methods include successive linear programming approaches and, more specifically, the successive linear-quadratic programming approach presented by Byrd et al. [*Math. Program.*, 100 (2004), pp. 27–48]. Every iteration requires the solution of two trust-region subproblems involving piecewise linear and quadratic models, respectively. It is shown that, for a fixed penalty parameter, the sequence of iterates approaches stationarity of the penalty function. A procedure for dynamically adjusting the penalty parameter is described, and global convergence results for it are established.

Key words. sequential linear programming, global convergence theory, nonlinear optimization, penalty parameter updates

AMS subject classifications. 65K05, 90C26, 90C30, 90C51

DOI. 10.1137/S1052623403426532

1. Introduction.

In this paper we study the global convergence properties of successive linear-quadratic programming (SLQP) algorithms for nonlinear programming. The problem under consideration is

$$\min_{x} \ f(x) \tag{1.1a}$$

$$\text{subject to} \ \ h(x) = 0 \tag{1.1b}$$

$$g(x) \geq 0, \tag{1.1c}$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$, and the constraint functions $h : \mathbb{R}^n \to \mathbb{R}^{m_h}$, $g : \mathbb{R}^n \to \mathbb{R}^{m_g}$, are assumed to be continuously differentiable.

The class of algorithms studied in this paper solves (1.1) via the related problem

$$\underset{x}{\text{minimize}} \quad \phi_\sigma(x), \tag{1.2}$$

where

$$\phi_\sigma(x) = f(x) + \sigma\|h(x)\| + \sigma\|g^-(x)\| \tag{1.3}$$

[*] Received by the editors April 17, 2003; accepted for publication (in revised form) April 18, 2005; published electronically November 14, 2005.

http://www.siam.org/journals/siopt/16-2/42653.html

[†] Department of Computer Science, University of Colorado, Boulder, CO 80309 (richard@cs.colorado.edu). This author was supported by Army Research Office grants DAAG55-98-1-0176 and DAAD19-02-1-0407 and by NSF grants CCR-0219190 and CHE-0205170.

[‡] Computational Science and Engineering Department, Rutherford Appleton Laboratory, Chilton, Oxfordshire OX11 0QX, England (n.i.m.gould@rl.ac.uk). This author was supported by EPSRC grants GR/R46641 and GR/S42170.

[§] Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL 60208-3118 (nocedal@ece.northwestern.edu, rwaltz@ece.northwestern.edu). These authors were supported by National Science Foundation grants ATM-0086579 and CCR-0219438 and by Department of Energy grant DE-FG02-87ER25047-A004.

is an exact penalty function [5, 12] composed of the objective and constraint functions from (1.1). Here $\|\cdot\|$ is a polyhedral norm, $g^-(x)$ is defined componentwise as

$$g_i^-(x) = \min(g_i(x), 0),$$

and $\sigma > 0$ is a parameter that is adaptively chosen so that critical points of (1.1) correspond to those of (1.2). For fixed σ, each iteration of a typical algorithm comprises two phases. In the first (linear) phase, a piecewise linear model of the penalty function ϕ_σ is minimized subject to a trust-region bound. The aim here is to compute a step for which convergence can be guaranteed. The second (quadratic) phase adjusts this step by reducing a quadratic model of the penalty function within a (second) trust-region bound, with the aim of accelerating the convergence of the method. A primary purpose of this article is to establish the global convergence of this class of methods. Once this has been established, it remains to consider methods for adjusting the penalty parameter so as to ensure convergence of the overall algorithm to KKT points for (1.1) or, failing this, critical points of some measure of constraint infeasibility.

This work is motivated by a recently proposed algorithm, described by the authors in [1], and is related to the SLQP algorithm proposed by Fletcher and Sainz de la Maza [8]. In [1] the ℓ_1-norm is used to define the penalty function (1.3). The linear phase utilizes a piecewise linear model of (1.3) at the current iterate x_k,

$$\ell(x_k, d) = f(x_k)+\nabla f(x_k)^T d+\sigma\|h(x_k)+\nabla h(x_k)^T d\|+\sigma\|(g(x_k)+\nabla g(x_k)^T d)^-\|. \tag{1.4}$$

Defining $\ell_k(d) \overset{\text{def}}{=} \ell(x_k, d)$ and imposing an ℓ_∞-norm trust region whose radius is given by the scalar parameter $\Delta_k^{\text{LP}} > 0$, the linear phase consists of solving the (piecewise) linear program (LP)

$$\begin{aligned} &\underset{d}{\text{minimize}} \;\; \ell_k(d) \\ &\text{subject to} \;\; \|d\|_\infty \le \Delta_k^{\text{LP}}, \end{aligned}$$

whose solution we denote by d_k^{LP}. A working set $\mathcal{W}_k$ is subsequently defined as the set of constraints that are active at the solution of this problem if these constraints are linearly independent, or otherwise some linearly independent subset of these.

The quadratic phase of the algorithm described in [1] computes a step d_k that makes progress on a piecewise quadratic function

$$q_k(d) = \ell_k(d) + \tfrac{1}{2} d^T B_k d, \tag{1.5}$$

subject to a trust-region constraint, where B_k approximates the Hessian of the Lagrangian of the nonlinear program (1.1). The step computation in the quadratic phase is carried out by solving an equality-constrained quadratic programming problem of the form

$$\underset{d}{\text{minimize}} \;\; \tfrac{1}{2} d^T B_k d + (\nabla\phi_\sigma)_k^T d \tag{1.6a}$$

$$\text{subject to} \;\; h_i(x_k) + \nabla h_i(x_k)^T d = 0, \quad i \in \mathcal{E} \cap \mathcal{W}_k, \tag{1.6b}$$

$$g_i(x_k) + \nabla g_i(x_k)^T d = 0, \quad i \in \mathcal{I} \cap \mathcal{W}_k, \tag{1.6c}$$

$$\|d\|_2 \le \Delta_k, \tag{1.6d}$$

where $(\nabla\phi_\sigma)_k$ is the gradient of the part of (1.3) corresponding to the objective function and the violated constraints, and $\mathcal{E}$ and $\mathcal{I}$ denote the sets of equality and inequality constraints, respectively. Notice that in this phase, an ℓ_2-norm trust region is used, and the trust-region parameter Δ_k is distinct from the trust-region parameter Δ_k^{LP} used in the linear phase. The overall step taken by the algorithm is obtained by minimizing q_k along a path formed by d_k^{LP} and d_k, in a manner described in [1].

Our algorithm [1] is distinct from the one proposed by Fletcher and Sainz de la Maza [8] in two important ways. First, the trial step generated by our algorithm is formed from a convex combination of the linear phase step d_k^{LP} and the quadratic phase step d_k, whereas *either* the step d_k or the step d_k^{LP} is taken in [8]. Second, our algorithm imposes a trust-region restriction on the second subproblem, and thus permits the use of second derivatives of the objective function and constraints in the definition of B. The two trust-region radii operate quasi-independently, and the update rules used in [1] will be shown in this paper to offer global convergence guarantees.

The organization of the paper is as follows. In the remainder of this section we discuss the application of SLQP methods to general composite nonsmooth problems and briefly review existing SLQP methods. In section 2 we present an algorithm for the minimization of the penalty function with fixed penalty parameter. We study the global convergence properties of such an algorithm in section 3. Procedures for updating the penalty parameter are studied in section 4. The paper concludes with some final remarks and perspectives.

1.1. The general composite nonsmooth context.

It is worth pointing out that problem (1.2) is a nonsmooth problem that is a special case of the more general class of *composite nonsmooth* optimization problems that can be represented as

$$\underset{x}{\text{minimize}}\ \omega(F(x)) \tag{1.7}$$

for some smooth function $F(x)$ and convex ω. Problem (1.2) has this form if we let

$$F(x) = (f(x), g(x), h(x)) \tag{1.8}$$

and define

$$\omega(F(x)) = f(x) + \sigma\|h(x)\| + \sigma\|g^-(x)\|. \tag{1.9}$$

Many nondifferentiable approximation problems may also be put in this form.

In this context, the linearized model $\ell(x_k, d)$ in (1.4) corresponds to

$$\omega\left(F(x_k) + F'(x_k)d\right). \tag{1.10}$$

The strategy described above corresponds to minimizing (1.10) at the current iterate x_k, subject to $\|d\|_\infty \le \Delta_k^{\text{LP}}$, and using the result to help compute a step making progress on the function

$$\ell_k(d) + \tfrac{1}{2}d^TB_kd. \tag{1.11}$$

The algorithm described in section 3 applies equivalently to problem (1.7), as does the convergence analysis in section 4.

1.2. Existing SLQP algorithms. To the best of our knowledge, the earliest SLQP method was proposed by Fletcher and Sainz de la Maza [8], based on ideas in [6, 13]. The method is described in terms of general composite nonsmooth optimization problems of the form (1.7). At the iterate x_k, a linearized approximation of the form (1.10) is minimized within a given trust region. A solution to this problem, d_k^{LP}, is then used to assess the suitability of a trial step d_k, obtained without regard to the trust region and by whatever means is appropriate. If a finite number of different attempts to find a suitable d_k have failed, the choice $d_k = d_k^{\text{LP}}$ is tried, and if this too fails, x_{k+1} is left at x_k and the trust-region radius reduced. Fletcher and Sainz de la Maza suggest using the subdifferential structure of ω predicted by $\ell_k(d_k^{\text{LP}})$ as one means of finding d_k. Specifically, the minimizer of the (locally) smooth part of the quadratic model q_k is minimized subject to the linearized (locally) nonsmooth part being unchanged. This "equality-constrained" quadratic program (EQP) is invariably a far simpler problem than trying to minimize q_k. Importantly, Fletcher and Sainz de la Maza show that, under reasonable nondegeneracy and second-order conditions, the "active" subdifferential structure of $\ell_k(d_k^{\text{LP}})$ ultimately predicts that of $\omega(F)$ at limit points of $\{x_k\}$, and thus that the EQP leads to fast asymptotic convergence.

A more recent SLQP method due to Chin [2] and Chin and Fletcher [3] is aimed specifically at the nonlinear programming problem (1.1). Rather than using the nonsmooth penalty function (1.3) to force convergence, Chin and Fletcher use a nonlinear programming "filter" [7] to do so. A succession of steps are allowed at each iteration, in which unbounded quadratic programming steps of various forms are given precedence over linear programming ones. Nevertheless, as with the methods in [1] and [8], the linear programming subproblem

$$\begin{array}{ll} \underset{d}{\text{minimize}} & d^T \nabla f(x_k) \\ \text{subject to} & h(x_k) + \nabla h(x_k)^T d = 0, \\ & g(x_k) + \nabla g(x_k)^T d \geq 0, \\ & \|d\|_\infty \leq \Delta_k \end{array} \tag{1.12}$$

is central and drives the convergence of the method. In particular, if d_k^{LP} is a solution[1] of (1.12), and if more complicated steps are unacceptable for the filter, the method reverts to a Cauchy step along d_k^{LP}. The trust-region radius will only be reduced as a last resort.

While this is undoubtedly an SLQP method, it is once again a trust region on the linear programming component that is used to force convergence. There appears to be no control of any quadratic programming component, and thus no precaution to guard against large or unbounded QP steps.

Most recently, Waltz [14] and Gate [9] suggested the idea of using a second trust region to control the EQP phase of SLQP methods. Waltz's method forms the basis of that described in [1] and analyzed here. Gate's method is an extension of the Chin–Fletcher filter approach, and although there is no formal analysis, this method appears to perform well in his numerical tests.

It should be noted that the theory of nonsmooth optimization developed by Yuan [15, 16] cannot be applied to the algorithm considered here and in [1, 14] because in these algorithms the two trust regions influence each other, whereas Yuan assumes that a single trust region is used. The analysis presented here is significantly different from that in the literature due to the effects caused by the interactions between the

[1] If (1.12) has no solution, a "restoration" phase [3] is entered.

two trust regions. In addition, we establish new results about update procedures for the penalty parameter.

2. An SLQP algorithm.

Our first goal is to propose and analyze an algorithm for minimizing the penalty function ϕ_σ, given by (1.3), for a fixed value of σ. Notice that this analysis presupposes that the penalty parameter σ has been fixed at a sufficiently large value such that critical points of (1.1) correspond to those of (1.2), but we will delay a discussions of suitable mechanisms to ensure that this is so until section 4.

As noted earlier, the algorithm consists of two phases based, respectively, on piecewise linear and piecewise quadratic models at the current estimate x_k of the minimizer. The first phase minimizes the piecewise linear model $\ell_k(d)$, given by (1.4). The second phase is based on an appropriate piecewise quadratic model $q_k(d)$ of the form (1.5) that includes a second-order term to account for curvature. For the linear model, we will use a trust region of the form $\|\cdot\|_{\text{LP}} \le \Delta^{\text{LP}}$ for some (polyhedral) norm $\|\cdot\|_{\text{LP}}$, while for the quadratic model it will be $\|\cdot\| \le \Delta$. Since all norms are equivalent in $\mathbb{R}^n$, there is a constant $\gamma \ge 1$ such that

$$\|d\| \le \gamma \|d\|_{\text{LP}} \tag{2.1}$$

for all $d \in \mathbb{R}^n$.

We now define our Algorithm 2.1 for minimizing the penalty function (1.3) for a fixed value of σ. Throughout this section we omit the subscript and refer to our penalty function simply as ϕ in the case where σ is fixed.

Step 1 of Algorithm 2.1 aims to find the largest reduction in the linearized model within its trust region—we refer to this as the *linearized* problem—and attach the suffix LP to quantities associated with it. The intentions here are twofold.

First, the aim is to identify constraints whose inclusion in the working set for an EQP results in progress in the overall minimization. Ideally, near the solution these will correspond to active constraints at the solution. This is not the issue under consideration here, but it does have some ramifications on the design of our algorithm since we hope that our algorithm class is broad enough to permit correct identification of the active constraint set at the solution.

Second, the direction given by d_k^{LP} is also used to define the Cauchy step d_k^{C}, which, as in many trust-region methods, is used to guarantee convergence to a critical point. This is because the value of the LP solution provides a measure of nearness to optimality, and the Cauchy step is a step that provides corresponding improvement on the quadratic model. Condition (2.5) ensures that d_k^{C} is short enough that the quadratic model value is related to the LP model. The descent properties of the Cauchy step are what drive the bulk of our convergence theory; thus we ensure in step 2b that the step actually taken, d_k, shares these descent properties. Note that the Cauchy step d_k^{C} satisfies the conditions of step 2b, but the intention is to find a better step by solving a problem of the form (1.6).

Steps 3 and 4 are standard trust-region acceptance rules [4]. The ratio ρ_k of the actual to the predicted reduction of ϕ is used as a step acceptance criterion. If this ratio is negative, or close to zero, the step is rejected and the overall trust-region radius reduced. Otherwise the step will be accepted and, if ρ_k is close to one, the radius may be enlarged. We say that iteration k is *successful* if $\rho_k \ge \rho_u$. It is *very successful* if $\rho_k \ge \rho_s$.

```
ALGORITHM 2.1: MINIMIZATION ALGORITHM FOR $\phi(\mathbf{x})$

Initial data: $x_0$, $\Delta_0 > 0$, $\Delta_0^{\text{LP}} > 0$, $0 < \rho_u \le \rho_s < 1$, $0 < \kappa_l \le \kappa_u < 1$, $\eta > 0$,
$0 < \tau < 1$, and $\theta > 0$.

For $k = 0, 1, \ldots$, until a stopping test is satisfied, perform the following steps.
1.  Compute a solution $d_k^{\text{LP}}$ to
        $\underset{\|d\|_{\text{LP}} \le \Delta_k^{\text{LP}}}{\text{minimize}} \;\; \ell_k(d).$
2a. Cauchy step. Compute $\alpha_k \le 1$ as the first member of the sequence
        $\{\tau^i \min(1, \Delta_k / \|d_k^{\text{LP}}\|)\}_{i=0,1,\ldots}$ for which
        (2.2)   $\phi(x_k) - q_k(\alpha_k d_k^{\text{LP}}) \ge \eta \, [\phi(x_k) - \ell_k(\alpha_k d_k^{\text{LP}})] \, .$
    Set $d_k^{\text{C}} = \alpha_k d_k^{\text{LP}}$.
2b. Compute $d_k$ so that $\|d_k\| \le \Delta_k$ and
        $q_k(d_k) \le q_k(d_k^{\text{C}}).$
3.  Compute
        $\rho_k = \dfrac{\phi(x_k) - \phi(x_k + d_k)}{\phi(x_k) - q_k(d_k)}.$
4a. If $\rho_k \ge \rho_s$, choose
        $\Delta_{k+1} \ge \Delta_k.$
    Otherwise set
        (2.3)   $\Delta_{k+1} \in [\kappa_l \|d_k\|, \kappa_u \Delta_k] \, .$
4b. If $\rho_k \ge \rho_u$, set
        $x_{k+1} = x_k + d_k.$
    Otherwise set
        $x_{k+1} = x_k.$
5.  LP trust-region update.
    If $\rho_k \ge \rho_u$, pick $\Delta_{k+1}^{\text{LP}}$ so that the following two conditions hold:
        (2.4)   (i)   $\Delta_{k+1}^{\text{LP}} \ge \|d_k^{\text{C}}\|_{\text{LP}}$ ,
        (2.5)   (ii)  $\Delta_{k+1}^{\text{LP}} \le \Delta_k^{\text{LP}}$   if  $\alpha_k < 1.$
    Otherwise pick
        (2.6)   $\Delta_{k+1}^{\text{LP}} \in [\min(\theta \|d_k\|_{\text{LP}} \, , \Delta_k^{\text{LP}}), \Delta_k^{\text{LP}}].$
```

Step 5 gives the conditions imposed on the radius for the linear model. In [1] a specific strategy is described that tries to relate Δ^{LP} to the expected step length so as to promote selection of a good active set. However, in this algorithm framework we only specify the characteristics such a strategy must have in order to guarantee global convergence. In the case of a successful step, we impose a limit on how much Δ^{LP} may be reduced, and allow increase only if the full LP step was taken. If the step d_k was not successful we allow for the possibility of decreasing Δ^{LP}, as Δ_k was decreased in step 4a.[2]

3. Convergence results for a fixed penalty function. In this section, we investigate the global convergence properties of Algorithm 2.1. In order to proceed, we need to make the following assumptions on the problem and the algorithm:

P1. The functions f, g, and h in (1.1) are Lipschitz continuous and have Lipschitz continuous derivatives over a bounded convex set whose interior contains the closure of the iterates $\{x_k\}$ generated by Algorithm 2.1.

P2. The sequence of Hessian matrices $\{B_k\}$ in (1.5) is bounded; thus there exists a constant $\beta > 0$ such that $|d^T B_k d| \le \beta \|d\|^2$ for all k and all $d \in \mathbb{R}^n$.

Assumption P2 is made to simplify the analysis; see [15] for an analysis of a composite nonsmooth optimization algorithm in which B_k is computed by quasi-Newton updating. (As pointed out in section 1.1, both Algorithm 2.1 and the analysis in this section apply also to the case where $\phi(x) = \omega(F(x))$, with ℓ_k and q_k given by (1.10) and (1.11). In this case assumption P1 requires Lipschitz continuity of F, F', and ω.)

Under assumption P1 it follows immediately that $\phi(x)$ and $\ell_k(d)$ are Lipschitz continuous, and in particular that

$$|\ell_k(d) - \ell_k(0)| \le \lambda \|d\|_{\text{LP}} \tag{3.1}$$

for some Lipschitz constant $\lambda > 0$.

The goal of our analysis is to prove that Algorithm 2.1 will find a critical point of ϕ. To do so, we follow Yuan [15] and define

$$\Psi(x, \Delta) = \ell(x, 0) - \min_{\|d\| \le \Delta} \ell(x, d), \tag{3.2}$$

which is the optimal decrease in the “linear” model $\ell(x, d)$ for a radius of size Δ. We can characterize criticality of ϕ using Ψ.

DEFINITION 3.1. *$x_* \in \mathbb{R}^n$ is a critical point (or stationary point) of ϕ if $\Psi(x_*, 1) = 0$.*

For future reference we note that, from assumption P2 and the subsequent convexity of $\ell(x, \cdot)$, we have in general

$$\ell(x, 0) - \ell(x, \alpha d) \ge \alpha[\ell(x, 0) - \ell(x, d)], \tag{3.3}$$

and more specifically

$$\phi(x_k) - \ell_k(\alpha d) \ge \alpha[\phi(x_k) - \ell_k(d)] \tag{3.4}$$

for any $\alpha \in [0, 1]$.

We now establish a number of intermediate lemmas leading up to our main global convergence result. Our first result provides bounds on the achievable reduction in

[2] The upper bound of one on α_k in (2.5) is used for simplicity. However, this bound can be generalized.

the linearized model for a radius of size Δ relative to that achieved with a radius of 1. From now on we use the following notation.

Notation. The solution d^{LP} of

$$\min_{\|d\|_{\text{LP}} \leq \Delta} \ell(x, d) \tag{3.5}$$

will also be denoted as d_Δ to emphasize its dependence on Δ. In particular, d_1 denotes the solution of (3.5) when $\Delta = 1$.

LEMMA 3.1. *Suppose that assumptions* P1 *and* P2 *hold. Then*

$$\max(\Delta, 1)\Psi(x_k, 1) \geq \Psi(x_k, \Delta) \geq \min(\Delta, 1)\Psi(x_k, 1) \tag{3.6}$$

for any scalar $\Delta > 0$.

Proof. Since d_Δ is a solution of (3.5),

$$\Psi(x_k, \Delta) = \ell(x_k, 0) - \ell(x_k, d_\Delta).$$

There are two cases to consider. First consider the case $\Delta \leq 1$. Since $\|d_\Delta\|_{\text{LP}} \leq 1$, the definition (3.2) implies that

$$\Psi(x_k, 1) \geq \ell(x_k, 0) - \ell(x_k, d_\Delta) = \Psi(x_k, \Delta),$$

which gives the left inequality of (3.6) in this case.

To get the right inequality, we need to show that $\Psi(x_k, \Delta) \geq \Delta\Psi(x_k, 1)$. By definition of d_Δ we have that $\|\Delta d_1\|_{\text{LP}} \leq \Delta$, and so by (3.2) and (3.3),

$$\begin{aligned}\Psi(x_k, \Delta) &\geq \ell(x_k, 0) - \ell(x_k, \Delta d_1)\\ &\geq \Delta(\ell(x_k, 0) - \ell(x_k, d_1))\\ &= \Delta\Psi(x_k, 1).\end{aligned}$$

This gives us (3.6) when $\Delta \leq 1$. In the case $\Delta \geq 1$, we need to establish

$$\Delta\Psi(x_k, 1) \geq \Psi(x_k, \Delta) \geq \Psi(x_k, 1),$$

but these inequalities follow immediately by making the above two-case argument with the values Δ and 1 interchanged. □

Lemma 3.1 essentially states that $\Psi(x, \cdot)$ is concave and monotonically increasing.

We shall also need the following result, which states that at a noncritical point of ϕ, the trust-region bound for the linearized problem, $\|d_\Delta\|_{\text{LP}} \leq \Delta$, is active whenever the radius Δ is small enough. For brevity, let $\Psi_k(\Delta) \overset{\text{def}}{=} \Psi(x_k, \Delta)$.

LEMMA 3.2. *Suppose that assumptions* P1 *and* P2 *hold (and thus that there is a Lipschitz constant* λ *for which* (3.1) *holds) and that* $\Psi_k(1) \neq 0$. *Then if* d_Δ *is a solution of* (3.5) *when* $x = x_k$,

$$\|d_\Delta\|_{LP} \geq \min\left(\Delta, \frac{\Psi_k(1)}{\lambda}\right). \tag{3.7}$$

Proof. As before, let d_1 denote a solution of (3.5) when $x = x_k$ and $\Delta = 1$. Suppose that $\|d_\Delta\|_{\text{LP}} < \Psi_k(1)/\lambda$. Then (3.1) gives that

$$\ell_k(d_\Delta) \geq \ell_k(0) - \lambda\|d_\Delta\|_{\text{LP}} > \ell_k(0) - \Psi_k(1) = \ell_k(d_1). \tag{3.8}$$

If $\Delta \geq 1$, this contradicts our definition of d_Δ as a solution of (3.5), so we must have $\|d_\Delta\|_{\text{LP}} \geq \Psi_k(1)/\lambda$ and thus (3.7) in this case. If $\Delta < 1$, then (3.8) and the convexity of ℓ_k imply that ℓ_k is strictly decreasing along a line from d_Δ to d_1 (at least initially). Therefore, since d_Δ minimizes ℓ_k, it cannot lie in the strict interior of the trust region $\|d\|_{\text{LP}} \leq \Delta$, and hence $\|d_\Delta\|_{\text{LP}} = \Delta$. □

The next result provides a lower bound on the achievable reduction in the piecewise quadratic model in terms of the stepsize, the trust-region radius for the linearized problem, and our criticality measure. At this point, recall that we use d_k^{LP} to refer to the solution of the linear subproblem (3.5) solved in step 1 of Algorithm 2.1.

LEMMA 3.3. *Suppose that assumptions* P1 *and* P2 *hold. Then the model decrease satisfies*

$$\phi(x_k) - q_k(d_k) \geq \phi(x_k) - q_k(d_k^C) \geq \eta\alpha_k\Psi_k(\Delta_k^{LP}) \geq \eta\alpha_k \min(\Delta_k^{LP}, 1)\Psi_k(1).$$

Proof. The first inequality follows directly from the requirement in step 2b of Algorithm 2.1. To prove the second, note that inequality (3.4) and the requirement in step 2a give that

$$\begin{aligned} \phi(x_k) - q_k(d_k^{\text{C}}) &= \phi(x_k) - q_k(\alpha_k d_k^{\text{LP}}) \geq \eta\,[\phi(x_k) - \ell_k(\alpha_k d_k^{\text{LP}})] \\ &\geq \eta\alpha_k\,[\phi(x_k) - \ell_k(d_k^{\text{LP}})] = \eta\alpha_k\Psi_k(\Delta_k^{\text{LP}}). \end{aligned}$$

The third inequality follows immediately from Lemma 3.1. □

Next, we establish an intuitive bound on the error introduced when using our quadratic approximation to ϕ.

LEMMA 3.4. *Suppose that assumptions* P1 *and* P2 *hold. Then*

$$|q_k(d_k) - \phi(x_k + d_k)| \leq M\|d_k\|^2$$

for some positive constant M.

Proof. As pointed out in section 1.1, the function ϕ can be expressed as $\phi(x) = \omega(F(x))$, where F and ω are defined as in (1.8) and (1.9). It follows from assumption P1 that F has a Lipschitz continuous derivative with constant λ^{F}, which implies that

$$\|F(x_k + d_k) - F(x_k) - F'(x_k)d_k\| \leq \lambda^{\text{F}}\|d_k\|^2.$$

Since the function ω is Lipschitz continuous with some constant λ^ω, this inequality, together with assumption P2, implies that

$$\begin{aligned} |q_k(d_k) - \phi(x_k + d_k)| &= |\omega(F(x_k) + F'(x_k)d_k) + \tfrac{1}{2}d_k^T B_k d_k - \omega(F(x_k + d_k))| \\ &\leq \lambda^\omega\|F(x_k + d_k) - F(x_k) - F'(x_k)d_k\| + \tfrac{1}{2}\beta\|d_k\|^2 \\ &\leq (\lambda^\omega\lambda^{\text{F}} + \tfrac{1}{2}\beta)\|d_k\|^2 \\ &= M\|d_k\|^2, \end{aligned}$$

where $M = \lambda^\omega\lambda^{\text{F}} + \frac{1}{2}\beta$. □

The following technical result essentially says that either the Cauchy step is on the boundary of one of our trust regions or it has a lower bound proportional to the optimality criterion.

LEMMA 3.5. *Suppose that assumptions* P1 *and* P2 *hold. Then at any iteration of Algorithm* 2.1

$$\alpha_k\Delta_k^{LP} \geq \|d_k^C\|_{LP} \geq \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{LP}, \frac{\Psi_k(1)}{\lambda}, \min\left(1, \frac{1}{\Delta_k^{LP}}\right)\frac{2(1-\eta)\tau\Psi_k(1)}{\beta\gamma^2}\right). \tag{3.9}$$

Proof. The first inequality in (3.9) follows immediately, since

$$\|d_k^{\mathrm{C}}\|_{\mathrm{LP}} = \alpha_k \|d_k^{\mathrm{LP}}\|_{\mathrm{LP}} \leq \alpha_k \Delta_k^{\mathrm{LP}}.$$

To establish the second inequality, suppose first that the decrease condition (2.2) in step 2a of Algorithm 2.1 is immediately satisfied for $\alpha_k = \min(1, \Delta_k/\|d_k^{\mathrm{LP}}\|)$. Then, using (2.1) and Lemma 3.2,

$$\begin{aligned} \|d_k^{\mathrm{C}}\|_{\mathrm{LP}} = \|\alpha_k d_k^{\mathrm{LP}}\|_{\mathrm{LP}} &= \min\left(\frac{\Delta_k}{\|d_k^{\mathrm{LP}}\|}, 1\right) \|d_k^{\mathrm{LP}}\|_{\mathrm{LP}} \\ &\geq \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\mathrm{LP}}, \frac{\Psi_k(1)}{\lambda}\right), \end{aligned} \tag{3.10}$$

which gives the first three terms in (3.9). On the other hand, if $\alpha_k < \min(1, \Delta_k/\|d_k^{\mathrm{LP}}\|)$, then the decrease condition (2.2) must have been violated for α_k/τ, and so

$$\begin{aligned} \phi(x_k) - q_k(\alpha_k d_k^{\mathrm{LP}}/\tau) &= \phi(x_k) - \ell_k(\alpha_k d_k^{\mathrm{LP}}/\tau) - \tfrac{1}{2}(\alpha_k/\tau)^2 (d_k^{\mathrm{LP}})^T B_k d_k^{\mathrm{LP}} \\ &\leq \eta \left[\phi(x_k) - \ell_k(\alpha_k d_k^{\mathrm{LP}}/\tau)\right]. \end{aligned} \tag{3.11}$$

Now using assumption P2, (2.1), (3.4), and Lemma 3.1, this inequality implies that

$$\begin{aligned} \tfrac{1}{2}(\alpha_k/\tau)^2 (d_k^{\mathrm{LP}})^T B_k d_k^{\mathrm{LP}} &\geq (1-\eta)\left[\phi(x_k) - \ell_k(\alpha_k d_k^{\mathrm{LP}}/\tau)\right] \\ \tfrac{1}{2}(\alpha_k/\tau)^2 \beta\gamma^2 \|d_k^{\mathrm{LP}}\|_{\mathrm{LP}}^2 &\geq (1-\eta)(\alpha_k/\tau)\Psi_k(\Delta_k^{\mathrm{LP}}) \\ \tfrac{1}{2}(\alpha_k/\tau)\beta\gamma^2 \|d_k^{\mathrm{LP}}\|_{\mathrm{LP}}\, \Delta_k^{\mathrm{LP}} &\geq (1-\eta)\min(\Delta_k^{\mathrm{LP}}, 1)\Psi_k(1) \\ \alpha_k \|d_k^{\mathrm{LP}}\|_{\mathrm{LP}} &\geq \frac{2(1-\eta)\tau}{\beta\gamma^2} \min\left(1, \frac{1}{\Delta_k^{\mathrm{LP}}}\right) \Psi_k(1). \end{aligned} \tag{3.12}$$

Since $\alpha_k d_k^{\mathrm{LP}} = d_k^{\mathrm{C}}$, this inequality combined with (3.10) gives the second inequality in (3.9). □

Our next result is crucial. It provides lower bounds on both the trust-region radius Δ_k and the length of the Cauchy step at a noncritical iterate in the case where the trust-region radius for the linearized problem stays bounded.

LEMMA 3.6. *Suppose Algorithm* 2.1 *is applied to the problem* (1.2) *and that assumptions* P1 *and* P2 *hold. Suppose that* $\{\Delta_k^{LP}\}$ *is bounded above and that* $\Psi_k(1) \geq \delta > 0$ *for all* k. *Then there exists a constant* $\Delta_{min} > 0$ *such that*

$$\Delta_k \geq \Delta_{min} \quad \textit{and} \quad \alpha_k \Delta_k^{LP} \geq \frac{\Delta_{min}}{\gamma} \tag{3.13}$$

for all k.

Proof. By assumption, there exists $\Delta_{\max} \geq 1$ such that

$$\Delta_k^{\mathrm{LP}} \leq \Delta_{\max} \text{ for all } k. \tag{3.14}$$

This inequality, the assumption $\Psi_k(1) \geq \delta$, and Lemma 3.5 imply

$$\|d_k^{\mathrm{C}}\|_{\mathrm{LP}} \geq \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\mathrm{LP}}, \Delta_{\mathrm{crit}}\right), \tag{3.15}$$

where

$$\Delta_{\mathrm{crit}} = \min\left(\frac{1}{\lambda}, \frac{2(1-\eta)\tau}{\beta\gamma^2\Delta_{\max}}\right)\delta. \tag{3.16}$$

If the iteration is successful ($\rho_k \geq \rho_u$), the rule (2.4) for choosing Δ_k^{LP} in step 5 of the algorithm ensures that $\Delta_{k+1}^{\text{LP}} \geq \|d_k^{\text{C}}\|_{\text{LP}}$ and therefore

$$\Delta_{k+1}^{\text{LP}} \geq \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\text{LP}}, \Delta_{\text{crit}}\right). \tag{3.17}$$

Let us now consider the case in which the iteration is unsuccessful. Using Lemma 3.3 and (3.15), we have that

$$\begin{aligned}\phi(x_k) - q_k(d_k) \geq \phi(x_k) - q_k(d_k^{\text{C}}) &\geq \eta\alpha_k \min(\Delta_k^{\text{LP}}, 1)\,\delta = \eta\alpha_k\Delta_k^{\text{LP}} \min\left(\frac{1}{\Delta_k^{\text{LP}}}, 1\right)\delta \\ &\geq \frac{\eta\delta}{\Delta_{\max}}\alpha_k\Delta_k^{\text{LP}} \quad \geq \frac{\eta\delta}{\Delta_{\max}} \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\text{LP}}, \Delta_{\text{crit}}\right).\end{aligned} \tag{3.18}$$

From Lemma 3.4 and (3.18) we have that

$$1 - \rho_k \leq \frac{|\phi(x_k + d_k) - q_k(d_k)|}{\phi(x_k) - q_k(d_k)} \leq \frac{M\|d_k\|^2\Delta_{\max}}{\eta\delta\min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\text{LP}}, \Delta_{\text{crit}}\right)}. \tag{3.19}$$

This implies that $\|d_k\|$ and $(1 - \rho_k)$ are related by the inequality

$$\|d_k\|^2 \geq \frac{(1 - \rho_k)\eta\delta}{M\Delta_{\max}} \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\text{LP}}, \Delta_{\text{crit}}\right) \tag{3.20}$$

at each step. Now, since the iteration is unsuccessful, $\rho_k < \rho_u$ and $1 - \rho_k > 1 - \rho_u$, which, using (2.1) and (3.20), implies

$$\begin{aligned}\theta^2\|d_k\|_{\text{LP}}^2 \geq \frac{\theta^2}{\gamma^2}\|d_k\|^2 &\geq \theta^2\frac{(1 - \rho_u)\eta\delta}{\gamma^2 M\Delta_{\max}} \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\text{LP}}, \Delta_{\text{crit}}\right) \\ &\geq \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\text{LP}}, \Delta_{\text{crit}}, \frac{(1 - \rho_u)\eta\theta^2\delta}{\gamma^2 M\Delta_{\max}}\right)^2.\end{aligned}$$

Using this fact and the lower bound in (2.6), we have that, if the step is unsuccessful,

$$\Delta_{k+1}^{\text{LP}} \geq \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\text{LP}}, \Delta_{\text{crit}}, \frac{(1 - \rho_u)\eta\theta^2\delta}{\gamma^2 M\Delta_{\max}}\right). \tag{3.21}$$

Since the right side of (3.21) is clearly less than or equal to the right side of (3.17), which holds when the step is accepted, then (3.21) must hold at each iteration.

We can consider Δ_k in a similar fashion. If Δ_k was decreased because $\rho_k < \rho_s$, then $1 - \rho_k > 1 - \rho_s$ and (3.20) implies

$$\begin{aligned}\frac{\kappa_l^2}{\gamma^2}\|d_k\|^2 &\geq \frac{(1 - \rho_s)\eta\kappa_l^2\delta}{\gamma^2 M\Delta_{\max}} \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\text{LP}}, \Delta_{\text{crit}}\right) \\ &\geq \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\text{LP}}, \Delta_{\text{crit}}, \frac{(1 - \rho_s)\eta\kappa_l^2\delta}{\gamma^2 M\Delta_{\max}}\right)^2.\end{aligned}$$

Together with (2.3) this implies

$$\frac{\Delta_{k+1}}{\gamma} \geq \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\text{LP}}, \Delta_{\text{crit}}, \frac{(1 - \rho_s)\eta\kappa_l^2\delta}{\gamma^2 M\Delta_{\max}}\right). \tag{3.22}$$

Since Δ_k is not reduced when $\rho_k \geq \rho_s$, (3.22) must then hold at each iteration.

Now we can combine the recursions (3.21) and (3.22) to yield

$$\min\left(\frac{\Delta_{k+1}}{\gamma}, \Delta_{k+1}^{\mathrm{LP}}\right) \geq \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\mathrm{LP}}, \Delta_{\mathrm{crit}}, \frac{(1-\rho_s)\eta\kappa_l^2\delta}{\gamma^2 M\Delta_{\max}}, \frac{(1-\rho_u)\eta\theta^2\delta}{\gamma^2 M\Delta_{\max}}\right), \tag{3.23}$$

which holds at every iteration. Applying this recursion over the entire sequence implies that for all k

$$\begin{aligned}\min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\mathrm{LP}}\right) &\geq \min\left(\frac{\Delta_0}{\gamma}, \Delta_0^{\mathrm{LP}}, \Delta_{\mathrm{crit}}, \frac{(1-\rho_s)\eta\kappa_l^2\delta}{\gamma^2 M\Delta_{\max}}, \frac{(1-\rho_u)\eta\theta^2\delta}{\gamma^2 M\Delta_{\max}}\right)\\ &\equiv \Delta_{\mathrm{low}}.\end{aligned}$$

Thus we can conclude that $\Delta_k \geq \Delta_{\min} \equiv \gamma\Delta_{\mathrm{low}}$ for all k. It then follows from (3.15) and the fact that $\Delta_{\mathrm{low}} \leq \Delta_{\mathrm{crit}}$ that

$$\alpha_k\Delta_k^{\mathrm{LP}} \geq \|d_k^{\mathrm{C}}\|_{\mathrm{LP}} \;\geq \Delta_{\mathrm{low}} = \frac{\Delta_{\min}}{\gamma}$$

for all k. □

This immediately enables us to deduce that if the algorithm is unable to make progress, it must be because it has reached a critical point.

COROLLARY 3.7. *Suppose that assumptions* P1 *and* P2 *hold and that there are only a finite number of iterations for which $\rho_k \geq \rho_u$. Then $x_k = x_*$ for all sufficiently large k, and x_* is a critical point of $\phi(x)$.*

Proof. Step 4 of the algorithm ensures that if there are only a finite number of (successful) iterations for which $\rho_k \geq \rho_u$, then $x_k = x_*$ for all $k > k_0$ for some $k_0 \geq 0$. Moreover, $\Psi_k(1) = \Psi_{k_0}(1)$ for all $k \geq k_0$. Furthermore, as $\rho_k < \rho_u$ for all $k \geq k_0$, the update rules for the trust regions imply that Δ_k converges to zero and Δ_k^{LP} is bounded above for all k. But then $\Psi_k(1) = 0$ for all $k \geq k_0$, since otherwise Lemma 3.6 contradicts the fact that Δ_k converge to zero. It thus follows from Definition 3.1 that x_* is a critical point of ϕ. □

Finally we are able to state our main global convergence result.

THEOREM 3.8. *Suppose Algorithm* 2.1 *is applied to problem* (1.2) *and that assumptions* P1 *and* P2 *hold. If the sequence $\{\phi(x_k)\}$ is bounded below, then either*

$$\Psi_l(1) = 0 \;\; \textit{for some} \;\; l \geq 0$$

or

$$\liminf_{k\to\infty} \Psi_k(1) = 0.$$

Proof. If there are only a finite number of successful iterations, the first of the stated possibilities follows immediately from Corollary 3.7. Otherwise, there is an infinite subsequence $\mathcal{K}$ of successful iterations. This means that $\rho_k \geq \rho_u$, and $\{\phi(x_k)\}$ is bounded from below for all $k \in \mathcal{K}$.

The proof now proceeds by contradiction. Assume there is a constant δ such that $\Psi_k(1) \geq \delta > 0$ for all $k \in \mathcal{K}$. We will consider separately the two cases: when the LP trust-region radius $\{\Delta_k^{\mathrm{LP}}\}$ is bounded above and when $\{\Delta_k^{\mathrm{LP}}\}$ is unbounded.

Case 1. If $\{\Delta_k^{\mathrm{LP}}\}$ is bounded above, it follows from Lemma 3.6 that $\Delta_k \geq \Delta_{\min} > 0$.

For our infinite subsequence $\mathcal{K}$ of successful iterations, Lemmas 3.3 and 3.6 give

$$\begin{aligned}\phi(x_k) - \phi(x_{k+1}) &\geq \rho_u(\phi(x_k) - q_k(d_k)) \\ &\geq \rho_u \eta \alpha_k \min(\Delta_k^{\text{LP}}, 1)\delta \\ &\geq \rho_u \eta \alpha_k \Delta_k^{\text{LP}} \min(1, 1/\Delta_k^{\text{LP}})\delta \\ &\geq \rho_u \eta \Delta_{\min}\delta/(\gamma\Delta_{\max}) > 0,\end{aligned}$$

for all $k \in \mathcal{K}$, where $\Delta_{\max} > 1$ is the upper bound for Δ_k^{LP}. But then summing this inequality over all $k \in \mathcal{K}$ contradicts the fact that the sequence $\{\phi(x_k)\}$ is bounded from below. Thus Case 1 does not occur.

Case 2. Suppose that the LP trust-region radius $\{\Delta_k^{\text{LP}}\}$ is unbounded. Then, since the radius is only increased in step 5 of Algorithm 2.1 when $\alpha_k \geq 1$, there is an infinite sequence $\mathcal{K}$ such that $\Delta_k^{\text{LP}} > 1$, $\alpha_k \geq 1$, and $\rho_k \geq \rho_u$ for all $k \in \mathcal{K}$. Then from Lemma 3.3 we have

$$\begin{aligned}\phi(x_k) - \phi(x_{k+1}) &\geq \rho_u(\phi(x_k) - q_k(d_k)) \\ &\geq \rho_u \eta \alpha_k \min(\Delta_k^{\text{LP}}, 1)\Psi_k(1) \\ &\geq \rho_u \eta \Psi_k(1) \\ &\geq \rho_u \eta \delta,\end{aligned}$$

for all $k \in \mathcal{K}$. This again contradicts the assumption that $\{\phi(x)\}$ is bounded from below, and Case 2 cannot occur.

Cases 1 and 2 therefore imply that the assumption $\Psi_k(1) \geq \delta > 0$ for all k must be false, which proves the desired result

$$\liminf_{k\to\infty} \Psi_k(1) = 0. \qquad \square$$

This result guarantees that if $\phi(x)$ is bounded below, the criticality criterion $\Psi_k(1)$ eventually becomes arbitrarily small. This implies that if the sequence $\{x_k\}$ is bounded, there exists an accumulation point of Algorithm 2.1 which is a critical point for (1.2).

4. A penalty method for nonlinear programming. We now discuss how to automatically adjust the penalty parameter σ as our algorithm proceeds so as to encourage convergence to a critical point of (1.1). We will make use of the following definitions.

We let

$$v(x) = \|h(x)\| + \|g^-(x)\| \tag{4.1}$$

be a measure of constraint violation, so that the penalty function (1.3) can be written as

$$\phi_\sigma(x) = f(x) + \sigma v(x). \tag{4.2}$$

We define a (piecewise) linear model of the constraint violation by

$$\ell^v(x,d) = \|h(x) + \nabla h(x)^T d\| + \|(g(x) + \nabla g(x)^T d)^-\|. \tag{4.3}$$

We can therefore write the model (1.4) of the penalty function as

$$\ell^{\phi_\sigma}(x,d) = f(x) + \nabla f(x)^T d + \sigma \ell^v(x,d). \tag{4.4}$$

Since the penalty parameter σ will now vary, we write the measure of criticality (3.2) for the penalty function as

$$\Psi_\sigma(x, \Delta) = \ell^{\phi_\sigma}(x, 0) - \min_{\|d\| \leq \Delta} \ell^{\phi_\sigma}(x, d). \tag{4.5}$$

Definition 3.1 states that $x_* \in \mathbb{R}^n$ is a critical point of the penalty function ϕ_σ if $\Psi_\sigma(x_*, 1) = 0$. Criticality of the measure of constraint violation $v(x)$ will be measured by the function

$$\theta(x, \Delta) = \ell^v(x, 0) - \min_{\|d\| \leq \Delta} \ell^v(x, d). \tag{4.6}$$

Definition 4.1. *$x_* \in \mathbb{R}^n$ is a critical point of the infeasibility measure $v(x)$ if $\theta(x_*, 1) = 0$.*

It is well known [10] that the penalty function (4.2) is exact in the sense that, for sufficiently large values of σ, strict local minimizers of the nonlinear program (1.1) that satisfy the Mangasarian–Fromovitz constraint qualification (MFCQ) are minimizers of ϕ_σ. We are also interested in the converse result, given that our algorithm minimizes the penalty function.

Theorem 4.1. *If x_* is a critical point of ϕ_σ for some σ and is feasible for* (1.1), *then x_* is a KKT point of the nonlinear program* (1.1). *If x_* is infeasible and is a critical point of ϕ_σ for all sufficiently large σ, then x_* is an infeasible critical point of $v(x)$.*

Proof. At a feasible critical point x_* of ϕ_σ, the vector $d = 0$ minimizes $\ell^{\phi_\sigma}(x_*, d)$, which implies that $d = 0$ is an optimal feasible solution of the linear program

$$\begin{array}{ll} \underset{d}{\text{minimize}} & d^T \nabla f(x_*) \\ \text{subject to} & h(x_*) + \nabla h_i(x_*)^T d = 0, \\ & g(x_*) + \nabla g_i(x_*)^T d \geq 0. \end{array} \tag{4.7}$$

Since the constraints of (4.7) are linear, the KKT conditions for (4.7) are satisfied, and the KKT conditions for (4.7) are identical to the KKT conditions for (1.1).

Suppose that x_* is infeasible and assume, by way of contradiction, that $\theta(x_*, 1) > 0$. Then by (4.6), there exists $\|d^*\| \leq 1$ such that $\ell^v(x_*, 0) - \ell^v(x_*, d^*) > 0$, and therefore for any σ large enough

$$-\nabla f(x_*)^T d^* + \sigma \left(\ell^v(x_*, 0) - \ell^v(x_*, d^*) \right) > 0$$

or

$$\ell^{\phi_\sigma}(x_*, 0) - \ell^{\phi_\sigma}(x_*, d^*) > 0.$$

By (4.4) this implies that $\Psi_\sigma(x_*, 1) > 0$ for arbitrarily large σ, contradicting the assumption that x_* is a critical point for ϕ_σ for all large σ. This contradiction implies that if x_* is infeasible, then $\theta(x_*, 1) = 0$. □

4.1. Penalty update procedure. Our penalty parameter strategy is based on our belief that it is as important to try to decrease the violation $v(x)$ as it is to aim for criticality of $\phi_\sigma(x)$. Since we cannot be sure that there is a (locally) feasible point for the constraints (1.1c)–(1.1b), we might instead measure the quality of the current violation in terms of its criticality, $\theta(x, \Delta)$. Thus we contend it is reasonable to ask that the current value of the penalty parameter σ always satisfies

$$\Psi_\sigma(x_k, 1) \geq \xi \sigma \theta(x_k, 1)$$

for some predefined constant $\xi \in (0,1)$, and to increase the current value if this inequality fails. Hence we cannot consider our iterate to be near a critical point for $\phi_\sigma(x)$ unless it is near a critical point of $v(x)$.

The use of the criticality measure $\Psi_\sigma(x_k, 1)$ requires the solution of an LP with radius 1. Since the algorithm computes the quantity $\Psi_\sigma(x_k, \Delta_k)$ at every iteration, we would instead prefer to use this quantity to estimate criticality, thereby avoiding the extra computational cost of solving a second LP. As we show below, this is possible so long as Δ_k lies within a preset interval $[\delta_{\min}, \delta_{\max}]$. If Δ_k is not in this interval, we will use $\Psi_\sigma(x_k, \delta_k)$ to measure criticality, where δ_k is the closest value to Δ_k in $[\delta_{\min}, \delta_{\max}]$.

Based on this strategy, the set of permissible penalty parameters at an iterate x_k, with trust-region radius Δ_k, is defined as

$$\Omega(x_k, \delta_k) \overset{\text{def}}{=} \{\sigma \mid \Psi_\sigma(x_k, \delta_k) \geq \xi\sigma\theta(x_k, \delta_k)\}.$$

Of course, computation of the quantity $\theta(x_k, \delta_k)$ also involves the solution of an additional LP, but once x_k is near the feasible region, linearized feasibility is likely to be attainable inside the trust region so that $\theta(x_k, \delta_k) = v(x_k)$, and the stronger but easier-to-check condition, $\Psi_\sigma(x_k, \delta_k) \geq \xi\sigma v(x_k)$, will often hold.

We now describe an algorithm for solving the nonlinear programming problem (1.1) in which the penalty parameter is updated at every iteration. It makes use of Algorithm 2.1 to generate steps.

```
ALGORITHM 4.1: PENALTY METHOD FOR SOLVING (1.1)

Initial data: $x_1$, $\sigma_0$. Set the initial parameters of Algorithm 2.1 as well as
$\epsilon > 0$, $0 < \xi < 1$, and $0 < \delta_{\min} \leq \delta_{\max}$.

For $k = 1, 2, \ldots$, until a stopping test for (1.1) is satisfied, perform the following steps.
1.  Let $\delta_k = \mathrm{mid}(\delta_{\min}, \Delta_k^{\mathrm{LP}}, \delta_{\max})$.
        If $\sigma_{k-1} \in \Omega(x_k, \delta_k)$,
            set $\sigma_k = \sigma_{k-1}$.
        Else,
            choose any $\sigma_k \in \Omega(x_k, \delta_k)$ for which $\sigma_k \geq \sigma_{k-1} + \epsilon$.
2.  Perform steps 1–5 of Algorithm 2.1.
```

As was our stated intention, the penalty update strategy in step 1 allows us to (re-)use quantities computed at Δ_k whenever $\Delta_k \in [\delta_{\min}, \delta_{\max}]$. It also ensures that

$$\Psi_{\sigma_k}(x_k, \delta_k) \geq \xi\sigma_k\theta(x_k, \delta_k) \tag{4.8}$$

is satisfied at each iteration, and that if σ is increased, it is because $\sigma_{k-1} \notin \Omega(x_k, \delta_k)$; i.e.,

$$\Psi_{\sigma_{k-1}}(x_k, \delta_k) < \xi\sigma_{k-1}\theta(x_k, \delta_k). \tag{4.9}$$

It is always possible to find a point in $\Omega(x_k, \delta_k)$ for any $\xi < 1$ so that step 1 of Algorithm 4.1 is well defined. To see this, note that definitions (4.1)–(4.6) imply that

$$\Psi_\sigma(x_k, \delta_k) = \sigma v(x_k) - \min_{\|d\| \leq \delta_k} \left(\nabla f(x_k)^T d + \sigma\ell^v(x_k, d)\right) \tag{4.10}$$

$$\begin{aligned}&\geq \sigma v(x_k) - \|\nabla f(x_k)\|\delta_k - \sigma \min_{\|d\|\leq\delta_k} \ell^v(x_k, d)\\ &= -\|\nabla f(x_k)\|\delta_k + \sigma\left(v(x_k) - \min_{\|d\|\leq\delta_k} \ell^v(x_k, d)\right)\\ &= -\|\nabla f(x_k)\|\delta_k + \sigma\theta(x_k, \delta), \end{aligned} \tag{4.11}$$

and thus that $\sigma \in \Omega(x_k, \delta_k)$ for all

$$\sigma \geq \frac{\|\nabla f(x_k)\|\delta_k}{(1-\xi)\theta(x_k, \delta_k)}. \tag{4.12}$$

Notice, however, that it is highly likely that this value will grow without bound if x_k approaches feasibility, so the simpler expedient of always setting σ_k to ensure (4.12) is not to be recommended.

4.2. Penalty method analysis. We begin by recasting the inequality (4.8) in terms of $\Psi_\sigma(x_k, 1)$ instead of $\Psi_\sigma(x_k, \delta_k)$. To do this we recall Lemma 3.1 and note that since the function $\theta(x, \delta)$, like $\Psi(x, \delta)$, is monotonically increasing and concave in δ, the same arguments as used in the proof of Lemma 3.1 imply that

$$\min(\delta_k, 1)\theta(x_k, 1) \leq \theta(x_k, \delta_k) \leq \max(\delta_k, 1)\theta(x_k, 1). \tag{4.13}$$

This then implies the following bound.

LEMMA 4.2. *The values σ_k generated by Algorithm 4.1 satisfy*

$$\Psi_{\sigma_k}(x_k, 1) \geq \xi \min\left(\delta_{\min}, \frac{1}{\delta_{\max}}\right)\sigma_k\theta(x_k, 1). \tag{4.14}$$

Proof. Using (3.6) followed by (4.8) followed by (4.13) yields

$$\Psi_{\sigma_k}(x_k, 1) \geq \frac{\Psi_{\sigma_k}(x_k, \delta_k)}{\max(\delta_k, 1)} \geq \frac{\xi\sigma_k\theta(x_k, \delta_k)}{\max(\delta_k, 1)} \geq \xi\sigma_k\frac{\min(\delta_k, 1)}{\max(\delta_k, 1)}\theta(x_k, 1),$$

which gives (4.14), since $\delta_k \in [\delta_{\min}, \delta_{\max}]$. □

We now present two convergence results for Algorithm 4.1 that rely heavily on the convergence properties of Algorithm 2.1. We first consider the case in which the penalty parameter is updated only a finite number of times.

THEOREM 4.3. *Suppose Algorithm* 4.1 *applied to* (1.1) *generates a bounded sequence of iterates and that assumptions* P1 *and* P2 *hold. If $\{\sigma_k\}$ is bounded, then there is a cluster point x_* of the sequence $\{x_k\}$ which is either a KKT point of the nonlinear program* (1.1) *or a critical point of v.*

Proof. Since $\{\sigma_k\}$ is bounded, it follows from step 1 of Algorithm 4.1 that $\sigma_k = \sigma$ is constant for all large k. Algorithm 4.1 therefore reduces to Algorithm 2.1, i.e., to the minimization of a single penalty function. By Theorem 3.8, if $\{\phi_{\sigma_k}(x_k)\}$ is bounded below, there is a limit point x_* of the sequence of iterates $\{x_k\}$ such that

$$\Psi_\sigma(x_*, 1) = 0. \tag{4.15}$$

If x_* is infeasible, since there is a subsequence $\{x_l\}$ with $\Psi_\sigma(x_l, 1) \to 0$ and since (4.14) holds at each iteration, we must have that $\theta(x_l, 1) \to 0$. Then, since $\theta(\cdot, 1)$ is continuous, $\theta(x_*, 1) = 0$. Therefore, x_* is an infeasible critical point.

If x_* is feasible, i.e., $v(x_*) = 0$, then it follows immediately from Theorem 4.1 that x_* is a KKT point for the nonlinear program (1.1). □

Our final result describes possible outcomes when the penalty parameter is unbounded.

THEOREM 4.4. *Suppose that Algorithm* 4.1 *generates a bounded sequence of iterates* $\{x_k\}$ *and that* $\{\sigma_k\} \to \infty$. *Then either*

(i) *the sequence* $\{x_k\}$ *is not asymptotically feasible (i.e.,* $v(x_k) \not\to 0$*), in which case there is an infeasible cluster point* x_* *that satisfies* $\theta(x_*, 1) = 0$*; or*

(ii) *the sequence* $\{x_k\}$ *is feasible in the sense that* $v(x_k) \to 0$. *In this case, either* (a) *there is a cluster point of* $\{x_k\}$ *that satisfies the KKT conditions, or* (b) *there is a feasible cluster point of* $\{x_k\}$ *at which MFCQ is violated.*

Proof. Consider the sequence of iterates at which the penalty parameter is increased. For each k in this subsequence, condition (4.9) holds, and thus we have

$$\Psi_{\sigma_{k-1}}(x_k, \delta_k) < \xi \sigma_{k-1} \theta(x_k, \delta_k).$$

Now (4.11) holds here, so

$$\Psi_{\sigma_{k-1}}(x_k, \delta_k) \geq -\|\nabla f(x_k)\| \delta_k + \sigma_{k-1} \theta(x_k, \delta_k)$$

and thus, using (4.13),

$$(1-\xi)\sigma_{k-1} \leq \frac{\|\nabla f(x_k)\|\delta_k}{\theta(x_k, \delta_k)} \leq \frac{\|\nabla f(x_k)\|\delta_k}{\theta(x_k, 1)\min(1, \delta_k)} \leq \frac{\|\nabla f(x_k)\|\delta_{\max}}{\theta(x_k, 1)}. \tag{4.16}$$

But as $\{\sigma_k\}$, and consequently $\{\sigma_{k-1}\}$, is assumed unbounded and $\{\nabla f(x_k)\}$ is bounded, it follows that, for that subsequence of $\{x_k\}$ for which σ was increased, we have $\theta(x_k, 1) \to 0$.

If $\limsup v(x_k) > 0$, then, since the sequence $\{x_k\}$ is bounded and $\theta(x_k, 1) \to 0$, there is a limit point with $v(\hat{x}) > 0$ and $\theta(\hat{x}, 1) = 0$; i.e., $\hat{x}$ is an infeasible stationary point of $v(x)$. This implies (i) in that case.

On the other hand, if $\lim v(x_k) = 0$, then there is a cluster point $\hat{x}$ with $v(\hat{x}) = 0$. If $\hat{x}$ satisfies MFCQ, then $\nabla h(\hat{x})$ has full rank and there is a direction $\|d^M\| < \delta_{\min}$ such that

$$\nabla h(\hat{x})^T d^M = 0 = -h(\hat{x}) \text{ and } \nabla g(\hat{x})^T d^M + g(\hat{x}) > 0.$$

Suppose by way of contradiction that $\hat{x}$ is not a KKT point. Then there is a first-order feasible descent direction $\|d^F\| < \delta_{\min}$ such that

$$\nabla h(\hat{x})^T d^F = 0 = -h(\hat{x}), \;\; \nabla g(\hat{x})^T d^F + g(\hat{x}) \geq 0, \;\text{ and }\; \nabla f(\hat{x})^T d^F < 0.$$

Clearly, there is a convex combination $\hat{d} = (1-\alpha)d^F + \alpha d^M$, with $\alpha \in (0,1)$, such that

$$\begin{gathered} \nabla h(\hat{x})^T \hat{d} + h(\hat{x}) = 0, \\ \nabla g(\hat{x})^T \hat{d} + g(\hat{x}) > 0, \;\text{ and }\; \nabla f(\hat{x})^T \hat{d} < 0. \end{gathered} \tag{4.17}$$

Now since $\nabla h(\hat{x})$ has full rank, for any x sufficiently near $\hat{x}$ there is a unique vector $d(x)$ of the form

$$d(x) = \hat{d} + \nabla h(\hat{x}) u(x) \tag{4.18}$$

for some $u(x) \in R^m$, which (nonuniquely) solves

$$h(x) + \nabla h(x)^T d(x) = 0. \tag{4.19}$$

To see this, note that (4.18)–(4.19) imply

$$\left[h(x) + \nabla h(x)^T \hat{d}\right] + \nabla h(x)^T \nabla h(\hat{x}) u(x) = 0. \tag{4.20}$$

Since h is smooth, this equation shows that $u(x)$ is uniquely defined in a neighborhood of $\hat{x}$ and varies continuously with x—and so does $d(x)$. Furthermore, by (4.17) the term in square brackets in (4.20) can be made arbitrarily small if x is close to $\hat{x}$, and hence $d(x)$ is arbitrarily close to $\hat{d}$.

Using these facts we have that $d(x)$ satisfies

$$\nabla g(x)^T d(x) + g(x) > 0, \tag{4.21}$$

$$\nabla f(x)^T d(x) < 0 \tag{4.22}$$

for x sufficiently near $\hat{x}$.

Now note that since $\|d(x)\| < \delta_{\min}$, we have by (4.3), (4.19), and (4.21) that $\ell^v(x, d(x)) = 0$. By the nonnegativity of $\ell^v(x, d(x))$ and the definition (4.6), this implies that $\theta(x) = \ell^v(x, 0) = v(x)$. In addition, since $\nabla f(x)^T d(x) < 0$, we have from (4.10) that

$$\Psi_\sigma(x, \delta) > \sigma v(x) = \sigma \theta(x, \delta)$$

for any $\delta \geq \delta_{\min}$. Therefore, for any iterate x_k sufficiently near $\hat{x}$, $\sigma \in \Omega(x_k, \delta_k)$ for all $\sigma \geq 0$. As a result, for this subsequence of iterates, σ is never updated in a neighborhood of $\hat{x}$.

This argument applies to any feasible limit point that satisfies MFCQ. Therefore, it is not possible for all such points to have a descent direction, for otherwise the penalty parameter would be updated only a finite number of times, contradicting the assumption that $\sigma_k \to \infty$. In other words, we cannot have that all limit points satisfy MFCQ and are not KKT points. This proves (ii). □

Thus we are able to embed a relatively simple penalty-parameter update scheme *within* Algorithm 2.1 and derive useful convergence results. Another possibility which could be tried is to update the penalty parameter as needed once a globally convergent method has approximately minimized ϕ_σ with the current (fixed) σ. Rules to achieve this are known [4, 11], but we are concerned that this may prove to be inefficient, particularly when an inappropriate initial σ is specified.

In the current version of the exact penalty method SLIQUE [1], the penalty parameter is updated by a procedure that requires $\sigma_k \in \Omega(x_k, \delta_k)$ at each iteration, as well as some further conditions. Therefore, Theorem 4.3 essentially holds for SLIQUE.[3] However, because of the additional conditions on σ_k, it is not clear whether a result like Theorem 4.4 can be proved for SLIQUE.

5. Conclusions and perspectives. In this paper we have proposed a trust-region algorithm for nonlinear optimization that uses a combination of linear and quadratic model steps and has separate, quasi-autonomous trust regions to control these. At least one subsequence generated by the algorithm is shown to be globally convergent to a critical point of the problem under modest assumptions. Our framework for trust-region radius updates is deliberately general. This is because we wished it to apply in the case of the current implementation of our evolving nonlinear programming code SLIQUE [1] as well as to cover its future evolution.

[3] This is true of the current implementation, but the description in [1] differs in some minor details.

We have not considered the ultimate convergence rate of the algorithm, nor its ability to identify the optimal active constraints in a finite number of iterations (these two aspects are most likely closely linked [8]), although we have strong numerical evidence to suggest that the latter does occur and that the convergence rate may thereafter be made to be superlinear. The study of these and other issues is ongoing.

Acknowledgment. The authors are grateful to two anonymous referees for their helpful comments on this paper.

REFERENCES

[1] R. H. Byrd, N. I. M. Gould, J. Nocedal, and R. A. Waltz, *An active set algorithm for nonlinear programming using linear programming and equality constrained subproblems*, Math. Program., 100 (2004), pp. 27–48.

[2] C. M. Chin, *A New Trust Region Based SLP-Filter Algorithm Which Uses EQP Active-Set Strategy*, Ph.D. thesis, University of Dundee, Scotland, 2001.

[3] C. M. Chin and R. Fletcher, *On the global convergence of an SLP-filter algorithm that takes EQP steps*, Math. Program., 96 (2003), pp. 161–177.

[4] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust Region Methods*, MPS-SIAM Ser. Optim. 1, SIAM, Philadelphia, 2000.

[5] A. R. Conn and T. Pietrzykowski, *A penalty function method converging directly to a constrained optimum*, SIAM J. Numer. Anal., 14 (1977), pp. 348–375.

[6] R. Fletcher, *Non-differential optimization*, Chapter 14 in Practical Methods of Optimization: Constrained Optimization, Vol. 2, John Wiley and Sons, Chichester, New York, 1981.

[7] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Math. Program., 91 (2002), pp. 239–269.

[8] R. Fletcher and E. Sainz de la Maza, *Nonlinear programming and non-smooth optimization by successive linear programming*, Math. Program., 43 (1989), pp. 235–256.

[9] R. W. Gate, *Development of Algorithms for Solving Large Optimization Problems*, Ph.D. thesis, University of Dundee, Scotland, 2004.

[10] S. P. Han and O. L. Mangasarian, *Exact penalty functions in nonlinear programming*, Math. Program., 17 (1979), pp. 251–269.

[11] D. Q. Mayne and E. Polak, *Feasible directions algorithms for optimisation problems with equality and inequality constraints*, Math. Program., 11 (1976), pp. 67–80.

[12] T. Pietrzykowski, *An exact potential method for constrained maxima*, SIAM J. Numer. Anal., 6 (1969), pp. 299–304.

[13] E. Sainz de la Maza, *Nonlinear Programming Algorithms Based on ℓ_1 Linear Programming and Reduced Hessian Approximation*, Ph.D. thesis, University of Dundee, Scotland, 1987.

[14] R. A. Waltz, *Algorithms for Large-Scale Nonlinear Optimization*, Ph.D. thesis, Northwestern University, Evanston, IL, 2002.

[15] Y. Yuan, *Conditions for convergence of trust region algorithms for non-smooth optimization*, Math. Program., 31 (1985), pp. 220–228.

[16] Y. Yuan, *On the convergence of a new trust region algorithm*, Numer. Math., 70 (1995), pp. 515–539.

SIAM J. Optim. © 2005 Society for Industrial and Applied Mathematics
Vol. 16, No. 2, pp. 490–514

SEMIDEFINITE APPROXIMATIONS FOR GLOBAL UNCONSTRAINED POLYNOMIAL OPTIMIZATION*

DORINA JIBETEAN† AND MONIQUE LAURENT‡

Abstract. We consider the problem of minimizing a polynomial function on $\mathbb{R}^n$, known to be hard even for degree 4 polynomials. Therefore approximation algorithms are of interest. Lasserre [*SIAM J. Optim.*, 11 (2001), pp. 796–817] and Parrilo [*Math. Program.*, 96 (2003), pp. 293–320] have proposed approximating the minimum of the original problem using a hierarchy of lower bounds obtained via semidefinite programming relaxations. We propose here a method for computing tight upper bounds based on perturbing the original polynomial and using semidefinite programming. The method is applied to several examples.

Key words. semidefinite programming, global optimization, approximation algorithm, positive polynomial, sum of squares of polynomials, moment matrix

AMS subject classifications. 90C22, 90C26, 49M20

DOI. 10.1137/04060562X

1. Introduction.

We consider the problem

$$p^* := \inf_{x\in\mathbb{R}^n} p(x) \tag{1}$$

of minimizing a polynomial p in n indeterminates over $\mathbb{R}^n$. We may assume that p has an even degree $2m$, since otherwise $p^* = -\infty$. There are three possibilities: Either p has an infinite infimum (i.e., $p^* = -\infty$), p has a finite infimum (e.g., for the polynomial $p(x_1, x_2) = x_1^2 + (x_1x_2 - 1)^2$), or p has a minimum. Computing the infimum of a polynomial is a hard problem, already for degree 4 polynomials. Indeed, it contains the problem of deciding whether a matrix is copositive, which is known to be co-NP-hard [21], with an $n \times n$ matrix P being copositive if $p(x) := \sum_{i,j=1}^n P_{ij}x_i^2x_j^2 \geq 0$ for all $x \in \mathbb{R}^n$, i.e., if $p^* = 0$. Alternatively, problem (1) contains the problem of deciding whether an integer sequence $a_1, \dots, a_n$ can be partitioned, which is known to be NP-complete [7], with $a_1, \dots, a_n$ being partitionable if there exists $x \in \{\pm1\}^n$ such that $a^Tx = 0$, i.e., if the infimum of the polynomial $p(x) := (a^Tx)^2 + \sum_{i=1}^n (x_i^2 - 1)^2$ is equal to 0.

1.1. Some known approaches to polynomial unconstrained minimization.

An approach followed by some authors (e.g., by Hägglöf, Lindberg, and Stevenson [8]) is to look at the first order conditions $\partial p/\partial x_i = 0$ $(i = 1, \dots, n)$. Various algebraic techniques can be used for determining the real solutions to this system of polynomial equations; e.g., using Gröbner bases and the eigenvalue method, using resultants and discriminants, or homotopy methods (see, e.g., [3]; see [25] for a discussion and comparison). However, there are several difficulties with such an approach. It is computationally expensive (e.g., computing a Gröbner basis may be computationally very demanding), the number of critical points can be infinite, and, moreover,

*Received by the editors March 25, 2004; accepted for publication (in revised form) April 22, 2005; published electronically December 8, 2005. This work was supported by the Netherlands Organization for Scientific Research through grant NWO 639.032.203.

http://www.siam.org/journals/siopt/16-2/60562.html

†TUe, Postbus 513, 5600 MB Eindhoven, The Netherlands (djibetean@win.tue.nl).

‡CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands (monique@cwi.nl).

this approach applies *only* if the polynomial p attains its minimum. We will come back to this type of approach later in this section.

Hanzon and Jibetean [9] (see also Jibetean [12]) proposed going around these difficulties by considering a perturbation

$$p_\lambda(x) := p(x) + \lambda \left(\sum_{i=1}^{n} x_i^{2m+2} \right) \tag{2}$$

of the original polynomial p for small $\lambda > 0$. Set

$$p_\lambda^* := \inf_{x \in \mathbb{R}^n} p_\lambda(x).$$

Thus, $p^* \leq p_\lambda^* \leq p^* + \lambda \|x^*\|^{2m+2}$ if x^* is a global minimizer of p. The perturbed polynomial has the following properties: p_λ attains its minimum, the set of critical points of p_λ is finite, and the limit of the minima p_λ^* as $\lambda \to 0$ is equal to the infimum p^* of p. Moreover, if p has a global minimum, then the limit set as $\lambda \downarrow 0$ of the set of global minimizers of p_λ is contained in the set of global minimizers of p, and each connected component of the set of global minimizers of p contains a point which is the limit of a branch of local minimizers of p_λ. Exploiting these facts, Hanzon and Jibetean proposed an exact algorithm for computing the limit p^* of the minima p_λ^* as well as a global minimizer of p (if some exist). Their algorithm uses algebraic techniques, some of them closely related to the algebraic machinery developed by Basu, Pollack, and Roy [1]. Hanzon and Jibetean's method applies to any polynomial p, i.e., no assumption is made on the existence of a minimum. However, its computational cost is very high and the algorithm can be applied in practice only to small instances.

Another type of approach consists of solving a convex (in fact, semidefinite) relaxation of the original problem; see, e.g., Lasserre [15], Parrilo [22, 23], and Shor [28]. The approach applies more generally to the problem

$$p^* := \inf_{x \in K} p(x), \text{ where } K := \{x \in \mathbb{R}^n \mid h_1(x) \geq 0, \ldots, h_\ell(x) \geq 0\} \tag{3}$$

of minimizing p over a set defined by polynomial inequalities and equations (treating an equation $h(x) = 0$ as two opposite inequalities: $h(x) \geq 0$, $-h(x) \geq 0$). Following Lasserre [15], set $d_i := \lceil \deg(h_i)/2 \rceil$ and, for an integer $k \geq \max(\lceil \deg(p)/2 \rceil, d_1, \ldots, d_\ell)$, consider the semidefinite program

$$p_{L,k}^* := \inf \ p^T y \ \text{ s.t. } M_k(y) \succeq 0, \ M_{k-d_i}(h_i y) \succeq 0 \ (i = 1, \ldots, \ell), \ y_0 = 1 \tag{4}$$

(the *moment relaxation of order* k of (3)), and its dual

$$\begin{array}{ll} \rho_k^* := \sup \ \rho \ \text{ s.t.} & p(x) - \rho = u_0 + \sum_{i=1}^{\ell} u_i h_i, \text{ where} \\ & u_0, u_1, \ldots, u_\ell \text{ are sum of squares of polynomials} \\ & \text{and } \deg(u_0), \deg(u_1 h_1), \ldots, \deg(u_\ell h_\ell) \leq 2k \end{array} \tag{5}$$

(the *sum of squares relaxation of order* k of (3)). Program (4) uses the variables $y = (y_\alpha)_{\alpha \in S_{2k}}$, $M_k(y) := (y_{\alpha+\alpha'})_{\alpha,\alpha' \in S_k}$ is the moment matrix of order k, $M_{k-d_i}(h_i y)$ are localizing matrices, and for an integer k, we set $S_k := \{\alpha \in \mathbb{Z}_+^n \mid |\alpha| := \sum_{i=1}^n \alpha_i \leq k\}$. Then $\rho_k^* \leq p_{L,k}^* \leq p^*$, $\rho_k^* \leq \rho_{k+1}^*$, and $p_{L,k}^* \leq p_{L,k+1}^*$. Under some assumption on K, there is asymptotic convergence of the parameters ρ_k^*, μ_k^* to p^*. The following cases are of particular interest for our purpose:

(I) $K = \{x \in \mathbb{R}^n \mid \sum_{i=1}^n x_i^2 \leq R^2\}$. Then there is asymptotic convergence of ρ_k^* and $p_{L,k}^*$ to p^* (see [15]).

(II) $K = \{x \in \mathbb{R}^n \mid h_1(x) = 0, \ldots, h_\ell(x) = 0\}$ and the polynomials $h_1, \ldots, h_\ell$ generate a zero-dimensional ideal I (i.e., they have finitely many common complex zeros). Then there is finite convergence of $p_{L,k}^*$ to p^*, and of ρ_k^* when $h_1, \ldots, h_\ell$ form a Gröbner basis of I (see [18]) or when I is radical (see [24]).

(III) $K = \{x \in \mathbb{R}^n \mid \frac{\partial p}{\partial x_i}(x) = 0 \ (i = 1, \ldots, n)\}$. Then there is asymptotic convergence of ρ_k^* and p_k^* to p^*, and finite convergence when the ideal I_{grad} generated by the polynomials $\frac{\partial p}{\partial x_i}$ $(i = 1, \ldots, n)$ is radical (see [6]). (By case (II) there is finite convergence of p_k^* to p^* when I_{grad} is zero-dimensional.)

Henrion and Lasserre [11] gave the following stopping criterion: If the optimum solution y to (4) satisfies the rank condition

$$\text{rank}\, M_k(y) = \text{rank}\, M_{k-d}(y), \text{ where } d := \max(d_1, \ldots, d_\ell), \tag{6}$$

then $p_k^* = p^*$. See section 2.2 for details.

For our original unconstrained minimization problem (1) (then $\ell = 0$ and $K = \mathbb{R}^n$), we have $p_{L,k}^* = p_{L,m}^* \leq p^*$ for all $k \geq m$, with equality $p_{L,m}^* = p^*$ if and only if $p - p^*$ is a sum of squares. One possible option to better approximate p^* is to transform the unconstrained problem (1) into a constrained problem of the form (3). This is possible if p attains its minimum, as p^* can then be formulated as

$$p^* = p_{grad}^* := \inf\ p(x) \text{ s.t. } \partial p(x)/\partial x_i = 0 \ (i = 1, \ldots, n). \tag{7}$$

The equality $p^* = p_{grad}^*$ does not hold in general if p does not attain its minimum; for instance, $p^* = 0$ and $p_{grad}^* = 1$ for $p(x_1, x_2) = x_1^2 + (x_1 x_2 - 1)^2$; $p^* = -\infty$ and $p_{grad}^* = 0$ for $p(x) = x^3$. If p has a minimum and if some upper bound R is known a priori on the norm of a global minimizer, then p^* can also be expressed as

$$p^* = \min\ p(x) \text{ s.t. } \sum_{i=1}^n x_i^2 \leq R^2. \tag{8}$$

A major drawback of approaches based on formulations like (7) or (8) is that it is not clear how to test whether a polynomial has a minimum and, for (8), how to find a ball containing a global minimizer. We will, however, present in section 2.1 a result of Marshall [19] concerning a class of polynomials for which such a ball can be determined beforehand.

1.2. Our approach. In this paper we propose the following strategy for getting around these difficulties. Following Hanzon and Jibetean [9], we consider the perturbed polynomial p_λ from (2). As computing the exact limit p^* of the minima p_λ^* is not a realistic option for large problems, we work toward the less ambitious goal of computing a good upper approximation p_λ^* of p^* for some small value of λ. As mentioned earlier, the polynomial p_λ enjoys several properties (that p may not have in general). Namely, p_λ attains its minimum, which can thus be formulated as

$$p_\lambda^* = \min_{x \in V_\lambda \cap \mathbb{R}^n} p_\lambda(x), \tag{9}$$

where

$$V_\lambda := \left\{ x \in \mathbb{C}^n \mid h_{\lambda,i}(x) := \frac{\partial p_\lambda}{\partial x_i}(x) = 0 \ (i = 1, \ldots, n) \right\},$$

and the set V_λ is finite ($|V_\lambda| \leq (2m+1)^n$). Moreover, one can give an explicit radius

$$R_\lambda = \frac{n^m}{\lambda} \sum_{\alpha \neq 0} |p_\alpha| \tag{10}$$

for a ball containing the global minima of p_λ (see Corollary 3); thus

$$p^*_\lambda = \min_{x \in B_\lambda} \; p_\lambda(x), \tag{11}$$

where

$$B_\lambda := \left\{ x \in \mathbb{R}^n \mid h_{\lambda,0}(x) := R_\lambda^2 - \sum_{i=1}^n x_i^2 \geq 0 \right\}.$$

By minimizing $p(x)$ over the algebraic set $V_\lambda \cap \mathbb{R}^n$ or over the ball B_λ, one obtains even better bounds μ^*_λ and β^*_λ, respectively; that is,

$$p^* \leq \mu^*_\lambda := \min_{x \in V_\lambda \cap \mathbb{R}^n} p(x) \leq p^*_\lambda, \quad p^* \leq \beta^*_\lambda := \min_{x \in B_\lambda} p(x) \leq p^*_\lambda.$$

As the parameters μ^*_λ and β^*_λ are expressed via constrained polynomial programs of the form (3), a first option is to apply Lasserre's approach for computing them. Namely, for any integer $k \geq m+1$, consider the programs

$$\begin{aligned} \mu^*_{L,k,\lambda} := \inf \; p^T y \;\; \text{s.t.} \;\; & y_0 = 1, \; M_k(y) \succeq 0, \\ & M_{k-m-1}(h_{\lambda,i} y) = 0 \; (i = 1, \ldots, n), \end{aligned} \tag{12}$$

$$\beta^*_{L,k,\lambda} := \inf \; p^T y \;\; \text{s.t.} \;\; y_0 = 1, \; M_k(y) \succeq 0, \; M_{k-1}(h_{\lambda,0} y) \succeq 0. \tag{13}$$

Then

$$\mu^*_{L,k,\lambda} \leq \mu^*_{L,k+1,\lambda} \leq \mu^*_\lambda, \quad \beta^*_{L,k,\lambda} \leq \beta^*_{L,k+1,\lambda} \leq \beta^*_\lambda \;\text{ for } k \geq m+1.$$

As k goes to infinity, there is asymptotic convergence of $\beta^*_{L,k,\lambda}$ to β^*_λ (recall case (I)) and finite convergence of the parameters $\mu^*_{L,k,\lambda}$ to μ^*_λ (recall case (II)).

As the set V_λ is finite, another option for computing the bound μ^*_λ is to apply the semidefinite representation result for finite varieties of Laurent [18]. Namely, μ^*_λ can be expressed as the optimum of the semidefinite program

$$\mu^*_\lambda = \min p^T y \;\; \text{s.t.} \;\; M_{\mathcal{B}}(y) \succeq 0, \; y_0 = 1, \tag{14}$$

involving a combinatorial moment matrix $M_{\mathcal{B}}(y)$. Here, $y = (y_\beta)_{\beta \in \mathcal{B}} \in \mathbb{R}^{\mathcal{B}}$, where

$$\mathcal{B} := \{ \beta \in \mathbb{Z}^n \mid 0 \leq \beta_i \leq 2m \; (i = 1, \ldots, n) \}$$

has the property that the set of monomials $\{x^\beta \mid \beta \in \mathcal{B}\}$ forms a basis of the space $\mathbb{R}[x_1, \ldots, x_n]/I_\lambda$, and I_λ is the ideal generated by $h_{\lambda,i} = \partial p_\lambda / \partial x_i$ $(i = 1, \ldots, n)$. The matrix $M_{\mathcal{B}}(y)$ is obtained from a classical moment matrix by "factoring" through I_λ, which, roughly speaking, means that the equations $h_{\lambda,i}(x) = 0$ are used for expressing any y_α ($\alpha \in \mathbb{Z}^n_+$) in terms of y_β ($\beta \in \mathcal{B}$). As a by-product, this implies the finite convergence of the bounds $\mu^*_{L,k,\lambda}$ from (12) to μ^*_λ; more precisely, $\mu^*_{L,k,\lambda} = \mu^*_\lambda$ for $k \geq 2nm$ (by Theorem 23 in [18]).

The semidefinite program (14) is more compact than (12) (for any k ensuring finite convergence). Indeed, program (14) involves only one linear matrix inequality (LMI) and $|\mathcal{B}| = (2m+1)^n$ variables, whereas (12) involves $n+1$ LMIs and $\binom{n+2k}{2k}$ variables. Moreover the size of the matrix $M_{\mathcal{B}}(y)$ is $|\mathcal{B}| = (2m+1)^n$, which is smaller than the size $\binom{n+k}{k}$ of the matrix $M_k(y)$ for any $k \geq 2nm$. Solving the semidefinite program (14) is, however, still out of reach for large n or m. Moreover, the entries of $M_{\mathcal{B}}(y)$ are polynomial in $1/\lambda$ (and linear in y) and thus, for λ close to 0, they may be ill-conditioned. These difficulties can be addressed in the following way. Given an integer k, $m \leq k \leq 2nm$, consider the truncated semidefinite program obtained by considering the principal submatrix of $M_{\mathcal{B}}(y)$, denoted $M_{\mathcal{B}_k}(y)$, indexed by the subset $\mathcal{B}_k := \mathcal{B} \cap S_k$, and set

$$\mu^*_{k,\lambda} := \inf p^T y \text{ s.t. } M_{\mathcal{B}_k}(y) \succeq 0, y_0 = 1. \tag{15}$$

Thus,

$$\mu^*_{k,\lambda} \leq \mu^*_{k+1,\lambda} \leq \mu^*_{2nm,\lambda} = \mu^*_\lambda.$$

When the optimum solution $M_{\mathcal{B}_k}(y)$ satisfies the following rank condition

$$\operatorname{rank} M_{\mathcal{B}_h}(y) = \operatorname{rank} M_{\mathcal{B}_{h-1}}(y) \tag{16}$$

for some $m \leq h \leq k$, one can conclude that the optimum value of the truncated problem (15) is an upper bound for the infimum p^*; that is, $p^* \leq \mu^*_{k,\lambda} \leq \mu^*_\lambda$. Moreover, one can extract a point x for which $p^* \leq p(x) \leq \mu^*_{k,\lambda}$, thus giving a certificate for the claimed upper bound $\mu^*_{k,\lambda}$ on p^* (see Corollary 19). In this way, one is (often) able to compute a very good upper approximation of p^* by solving a much smaller semidefinite program. Moreover the degree in $1/\lambda$ of the entries of $M_{\mathcal{B}_k}(y)$ is at most $k-m$ (see Theorem 18) and thus remains small for small values of k. Several examples illustrating this procedure are given in section 3.2. In most cases one is able to conclude that the parameter $\mu^*_{k,\lambda}$ from program (15) is an upper bound for p^* already for $k = m+1$ or $m+2$, in which case the entries of $M_{\mathcal{B}_k}(y)$ are at most quadratic in $1/\lambda$, and we are thus able to carry out the computations for a small perturbation parameter $\lambda \sim 10^{-4}$ and sometimes even smaller. By the results of [9], for such small λ, the extracted minimizer x_λ is very close to a global minimizer of p (if some exist); this will be verified in the examples.

Given an integer $k \geq m$, program (15) can be seen as a "compact" analogue of program (12). We can prove the following interlacing property for their optimal values (see Theorem 17):

$$\mu^*_{k,\lambda} \leq \mu^*_{L,k+1,\lambda} \leq \mu^*_{k+1,\lambda} \tag{17}$$

for $m \leq k \leq 2nm$, with equality $\mu^*_{2nm,\lambda} = \mu^*_{L,2nm,\lambda} = \mu^*_\lambda$; see Examples 4, 5, 6 for a numerical comparison. Program (12) involves matrices of size $|S_k| = \binom{n+k}{k}$ and $|S_{2k}| = \binom{n+2k}{2k}$ variables, whereas its compact analogue (15) involves matrices of size $|\mathcal{B}_k| = |S_k \cap \mathcal{B}|$ and $|\mathcal{B}_{2k}| = |S_{2k} \cap \mathcal{B}|$ variables. For $k \leq 2m$, $\mathcal{B}_k = S_k$, but $\mathcal{B}_{2k}$ is then already significantly smaller than S_{2k}. This is illustrated in Table 1, which displays some values of $|S_{2k} \setminus \mathcal{B}_{2k}| = |S_{2k} \setminus \mathcal{B}|$ for $k = m+1, m+2$.

1.3. Contents of the paper.

The paper is organized as follows. Section 2 contains preliminaries about polynomials and about classical and combinatorial moment matrices and their application to polynomial optimization. In section 3, we present

Table 1
Gain in number of variables when using program (15) *instead of program* (12).

	$\lvert S_{2m+2} \setminus \mathcal{B}\rvert$	$\lvert S_{2m+4} \setminus \mathcal{B}\rvert$ for $m \geq 2$	$\lvert S_{2m+4} \setminus \mathcal{B}\rvert$ for $m = 1$
n	$n(n+1)$	$4n + 12\binom{n}{2} + 12\binom{n}{3} + 4\binom{n}{4}$	$4n + 11\binom{n}{2} + 12\binom{n}{3} + 4\binom{n}{4}$
$n = 2$	6	20	19
$n = 3$	12	60	57
$n = 4$	20	140	136
$n = 5$	30	280	275
$n = 10$	110	2860	2850

our method for computing the upper approximations μ_λ^* for the infimum p^* of a polynomial p over $\mathbb{R}^n$, and in section 3.2 we present several examples on which our method has been tested.

2. Preliminaries.

2.1. Polynomials.

We begin with some preliminaries on ideals of polynomials. Throughout the paper, $\mathbb{R}[x_1, \ldots, x_n]$ denotes the ring of real polynomials in n indeterminates. For an integer $k \geq 0$, S_k denotes the set of $\alpha \in \mathbb{Z}_+^n$ with $|\alpha| := \sum_{i=1}^n \alpha_i \leq k$. Write a polynomial $p \in \mathbb{R}[x_1, \ldots, x_n]$ with (total) degree at most k as $p(x) = \sum_{\alpha \in S_k} p_\alpha x^\alpha$, where x^α denotes the monomial $x^\alpha := x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. As usual, we identify a polynomial p of degree at most k with the sequence of its coefficients $p = (p_\alpha)_{\alpha \in S_k}$.

Let I be an ideal in $\mathbb{R}[x_1, \ldots, x_n]$. The set

$$V = V(I) := \{x \in \mathbb{C}^n \mid f(x) = 0 \; \forall f \in I\}$$

is its associated (complex) variety. The ideal I is said to be *zero-dimensional* if $|V| < \infty$. The sets $I(V) := \{f \in \mathbb{R}[x_1, \ldots, x_n] \mid f(v) = 0 \; \forall v \in V\}$ and $\sqrt{I} := \{f \in \mathbb{R}[x_1, \ldots, x_n] \mid f^k \in I \text{ for some integer } k \geq 1\}$ are again ideals in $\mathbb{R}[x_1, \ldots, x_n]$, which obviously contain the ideal I. The Nullstellensatz asserts that these two ideals coincide; namely, $\sqrt{I} = I(V)$. The ideal I is said to be *radical* when $I = \sqrt{I}$. Hence, by the Nullstellensatz,

$$I \text{ is radical} \iff \begin{array}{l}\text{the polynomials vanishing at all points of } V \\ \text{are precisely the polynomials in } I.\end{array} \tag{18}$$

The following result, relating the dimension of the quotient vector space $\mathbb{R}[x_1, \ldots, x_n]/I$ and the cardinality of V, can be found, e.g., in [2, section 5.3]:

$$\begin{gathered} |V| < \infty \iff \dim \mathbb{R}[x_1, \ldots, x_n]/I < \infty, \\ |V| \leq \dim\ \mathbb{R}[x_1, \ldots, x_n]/I, \text{ with equality if and only if } I \text{ is radical.} \end{gathered} \tag{19}$$

We now recall a result of Marshall [19] giving a sufficient condition for a polynomial to have a minimum. Given a nonzero polynomial p, let $\tilde{p}$ be its *highest degree homogeneous component*, defined as the sum of the terms of p having maximum degree, and set

$$\tilde{p}_S := \min_{x \in S} \tilde{p}(x), \quad \text{where } S := \left\{ x \in \mathbb{R}^n \,\middle|\, \sum_{I=1}^n x_i^2 = 1 \right\}.$$

If $\tilde{p}_S < 0$, then p has obviously an infinite infimum, i.e., $p^* = -\infty$. If $\tilde{p}_S > 0$, then, following Marshall [19], p is said to be *stably bounded from below* and, as the next result shows, p attains its minimum. On the other hand, no conclusion can be drawn when $\tilde{p}_S = 0$; indeed, p may have an infinite infimum (e.g., for $p(x_1, x_2) = x_1^2 + x_2$), a finite infimum (e.g., for $p(x_1, x_2) = x_1^2 + (x_1x_2 - 1)^2$), or a minimum (e.g., for $p(x_1, x_2) = x_1^2x_2^2$).

LEMMA 1 (see [19]). *Assume p is stably bounded from below. Given $x \in \mathbb{R}^n$,*

$$p(x) \leq 0 \Longrightarrow \|x\| \leq \max\left(\frac{1}{\tilde{p}_S} \sum_{\alpha:|\alpha|\leq \deg(p)-1} |p_\alpha|, 1\right). \tag{20}$$

In particular, any global minimum of p belongs to the ball centered at the origin with radius $R_p := \max(1, \frac{1}{\tilde{p}_S} \sum_{\alpha:1\leq|\alpha|\leq\deg(p)-1} |p_\alpha|)$.

Proof. Say p has degree d and write $p = \tilde{p} + g$, where all terms of $\tilde{p}$ have degree d and all terms of g have degree $\leq d - 1$. Let $x \in \mathbb{R}^n \setminus \{0\}$ such that $p(x) \leq 0$. Thus, $\tilde{p}(x) \leq -g(x) \leq \sum_{\alpha:|\alpha|\leq d-1} |p_\alpha||x^\alpha|$. By assumption, $\tilde{p}(x) = \|x\|^d\tilde{p}(\frac{x}{\|x\|}) \geq \|x\|^d\tilde{p}_S > 0$. On the other hand, if $\|x\| \geq 1$ and $|\alpha| \leq d - 1$, then $|x^\alpha| \leq \|x\|^{|\alpha|} \leq \|x\|^{d-1}$. Combining these two facts, we find the relation (20). If x is a global minimum of p, then $p(x) \leq p(0)$ and thus $\|x\| \leq R_p$ follows from (20) applied to the polynomial $p - p(0)$. □

In general, the polynomial p may not be stably bounded from below and it may not even have a minimum. However, for any positive λ, the perturbed polynomial p_λ is stably bounded from below. Indeed, if p has degree $2m$, then the highest degree homogeneous component of p_λ is equal to $\lambda \sum_{i=1}^n x_i^{2m+2}$, whose minimum value over the unit sphere is equal to $\frac{\lambda}{n^m}$ as the next lemma shows.

LEMMA 2. *Given an integer $m \geq 2$, the minimum value taken by $\sum_{i=1}^n x_i^{2m}$ over the unit sphere is equal to $\frac{1}{n^{m-1}}$.*

Proof. By evaluating $f(x) := \sum_i x_i^{2m}$ at the point $x := \frac{1}{\sqrt{n}}(1, \ldots, 1)$, we find that the minimum value f_S of f over the unit sphere is at most $\frac{1}{n^{m-1}}$. To show the reverse inequality, note that f_S is equal to the minimum value of $g(x) := \sum_{i=1}^n x_i^m$ over $x \in \mathbb{R}_+^n$ with $\sum_{i=1}^n x_i = 1$. Let x be a minimizer to this program. Applying the Karush–Kuhn–Tucker conditions, there exist $\lambda \in \mathbb{R}$, $z \in \mathbb{R}_+^n$ such that $\nabla g(x) - \lambda e - z = 0$ and $x^T z = 0$. As $x, z \geq 0$, $x_iz_i = 0$ for all i and $\frac{\partial g}{\partial x_i}(x) = \lambda$ if $z_i = 0$. Say, $z_1 = \cdots = z_p = 0$, $z_{p+1}, \ldots, z_n > 0$ for some $p \leq n$; thus $x_{p+1} = \cdots = x_n = 0$. For $i = 1, \ldots, p$, $\frac{\partial g}{\partial x_i}(x) = \lambda = mx_i^{m-1}$. From this follows that $x_1 = \cdots = x_p = \frac{1}{p}$. Now, $g(x) = \frac{1}{p^{m-1}} \geq \frac{1}{n^{m-1}}$ as $p \leq n$. □

COROLLARY 3. *Given a polynomial p of degree $2m$, the global minima of the perturbed polynomial $p_\lambda(x) = p(x) + \lambda(\sum_{i=1}^n x_i^{2m+2})$ are located in the ball B_λ with radius $R_\lambda := \frac{n^m}{\lambda} \sum_{\alpha\neq 0} |p_\alpha|$.*

2.2. Moment matrices. We recall here some results about moment matrices that we need in the paper. Given a probability measure μ on $\mathbb{R}^n$, the quantity $y_\alpha := \int x^\alpha \mu(dx)$ is called its *moment of order* α. A probability measure with finite support is of the form $\mu = \sum_{i=1}^r \lambda_i\delta_{x_i}$, where $\lambda_i > 0$, $\sum_{i=1}^r \lambda_i = 1$, $x_i \in \mathbb{R}^n$ (the *atoms* of the measure); then μ is said to be r*-atomic*. Here, δ_x is the Dirac measure at $x \in \mathbb{R}^n$, having mass 1 at x and mass 0 elsewhere.

The moment problem concerns the characterization of the sequences $y \in \mathbb{R}^{S_{2k}}$ ($k \geq 1$) that are the sequences of moments of some probability measure μ; in that case one also says that μ is a representing measure for y. Given $y \in \mathbb{R}^{S_{2k}}$, its *moment matrix of order* k is the matrix $M_k(y)$ indexed by S_k with (α, β)th entry $y_{\alpha+\beta}$ for

$\alpha, \beta \in S_k$. Given a polynomial $h(x)$ of degree $2d$ or $2d-1$, define the vector hy with entries $(hy)_\alpha := \sum_\gamma h_\gamma y_{\alpha+\gamma}$ for $\alpha \in S_{2k-2d}$; $M_{k-d}(hy)$ is known as a *localizing moment matrix.* A well-known necessary condition for the existence of a representing measure for y is the positive semidefiniteness of its moment matrix.

LEMMA 4. *If $y \in \mathbb{R}^{S_{2k}}$ has a representing measure μ, then $M_k(y) \succeq 0$. Moreover, if the support of μ is contained in the set $\{x \mid h(x) \geq 0\}$, where $h(x)$ is a polynomial of degree $2d$ or $2d-1$, then $M_{k-d}(hy) \succeq 0$.*

Proof. For $p \in \mathbb{R}^{S_k}$, we have

$$p^T M_k(y) p = \sum_{\alpha,\beta \in S_k} p_\alpha p_\beta y_{\alpha+\beta} = \sum_{\alpha,\beta \in S_k} p_\alpha p_\beta \int x^{\alpha+\beta} d\mu(x) = \int p(x)^2 d\mu(x) \geq 0,$$

which shows that $M_k(y) \succeq 0$. If the support of μ is contained in $\{x \mid h(x) \geq 0\}$, one can verify that $p^T M_{k-d}(hy) p = \int p(x)^2 h(x) d\mu(x) \geq 0$ for all $p \in \mathbb{R}^{S_{k-d}}$, which shows that $M_{k-d}(hy) \succeq 0$. □

Curto and Fialkow [4, 5] prove some results showing that, under some rank condition, the necessary conditions from the above lemma are also sufficient for the existence of a representing measure. A key notion is that of "flat extension." Let X be a symmetric matrix and let A be a principal submatrix of X. One says that X is a *flat extension* of A if rank $X =$ rank A. Then $X \succeq 0 \Longleftrightarrow A \succeq 0$.

THEOREM 5 (see [4]). *Let $y \in \mathbb{R}^{S_{2k}}$. If $M_k(y) \succeq 0$ and $M_k(y)$ is a flat extension of $M_{k-1}(y)$, then y has a representing measure which is* (rank $M_k(y)$)*-atomic.*

The proof uses the following property of the kernel of $M_k(y)$, which also permits one to derive Corollary 7 below.

LEMMA 6 (see [4]). *Assume that $M_k(y) \succeq 0$ and let $f, g \in \mathbb{R}[x_1, \dots, x_n]$, whose product $h := fg$ has degree* $\deg(h) \leq k-1$. *Then $M_k(y)f = 0$ implies $M_k(y)h = 0$.*

COROLLARY 7. *If $M_k(y) \succeq 0$ and* rank $M_h(y) =$ rank $M_{h-1}(y)$ *for some $1 \leq h \leq k-1$, then* rank $M_{k-1}(y) =$ rank $M_{k-2}(y)$.

THEOREM 8 (see [5]; see [17] for a short proof). *Let $y \in \mathbb{R}^{S_{2k}}$, $h_1, \dots, h_\ell \in \mathbb{R}[x_1, \dots, x_n]$, $d_i := \lceil \deg(h_i)/2 \rceil$, and $d := \max(d_1, \dots, d_\ell)$. Assume that $M_k(y) \succeq 0$, $M_{k-d_i}(h_i y) \succeq 0$ (for $i = 1, \dots, \ell$), and* rank $M_k(y) =$ rank $M_{k-d}(y)$. *Then y has a representing measure μ supported by the set $\{x \mid h_1(x) \geq 0, \dots, h_\ell(x) \geq 0\}$; moreover μ is* (rank $M_k(y)$)*-atomic.*

The above results underlie the semidefinite relaxations (4) and (5) of problem (3). In particular, as an application of Theorem 8, one finds the stopping criterion of Henrion and Lasserre [11]: If $M_k(y)$ is an optimum solution to (4) satisfying the rank condition (6), then $p^*_{L,k} = p^*$. This is a very useful fact, as it permits one very often in practice to conclude that the relaxation (4) of a given order k solves the original problem (3) at optimality for small values of k. The following two results imply the asymptotic (or finite) convergence of the parameters ρ^*_k and $p^*_{L,k}$ to the optimum p^* in cases (I) and (III) mentioned in section 1.1.

THEOREM 9 (see [26]). *Let $K = \{x \in \mathbb{R}^n \mid h_1(x) \geq 0, \dots, h_\ell(x) \geq 0\}$ and $M := \{u_0 + \sum_{i=1}^\ell u_i h_i \mid u_0, u_1, \dots, u_\ell$ are sums of squares of polynomials$\}$. Assume that K is compact and that there exists a polynomial $u \in M$ for which the set $\{x \in \mathbb{R}^n \mid u(x) \geq 0\}$ is compact. Then every positive polynomial on K belongs to M.*

THEOREM 10 (see [6]). *Given a polynomial p, define $K := \{x \in \mathbb{R}^n \mid \frac{\partial p}{\partial x_i}(x) = 0\ (i = 1, \dots, n)\}$ and let I_{grad} be the ideal generated by $\partial p/\partial x_i$ $(i = 1, \dots, n)$. If p is positive on K, then p is a sum of squares of polynomials modulo I_{grad}. When I_{grad} is radical, the same conclusion holds if p is nonnegative on K.*

2.3. Combinatorial moment matrices. Let I be a zero-dimensional ideal in $\mathbb{R}[x_1,\dots,x_n]$ with $V = V(I)$ as associated complex variety. With respect to a given monomial ordering, let G be a Gröbner basis of I and let $\mathcal{S}$ be the associated set of standard monomials consisting of the monomials that are not divisible by the leading term of any polynomial in G. Let $\mathcal{B}$ be the set of exponents of the standard monomials; that is, $\mathcal{S} = \{x^\beta \mid \beta \in \mathcal{B}\}$. The set $\mathcal{S}$ is a basis of $\mathbb{R}[x_1,\dots,x_n]/I$; that is, for every polynomial $f \in \mathbb{R}[x_1,\dots,x_n]$, there exists a unique polynomial $r(x) = \sum_{\beta\in\mathcal{B}} r_\beta x^\beta$ for which $f - r \in I$; r is called the *residue* of f modulo I.

Given $y = (y_\beta)_{\beta\in\mathcal{B}} \in \mathbb{R}^{\mathcal{B}}$, let $M_{\mathcal{B}}(y)$ be the $\mathcal{B}\times\mathcal{B}$ matrix whose (α,β)th entry is equal to $\sum_{\gamma\in\mathcal{B}} r_\gamma y_\gamma$ for $\alpha,\beta\in\mathcal{B}$, where $\sum_{\gamma\in\mathcal{B}} r_\gamma x^\gamma$ is the residue of $x^\alpha x^\beta$ modulo I; $M_{\mathcal{B}}(y)$ is called the *combinatorial moment matrix* of y. In other words, $M_{\mathcal{B}}(y)$ is obtained from a classical moment matrix by expressing all entries of y in terms of those indexed by the standard monomials using the equations defining I. For $v \in \mathbb{R}^n$, define the vector $\zeta_v := (v^\beta)_{\beta\in\mathcal{B}} \in \mathbb{R}^{\mathcal{B}}$. It is not difficult to check that if $v \in V \cap \mathbb{R}^n$, then $M_{\mathcal{B}}(\zeta_v) = \zeta_v\zeta_v^T$ is positive semidefinite. Hence, $M_{\mathcal{B}}(y) \succeq 0$ if y belongs to the cone generated by the vectors ζ_v ($v \in V\cap\mathbb{R}^n$). Laurent [18] shows that equivalence holds.

THEOREM 11 (see [18]). *Let I be a zero-dimensional ideal in $\mathbb{R}[x_1,\dots,x_n]$, let V be the associated variety, and let $\{x^\beta \mid \beta\in\mathcal{B}\}$ be the set of standard monomials with respect to some monomial ordering. Let $y \in \mathbb{R}^{\mathcal{B}}$ and let $M_{\mathcal{B}}(y)$ be its associated combinatorial moment matrix. Then $M_{\mathcal{B}}(y) \succeq 0$ if and only if y belongs to the cone generated by ζ_v ($v \in V\cap\mathbb{R}^n$); that is, y is the sequence of moments (of order $\alpha\in\mathcal{B}$) of a nonnegative atomic measure μ whose support is contained in $V\cap\mathbb{R}^n$.*

2.4. Truncated combinatorial moment matrices. We assume in this section and the next one that the ideal I is generated by n polynomials of the form

$$h_i(x) := x_i^{2m+1} - \tilde{h}_i(x) \quad \text{for } i = 1,\dots,n, \tag{21}$$

where $\deg(\tilde{h}_i) \leq 2m$ and $m \geq 1$ is a given integer. In that case, we can prove some results about flat extensions of truncated combinatorial moment matrices, which will be useful for our application to optimization.

The polynomials $h_1,\dots,h_n$ form a Gröbner basis of the ideal I (with respect to a total degree monomial ordering) (apply [2, section 2.6]). Therefore, the set of standard monomials is $\mathcal{S} = \{x^\beta \mid \beta\in\mathcal{B}\}$, where

$$\mathcal{B} := \{\beta\in\mathbb{Z}^n \mid 0 \leq \beta_i \leq 2m \ \forall i = 1,\dots,n\}. \tag{22}$$

It follows from (19) that the ideal I is zero-dimensional. Given an integer $1 \leq k \leq 2nm$, define

$$\mathcal{B}_k := \mathcal{B}\cap S_k = \{\beta\in\mathcal{B} \mid |\beta| \leq k\}. \tag{23}$$

LEMMA 12. *Given $f \in \mathbb{R}[x_1,\dots,x_n]$, let r be its residue modulo I. Then* $\deg(r) \leq \deg(f)$.

Proof. Fix a total degree monomial ordering. Then the division algorithm applied for dividing f by $h_1,\dots,h_n$ yields a decomposition $f = \sum_{i=1}^n u_i h_i + r$, where $r(x) = \sum_{\beta\in\mathcal{B}} r_\beta x^\beta$ is the residue of f, and $\deg(u_i h_i) \leq \deg(f)$ whenever $u_i \neq 0$ (see [2, section 2.3]). Therefore, $\deg(r) \leq \deg(f)$. □

For a monomial x^α, let $r^{(\alpha)}(x)$ denote its residue modulo I; by Lemma 12, $r^{(\alpha)}(x)$ is of the form $r^{(\alpha)}(x) = \sum_{\beta\in\mathcal{B}_k} r_\beta^{(\alpha)} x^\beta$ if $|\alpha| \leq k$. Therefore, given a truncated sequence $y \in \mathbb{R}^{\mathcal{B}_{2k}}$, one can define its truncated combinatorial moment matrix $M_{\mathcal{B}_k}(y)$ as the matrix indexed by $\mathcal{B}_k$ whose (α, β)th entry is $y^T r^{(\alpha+\beta)}$ for $\alpha, \beta \in \mathcal{B}_k$. We now indicate how to extend a combinatorial moment matrix to a classical moment matrix.

DEFINITION 13. *Given $y \in \mathbb{R}^{\mathcal{B}_{2k}}$, extend y to $\tilde{y} \in \mathbb{R}^{S_{2k}}$ by setting*

$$\tilde{y}_\gamma := y^T r^{(\gamma)} \text{ for } \gamma \in S_{2k}, \tag{24}$$

where $r^{(\gamma)}(x)$ is the residue of x^γ modulo I.

LEMMA 14. *Let $y \in \mathbb{R}^{\mathcal{B}_{2k}}$, with $\tilde{y} \in \mathbb{R}^{S_{2k}}$ its extension from* (24). *Let f be a polynomial of degree at most $2k$ and r its residue modulo I. Then $f^T\tilde{y} = r^T y$.*

Proof. Using (24), we find that $f^T\tilde{y} = \sum_\delta f_\delta \tilde{y}_\delta = \sum_\delta f_\delta y^T r^{(\delta)} = \sum_{\beta,\delta} f_\delta r_\beta^{(\delta)} y_\beta$, while $y^T r = \sum_\beta r_\beta y_\beta$. Hence it suffices to show that the two polynomials $r(x)$ and $s(x) := \sum_{\beta,\delta} f_\delta r_\beta^{(\delta)} x^\beta$ are identical. For this, note that $s(x) = \sum_\delta f_\delta r^{(\delta)}(x) \equiv \sum_\delta f_\delta x^\delta = f(x) \equiv r(x)$ modulo I. Hence, $r = s$, since both r and s are polynomials using only standard monomials. □

LEMMA 15. *Let $y \in \mathbb{R}^{\mathcal{B}_{2k}}$, with $\tilde{y} \in \mathbb{R}^{S_{2k}}$ its extension from* (24). *Then $M_k(\tilde{y})$ is a flat extension of $M_{\mathcal{B}_k}(y)$.*

Proof. By the definition of $\tilde{y}$, the principal submatrix of $M_k(\tilde{y})$ indexed by $\mathcal{B}_k$ coincides with $M_{\mathcal{B}_k}(y)$. Consider a column C_γ of $M_k(\tilde{y})$ indexed by $\gamma \in S_k \setminus \mathcal{B}_k$. We verify that $C_\gamma = \sum_{\beta\in\mathcal{B}_k} r_\beta^{(\gamma)} C_\beta$; that is,

$$\tilde{y}_{\alpha+\gamma} = \sum_{\beta\in\mathcal{B}_k} r_\beta^{(\gamma)} \tilde{y}_{\alpha+\beta} \ \forall \alpha \in S_k.$$

For this consider the polynomial $f(x) := x^{\alpha+\gamma} - \sum_{\beta\in\mathcal{B}_k} r_\beta^{(\gamma)} x^{\alpha+\beta}$. As f has degree at most $2k$ and $f \in I$, it follows from Lemma 14 that $f^T\tilde{y} = 0$, which gives the desired relation. □

COROLLARY 16. *Let $y \in \mathbb{R}^{\mathcal{B}_{2k}}$, with $\tilde{y} \in \mathbb{R}^{S_{2k}}$ its extension from* (24). *Assume that $M_{\mathcal{B}_h}(y)$ is a flat extension of $M_{\mathcal{B}_{h-1}}(y)$ for some $1 \le h \le k$. (Then this holds for $h = k$ or $k - 1$ by Corollary 7.) Then $(\tilde{y}_\alpha)_{\alpha\in S_{2h}}$ (and thus $(y_\alpha)_{\alpha\in\mathcal{B}_{2h}}$) is the sequence of moments of an r-atomic measure μ, where $r :=$ rank $M_{\mathcal{B}_h}(y)$. Moreover, if $h \ge 2m+1$, then the support of μ is contained in V.*

Proof. By Lemma 15, $M_h(\tilde{y})$ is a flat extension of $M_{h-1}(\tilde{y})$. Hence, by Theorem 5, $(y_\alpha)_{\alpha\in S_{2h}}$ has an r-atomic representing measure μ, where $r = \text{rank } M_h(\tilde{y}) = \text{rank } M_{\mathcal{B}_h}(y)$. If $h \ge 2m+1$, then the polynomials $h_i(x)$ $(i = 1, \dots, n)$ generating the ideal I belong to the kernel of $M_h(\tilde{y})$ (by the construction of $\tilde{y}$). Hence, the support of μ is contained in the set of common zeros of the h_i's, i.e., in the variety V. □

2.5. Optimization and extraction of solutions. Given a polynomial $p \in \mathbb{R}[x_1, \dots, x_n]$, consider the problem

$$p^* := \min\ p(x) \text{ s.t. } h_1(x) = 0, \dots, h_n(x) = 0,$$

where $h_1, \dots, h_n$ are as in (21). We can assume that p has degree at most $2m$; otherwise replace p by its residue modulo the ideal I. We first compare the following two hierarchies of lower bounds for p^*, defined for $k \ge m$:

$$\mu_k^* := \inf p^T y \text{ s.t. } M_{\mathcal{B}_k}(y) \succeq 0,\ y_0 = 1, \tag{25}$$

$$\begin{aligned} \mu_{L,k}^* := \inf p^T y \text{ s.t. } & y_0 = 1,\ M_k(y) \succeq 0, \\ & M_{k-m-1}(h_i y) = 0\ (i = 1, \dots, n), \end{aligned} \tag{26}$$

where we omit the condition $M_{k-m-1}(h_i y) = 0$ when $k = m$. If $k = m$, the two programs (25) and (26) are identical and thus $\mu_m^* = \mu_{L,m}^*$. Moreover, by Theorem 11 (and Theorem 23 in [18]), $\mu_{2nm}^* = \mu_{L,2nm}^* = p^*$. One can show the following interlacing property for the parameters μ_k^* and $\mu_{L,k}^*$, which implies the interlacing property (17) for the two hierarchies of bounds from (12) and (15).

THEOREM 17. *$\mu_{k-1}^* \leq \mu_{L,k}^* \leq \mu_k^*$ for all $k \geq m+1$.*

Proof. Let z be a feasible solution to (26), i.e., $M_k(z) \succeq 0$, $M_{k-m-1}(h_i z) = 0$, $z_0 = 1$. We observe first that $f^T z = 0$ for every polynomial $f \in I$ with degree at most $2k-1$. Indeed, as $f \in I$, $f = \sum_{i=1}^n u_i h_i$, where $\deg(u_i h_i) \leq \deg(f) \leq 2k-1$, i.e., $\deg(u_i) \leq 2k-1-(2m+1) = 2k-2m-2$ whenever $u_i \neq 0$. Moreover, $f(x) = \sum_{i=1}^n \sum_{\gamma,\delta} (u_i)_\gamma (h_i)_\delta x^{\gamma+\delta}$. Hence,

$$f^T z = \sum_\beta f_\beta z_\beta = \sum_\beta z_\beta \sum_{i=1}^n \sum_{\gamma,\delta|\gamma+\delta=\beta} (u_i)_\gamma (h_i)_\delta = \sum_{i=1}^n \sum_\gamma (u_i)_\gamma \sum_\delta (h_i)_\delta z_{\gamma+\delta}.$$

Now, $\sum_\delta (h_i)_\delta z_{\gamma+\delta} = (h_i z)_\gamma = 0$ since $|\gamma| \leq \deg(u_i) \leq 2k-2m-2$ and $M_{k-m-1}(h_i z) = 0$. Therefore, we find that $f^T z = 0$. Hence, if we denote by y the restriction of z to $\mathbb{R}^{\mathcal{B}_{2k}}$, then $z_\gamma = y^T r^{(\gamma)}$ for $|\gamma| \leq 2k-1$. Hence $M_{\mathcal{B}_{k-1}}(y)$ coincides with the principal submatrix of $M_k(z)$ indexed by $\mathcal{B}_{k-1}$ and thus $M_{\mathcal{B}_{k-1}}(y) \succeq 0$. This implies that $p^T z = p^T y \geq \mu_{k-1}^*$ and thus $\mu_{L,k}^* \geq \mu_{k-1}^*$.

Consider now a feasible solution $y \in \mathbb{R}^{\mathcal{B}_{2k}}$ to (25). Let $\tilde{y}$ be its extension to $\mathbb{R}^{S_{2k}}$ from (24). Then $M_k(\tilde{y}) \succeq 0$ by Lemma 15. It remains to verify that $M_{k-m-1}(h_i \tilde{y}) = 0$, i.e., that $(h_i \tilde{y})_\alpha = \sum_\gamma (h_i)_\gamma \tilde{y}_{\alpha+\gamma}$ is equal to 0 for $|\alpha| \leq 2k-2m-2$. As the polynomial $f(x) := h_i(x) x^\alpha$ belongs to I and its degree is at most $2k$, it follows from Lemma 14 that $f^T \tilde{y} = 0$, which gives the desired relation. Hence, $\tilde{y}$ is feasible for (26), which implies that $p^T y = p^T \tilde{y} \geq \mu_{L,k}^*$ and thus $\mu_k^* \geq \mu_{L,k}^*$. $\square$

Let y be an optimum solution to (25). Assume that $\operatorname{rank} M_{\mathcal{B}_h}(y) = \operatorname{rank} M_{\mathcal{B}_{h-1}}(y) =: r$ for some $1 \leq h \leq k$. By Corollary 16, $(y_\beta)_{\beta \in \mathcal{B}_{2h}}$ is the sequence of moments of a measure $\mu = \sum_{i=1}^r \lambda_i \delta_{v_i}$ ($\lambda_i > 0$, $\sum_i \lambda_i = 1$, $v_i \in \mathbb{R}^n$). If $h \geq m$, then $p^* \geq \mu_k^* = p^T y = \sum_i \lambda_i p(v_i) \geq \min_i p(v_i)$; moreover, $v_1, \ldots, v_r$ belong to $V(I)$ and thus are global minimizers of p over the set $\{x \in \mathbb{R}^n \mid h_1(x) = \cdots = h_n(x) = 0\}$ when $h \geq 2m+1$. We now indicate how to extract the points $v_1, \ldots, v_r$ from the matrix $M_{\mathcal{B}_h}(y)$; this is analogous to the extraction procedure in [11] (for program (26)).

As $\operatorname{rank} M_{\mathcal{B}_h}(y) = \operatorname{rank} M_{\mathcal{B}_{h-1}}(y) = r$, one can find a subset $\mathcal{A}$ of $\mathcal{B}_{h-1}$, $|\mathcal{A}| = r$, indexing a positive definite principal submatrix A of $M_{\mathcal{B}_h}(y)$. If $h \leq 2m$, let J denote the ideal generated by the kernel of $M_{\mathcal{B}_h}(y)$ and, if $h \geq 2m+1$, let J be the ideal generated by I and the kernel of $M_{\mathcal{B}_h}(y)$. Obviously, $\{v_1, \ldots, v_r\} \subseteq V(J)$. On the other hand, $\mathcal{A}$ is a basis of $\mathbb{R}[x_1, \ldots, x_n]/J$ (easy to verify) and thus $\dim \mathbb{R}[x_1, \ldots, x_n]/J = r$, which implies that $|V(J)| \leq r$ (by (19)). Therefore, $V(J) = \{v_1, \ldots, v_r\}$ and J is a zero-dimensional radical ideal. Thus, determining $v_1, \ldots, v_r$ amounts to finding the common zeros to the polynomials in J, which can be done with the eigenvalue method, briefly described below (see, e.g., [3, Chapter 2, section 4]).

For a polynomial f, the *multiplication matrix* M_f is the $|\mathcal{A}| \times |\mathcal{A}|$ matrix whose αth column (for $\alpha \in \mathcal{A}$) contains the coefficients in the base $\mathcal{A}$ of the residue modulo J of the polynomial $x^\alpha f(x)$. If f is chosen in such a way that the values $f(v)$ are distinct for $v \in V(J)$, then the right eigenspaces of M_f are one-dimensional and spanned by the vectors $(v^\alpha)_{\alpha \in \mathcal{A}}$ (for $v \in V(J)$) (Proposition 4.7 in [3]). Hence, the points $v_1, \ldots, v_r$ of $V(J)$ can be determined from the right eigenvectors of M_f.

In our extraction procedure, we construct the base $\mathcal{A}$ in a "greedy manner"; starting from the constant monomial 1, we insert in $\mathcal{A}$ as many low degree monomials as possible. Then, given an eigenvector $(v^\alpha)_{\alpha\in\mathcal{A}}$ (or a scalar multiple of it), it is easy to recover the components of v (in fact, immediate, if $\mathcal{A}$ contains the monomials $x_1,\dots,x_n$). We determine the multiplication matrices M_{x_i} (for $f = x_i$, $i = 1,\dots,n$) in the following way. As before let A be the principal submatrix of $M_{\mathcal{B}_h}(y)$ indexed by $\mathcal{A}$ and let U_i be the submatrix of $M_{\mathcal{B}_h}(y)$ with row indices $\mathcal{A}$ and column indices the set $x_i\mathcal{A} := \{x_i x^\alpha \mid \alpha \in \mathcal{A}\}$. When $h \le 2m$ (which is the case considered for practical applications), $M_{x_i} = A^{-1}U_i$. (Indeed, given $\beta \in \mathcal{A}$, let v be the column of $M_{\mathcal{B}_h}(y)$ indexed by $x_i x^\beta$, $u := (v_\alpha)_{\alpha\in\mathcal{A}}$ the corresponding column of U_i, and $c = (c_\alpha)_{\alpha\in\mathcal{A}}$ the unique scalars permitting to express v as $v = \sum_{\alpha\in\mathcal{A}} c_\alpha C_\alpha$, with C_α being the column of $M_{\mathcal{B}_h}(y)$ indexed by x^α. Then $c = A^{-1}u$ and the polynomial $x_i x^\beta - \sum_{\alpha\in\mathcal{A}} c_\alpha x^\alpha$ belongs to the ideal J generated by the kernel of $M_{\mathcal{B}_h}(y)$. Thus $\sum_{\alpha\in\mathcal{A}} c_\alpha x^\alpha$ is the residue of $x_i x^\beta$ modulo J; i.e., c is the corresponding column of M_{x_i}.) Then, for an arbitrary polynomial f, its multiplication matrix M_f is given by $M_f = f(M_{x_1},\dots,M_{x_n})$, whose eigenvectors can be used for extracting the global optimizers.

Let us make a comment at this point. For solving our original problem of minimizing p over the set of real points in $V(I)$, one could use the following strategy: Determine all points in $V(I)$ (using the eigenvalue method) and evaluate p at the real points. This is, however, computationally expensive, as this involves computing the eigenvalues of a multiplication matrix whose size is $|\mathcal{B}| = (2m+1)^n$, thus exponential in the number of variables. Instead, we propose to solve the relaxed convex program (25) for small values of k. Typically it has an optimum solution of small rank r and, when the rank condition holds, one can extract a solution by computing the eigenvalues of a much smaller matrix of size r.

3. Application to unconstrained polynomial minimization.

3.1. Our method.

Let us return to the problem (1) of computing the infimum p^* of a polynomial p over $\mathbb{R}^n$. As before, we assume that p has degree $2m$ and, for $\lambda > 0$, we consider the perturbed polynomial p_λ as in (2) and set $p^*_\lambda := \inf_{x\in\mathbb{R}^n} p_\lambda(x)$. For $i = 1,\dots,n$, let

$$h_{\lambda,i}(x) := \partial p_\lambda(x)/\partial x_i = \partial p(x)/\partial x_i + \lambda(2m+2)x_i^{2m+1} \tag{27}$$

denote the partial derivatives of $p_\lambda(x)$. Let I_λ be the ideal generated by $h_{\lambda,1},\dots,h_{\lambda,n}$ and let $V_\lambda := V(I_\lambda)$ be its associated variety. Up to a constant factor, each $h_{\lambda,i}(x)$ is of the form $x_i^{2m+1} + \tilde{h}_i(x)$, where $\tilde{h}_i(x)$ has degree at most $2m-1$, and thus we are in the situation of section 2.4. Therefore, for $\lambda \ne 0$, the set $\{x^\beta \mid \beta \in \mathcal{B}\}$, where $\mathcal{B}$ is as in (22), is the set of standard monomials, forming a basis of $\mathbb{R}[x_1,\dots,x_n]/I_\lambda$, and I_λ is a zero-dimensional ideal.

As p_λ attains its minimum, it follows that it attains its minimum at a critical point. That is, $\inf_{x\in\mathbb{R}^n} p_\lambda(x) = \min_{x\in V_\lambda\cap\mathbb{R}^n} p_\lambda(x)$. If x^* is a global minimizer of p, then $p^* \le p^*_\lambda \le p_\lambda(x^*) \le p^* + \lambda\|x^*\|^{2m+2}$. As $p(x) \le p_\lambda(x)$ for all x, we have

$$p^* \le \mu^*_\lambda := \min_{x\in V_\lambda\cap\mathbb{R}^n} p(x) \le \min_{x\in V_\lambda\cap\mathbb{R}^n} p_\lambda(x).$$

As I_λ is a zero-dimensional ideal, we can apply Theorem 11 and compute the bound μ^*_λ via the following semidefinite program:

$$\mu^*_\lambda = \min p^T y \;\text{ s.t. }\; M_{\mathcal{B}}(y) \succeq 0,\; y_0 = 1, \tag{28}$$

where $\mathcal{B}$ is defined in (22). Given an integer $m \leq k \leq 2nm$, one can consider the following semidefinite program involving truncated combinatorial moment matrices:

$$\mu^*_{k,\lambda} := \inf p^T y \ \text{ s.t. } \ M_{\mathcal{B}_k}(y) \succeq 0, \ y_0 = 1, \tag{29}$$

where $\mathcal{B}_k$ is as in (23). These parameters define a hierarchy of lower bounds for μ^*_λ,

$$\mu^*_{m,\lambda} \leq \cdots \leq \mu^*_{k,\lambda} \leq \cdots \leq \mu^*_{2nm,\lambda} = \mu^*_\lambda, \tag{30}$$

where the last equality holds since $\mathcal{B}_{2nm} = \mathcal{B}$.

Let us give some information about the structure of the matrix $M_{\mathcal{B}_k}(y)$. For $\alpha, \beta \in \mathcal{B}_k$, the (α, β)th entry of $M_{\mathcal{B}_k}(y)$ is equal to $y^T r^{(\alpha+\beta)}$, where $r^{(\alpha+\beta)}(x)$ is the residue of $x^{\alpha+\beta}$ modulo the ideal I_λ. This residue is obtained by dividing the monomial $x^{\alpha+\beta}$ by the polynomials $h_{\lambda,i}$ from (27), forming a Gröbner basis of I_λ. Hence, the entries of $M_{\mathcal{B}_k}(y)$ are polynomial in $1/\lambda$ (and linear in y). The next result gives an estimate on the degree in $1/\lambda$ of the entries of $M_{\mathcal{B}_k}(y)$.

THEOREM 18. *For $k = m, \ldots, 2nm$, the matrices $M_{\mathcal{B}_k}(y)$ are polynomial matrices in $1/\lambda$; the maximal degree in $1/\lambda$ of the entries of $M_{\mathcal{B}_k}(y)$ is at most $k - m$.*

Proof. Consider a monomial x^γ where $\gamma \in \mathbb{Z}^n_+$ with $|\gamma| \geq 2m$. We show by induction on $|\gamma|$ that the coefficients of the residue of x^γ modulo the ideal I_λ are polynomial in $1/\lambda$ with degree at most $\lceil (|\gamma| - 2m)/2 \rceil$. If $\gamma_i \leq 2m$ for all $i = 1, \ldots, n$, then x^γ is a standard monomial; that is, its residue is x^γ whose degree in $1/\lambda$ is 0. Suppose, e.g., that $\gamma_1 \geq 2m + 1$. Then $x^\gamma = x_1^{2m+1} x^{\tilde{\gamma}}$, where $\tilde{\gamma}_1 = \gamma_1 - 2m - 1$ and $\tilde{\gamma}_i = \gamma_i$ for $i \geq 2$. Thus, $|\tilde{\gamma}| = |\gamma| - 2m - 1$ and $x^\gamma \equiv -\frac{1}{2m+2} \frac{1}{\lambda} \frac{\partial p(x)}{\partial x_1} x^{\tilde{\gamma}}$ modulo I_λ. As the degree of $x^{\tilde{\gamma}} \partial p(x)/\partial x_1$ is at most $2m - 1 + |\tilde{\gamma}| = |\gamma| - 2$, we know by induction that the degree in $1/\lambda$ of its residue is at most $\lceil (|\gamma| - 2 - 2m)/2 \rceil = \lceil (|\gamma| - 2m)/2 \rceil - 1$. Therefore, the degree in $1/\lambda$ of the residue of x^γ is at most $\lceil (|\gamma| - 2m)/2 \rceil$. The theorem now follows since each entry of $M_{\mathcal{B}_k}(y)$ is the residue of a monomial of degree at most $2k$. ◻

As $M_{\mathcal{B}_m}(y)$ does not depend on λ, the matrix $M_{\mathcal{B}_m}(y)$ coincides with the classical matrix $M_m(y)$. Hence, the first member $\mu^*_{m,\lambda}$ in the hierarchy (30) does not depend on λ and is equal to $p^*_{L,m}$, the Lasserre lower bound for p^* from (4); thus,

$$\mu^*_{m,\lambda} = p^*_{L,m} \leq p^*.$$

It is not clear a priori on which side of p^* the parameter $\mu^*_{k,\lambda}$ is located when $m < k < 2nm$. In some cases, one can derive this information with the help of the following result.

COROLLARY 19. *Let $M_{\mathcal{B}_k}(y)$ be an optimum solution to program* (29) *defining $\mu^*_{k,\lambda}$. Assume that* rank $M_{\mathcal{B}_h}(y) =$ rank $M_{\mathcal{B}_{h-1}}(y)$ *for some $m \leq h \leq k$. Then $p^* \leq \mu^*_{k,\lambda} \leq \mu^*_\lambda$ and one can extract a point $x \in \mathbb{R}^n$ for which $p^* \leq p(x) \leq \mu^*_{k,\lambda}$. Moreover, $\mu^*_{k,\lambda} = \mu^*_\lambda$ if $h \geq 2m + 1$.*

Proof. By Corollary 16, $(y_\alpha)_{\alpha \in \mathcal{B}_{2h}}$ is the sequence of moments of a probability measure $\mu = \sum_{i=1}^r \lambda_i \delta_{v_i}$. Hence, $\mu^*_{k,\lambda} = p^T y = \sum_{i=1}^r \lambda_i p(v_i) \geq \min_i p(v_i) \geq p^*$. If $h \geq 2m + 1$, then $v_1, \ldots, v_r \in V_\lambda$ and thus $\mu^*_{k,\lambda} = \mu^*_\lambda$. ◻

Let us point out that, for the problem of computing the minimum p^* of a polynomial of the form $p = \sum_{i=1}^n c_i x_i^{2m+2} + p_0$ where $\deg p_0 \leq 2m+1$ ($c_1, \ldots, c_n \in \mathbb{R} \setminus \{0\}$), our method can be applied directly to p, without any perturbation. Namely, let r be the residue of p modulo the ideal generated by $\partial p/\partial x_i$ ($i = 1, \ldots, n$); then

$$p^* = \min \ r^T y \ \text{ s.t. } \ M_{\mathcal{B}}(y) \succeq 0, \ y_0 = 1,$$

and the parameters μ_k^* from (25) are lower bounds for p^*, with equality $\mu_k^* = p^*$ if $\operatorname{rank} M_{\mathcal{B}_h}(y) = \operatorname{rank} M_{\mathcal{B}_{h-1}}(y)$ for some $2m+1 \le h \le k$. See Examples 9 and 10 in the next section for an illustration.

We now illustrate our method on two small examples; both will be revisited in the next section.

Example 1. Consider the polynomial $p(x_1, x_2) = x_1^2 + x_2$ and its perturbation $p_\lambda(x_1, x_2) = p(x_1, x_2) + \lambda(x_1^4 + x_2^4)$. Then $p^* = -\infty$. One can compute explicitly the set V_λ of solutions to the system

$$\frac{\partial p_\lambda}{\partial x_1} = 2x_1(2\lambda x_1^2 + 1) = 0, \quad \frac{\partial p_\lambda}{\partial x_2} = 4\lambda x_2^3 + 1 = 0.$$

Namely, V_λ consists of the nine points (x_1, x_2) with $x_1 = 0, \pm i\sqrt{1/2\lambda}$, and $x_2 = -\sqrt[3]{1/4\lambda}, -j\sqrt[3]{1/4\lambda}, -j^2\sqrt[3]{1/4\lambda}$ (where $i, j \in \mathbb{C}$, $i^2 = -1$, $j^3 = 1$). Hence, $(0, -\sqrt[3]{1/4\lambda})$ is the only real point in V_λ and thus the unique minimizer of p over V_λ. This implies that $\mu_\lambda^* = -\sqrt[3]{1/4\lambda}$.

Example 2. Consider the polynomial $p(x_1, x_2) = (x_1^2 + x_2^2 - 1)^2$ whose minimum is $p^* = 0$ attained at all points on the unit circle. One can verify that the set V_λ contains 25 points, among them 9 real points, namely, $(0, 0)$ and

(i) $(x_1, x_2) = \pm(0, a)$, $\pm(a, 0)$, where $a := \sqrt{(-1 + \sqrt{6\lambda + 1})/3\lambda}$;

(ii) $(x_1, x_2) = (\pm b, \pm b)$, where $b := \sqrt{(-2 + \sqrt{6\lambda + 4})/3\lambda}$.

The minimum of p over V_λ is $\mu_\lambda^* = (2b^2 - 1)^2$, which is attained at the points $(\pm b, \pm b)$ in (ii). As $a = 1 + o(1)$ and $b = 1/\sqrt{2} + o(1)$, the limit as $\lambda \downarrow 0$ of the real points in V_λ are the points $(0, \pm 1)$, $(\pm 1, 0)$, $(\pm 1/\sqrt{2}, \pm 1/\sqrt{2})$ on the unit circle together with the origin.

3.2. Examples. We present here several examples on which our method has been tested. Let p be the polynomial whose infimum p^* is to be found and let $2m$ be its degree. We compute the approximations $\mu_{k,\lambda}^*$ of p^* provided by program (29). The computation is carried out for several values of λ, ranging typically from 10^{-1} to 10^{-4} (sometimes much smaller). We solve the program (29) for increasing values of k starting from $k = m$. Let $M_{\mathcal{B}_k}(y^*)$ be the returned optimum solution and $\mu_{k,\lambda}^*$ the returned optimum value. At $k = m$, we find the Lasserre lower bound $p_{L,m}^*$ for p^*.

At each step k, we check whether the rank condition (16) holds; if not, we go to the next step $k+1$. More precisely, we have the following:

- If $k = m$, then $\mu_{m,\lambda}^* = p_{L,m}^* \le p^*$. Moreover, $\mu_{m,\lambda}^* = p_{L,m}^* = p^*$ if $\operatorname{rank} M_{\mathcal{B}_m}(y^*) = \operatorname{rank}\ M_{\mathcal{B}_{m-1}}(y^*)$; i.e., the infimum p^* has been found.
- If $k \ge m+1$, and $\operatorname{rank}\ M_{\mathcal{B}_h}(y^*) = \operatorname{rank}\ M_{\mathcal{B}_{h-1}}(y^*) =: r$ for $h = k$ or $h = k-1$, then $\mu_{k,\lambda}^* \ge p^*$; moreover, one can extract r points $x \in \mathbb{R}^n$, and evaluating p at any such point x gives a certified upper bound on p^*.

There are two phases in the resolution of program (29): (1) Compute the entries of the matrix variable $M_{\mathcal{B}_k}(y)$ in (29); that is, compute the residue of $x^{\alpha+\beta}$ modulo I_λ with respect to the basis $\mathcal{B}$ for each $\alpha, \beta \in \mathcal{B}_k$; and (2) solve the semidefinite program (29). The first phase is carried out using *Mathematica* 4.2, and the semidefinite programming problem is solved with SeDuMi 1.05 (used with accuracy parameter pars.eps = 0). When evaluating the rank of a matrix we consider the eigenvalues with a precision of 10^{-3}; that is, we ignore all decimals starting with the fifth one.

In the tables below, at a given order k, (r_k, r_{k-1}, r_{k-2}) is the triple consisting of the ranks of the matrices $M_{\mathcal{B}_k}(y^*)$, $M_{\mathcal{B}_{k-1}}(y^*)$, $M_{\mathcal{B}_{k-2}}(y^*)$, where $M_{\mathcal{B}_k}(y^*)$ is the optimum solution to (29) returned by the algorithm.

In some examples, we also compute the upper approximations $\mu^*_{L,k,\lambda}$ on p^* obtained from program (12), and some other approximations obtained by minimizing p over a ball. Then $(r_k, r_{k-1}, \ldots)$ contains the ranks of the matrices $M_k(y^*)$, $M_{k-1}(y^*), \ldots$, where $M_k(y^*)$ is the optimum solution to (12) (or (4) when optimizing over a ball).

Example 1 (revisited). Consider again the polynomial $p(x_1, x_2) = x_1^2 + x_2$ with infimum $p^* = -\infty$. Then $n = 2$, $m = 1$, $|\mathcal{B}_1| = 3$, $|\mathcal{B}_2| = 6$, $|\mathcal{B}| = 9$. When computing the Lasserre lower bound $p^*_{L,1}$, GloptiPoly returns as expected that the "SeDuMi dual may be unbounded." As can be seen in Table 2, our algorithm retrieves a very accurate estimate of the minimizer $(0, -\sqrt[3]{1/4\lambda})$.

Table 2
Bounds $\mu^*_{k,\lambda}$ *for Example* 1.

λ	Order k	(r_k, r_{k-1}, r_{k-2})	$\mu^*_{k,\lambda}$	Extracted solutions
10^{-3}	2	(1,1,1)	-6.2996	$(0,-6.2996)$
10^{-6}	2	(1,1,1)	-62.9961	$(0,-62.9961)$
10^{-9}	2	(1,1,1)	-629.9606	$(0,-629.9606)$

Example 2 (revisited). Consider again the polynomial $p(x_1, x_2) = (x_1^2 + x_2^2 - 1)^2$ with infimum $p^* = 0$ attained at the points of the unit circle. Then $n = 2$, $m = 2$, $|\mathcal{B}_2| = 6$, $|\mathcal{B}_3| = 10$, $|\mathcal{B}_4| = 15$, $|\mathcal{B}| = 25$. The Lasserre lower bound is $p^*_{L,2} = 2.82\ 10^{-11} \leq p^*$ (with $r_2 = 5$, $r_1 = 3$).

Again, one can see in Table 3 that the algorithm retrieves very accurate estimates of the four minimizers $(\pm b, \pm b)$ of p over V_λ. Moreover, $\mu^*_{4,10^{-3}} \geq p^*$ and $\mu^*_{4,10^{-3}} \sim 10^{-7}$ is an accurate estimate of $p^* = 0$.

Table 3
Bounds $\mu^*_{k,\lambda}$ *for Example* 2.

λ	Order k	(r_k, r_{k-1}, r_{k-2})	$\mu^*_{k,\lambda}$	Extracted solutions
10^{-2}	3	(4,4,4)	$1.3854\ 10^{-5}$	$(\pm 0.7058, \pm 0.7058)$
10^{-3}	3	(9,5,3)	$1.3320\ 10^{-7}$	none
10^{-3}	4	(4,4,4)	$1.4043\ 10^{-7}$	$(\pm 0.7070, \pm 0.7070)$

Example 3. Consider the polynomial $p(x_1, x_2) = (x_1^2 + 1)^2 + (x_2^2 + 1)^2 - 2(x_1 + x_2 + 1)^2$. Then $n = 2$, $m = 2$, $|\mathcal{B}_3| = 10$, $|\mathcal{B}_4| = 15$, $|\mathcal{B}| = 25$. It is known (see [15]) that $p^* = -11.4581$ is attained at the point $(1.3247, 1.3247)$, and that the polynomial $p(x) - p^*$ is a sum of squares. Indeed, $p^*_{L,2} = -11.4581$ and, as $r_2 = r_1 = 1$, GloptiPoly extracts the minimizer $(1.3247, 1.3247)$. Nevertheless Table 4 shows the behavior of our method on this example.

We have also computed the bound $\beta^*_{L,k,\lambda}$ from (13), computing the order k moment relaxation for the minimum of p over the ball with radius R_λ as in (10). Here, $R_\lambda = \frac{56}{\lambda}$. For $\lambda = 10^{-1}$, $k = 2$, $R_\lambda = 560$ and GloptiPoly returns the value $\beta^*_{L,2,1/10} = -11.4581$ and extracts the solution $(1.3247, 1.3247)$.

Example 4. Consider the polynomial $p(x_1, x_2) = 1/27 + x_1^2 x_2^2 (x_1^2 + x_2^2 - 1)$, a dehomogenized version of the Motzkin polynomial, considered in [11]. Then $n = 2$,

Table 4
Bounds $\mu^*_{k,\lambda}$ *for Example* 3.

λ	Order k	(r_k, r_{k-1}, r_{k-2})	$\mu^*_{k,\lambda}$	Extracted solutions
10^{-2}	3	(1,1,1)	-11.4548	$(1.3109, 1.3109)$
10^{-3}	3	(1,1,1)	-11.4580	$(1.3233, 1.3233)$
10^{-4}	3	(1,1,1)	-11.4581	$(1.3246, 1.3246)$
10^{-5}	3	(1,1,1)	-11.4581	$(1.3247, 1.3247)$

Table 5a
Bounds $\mu^*_{k,\lambda}$ *for Example* 4.

λ	Order k	(r_k, r_{k-1}, r_{k-2})	$\mu^*_{k,\lambda}$	Extracted solutions
10^{-2}	4	(11,8,6)	$-8.8740\ 10^{-5}$	none
10^{-2}	5	(4,4,4)	$2.1500\ 10^{-6}$	$(\pm 0.5761, \pm 0.5761)$
10^{-3}	4	(11,8,6)	-0.0060	none
10^{-3}	5	(4,4,4)	$2.1897\ 10^{-8}$	$(\pm 0.5772, \pm 0.5772)$
10^{-4}	4	(11,8,6)	-0.0336	none
10^{-4}	5	(4,4,4)	$1.9042\ 10^{-10}$	$(\pm 0.5773, \pm 0.5773)$

Table 5b
Bounds $\mu^*_{L,k,\lambda}$ *for Example* 4.

λ	Order k	$(r_k, r_{k-1}, \dots, r_1)$	$\mu^*_{L,k,\lambda}$	Extracted solutions
10^{-2}	4	(15,10,6,3)	-1.0815	none
10^{-2}	5	(21,15,8,6,3)	-0.0060	none
10^{-2}	6	(11,4,4,4,4,3)	$2.1904\ 10^{-8}$	none*
10^{-2}	7	(-,4,4,4,4,4,3)	$2.1904\ 10^{-8}$	$(\pm 0.5772, \pm 0.5772)$
10^{-3}	4	(15,10,6,3)	-1.3072	none
10^{-3}	5	(21,15,8,6,3)	-0.0332	none
10^{-3}	6	(11,4,4,4,4,3)	$2.1993\ 10^{-10}$	none*
10^{-3}	7	(-,4,4,4,4,4,3)	$2.3084\ 10^{-10}$	$(\pm 0.5773, \pm 0.5773)$
10^{-4}	4	(15,10,6,3)	-1.1225	none
10^{-4}	5	(21,15,10,6,3)	-0.0909	none
10^{-4}	6	(11,4,4,4,4,3)	$1.0209\ 10^{-11}$	none*
10^{-4}	7	(-,4,4,4,4,4,3)	$1.498\ 10^{-11}$	$(\pm 0.5773, \pm 0.5773)$

$m = 3$, $|\mathcal{B}_3| = 10$, $|\mathcal{B}_4| = 15$, $|\mathcal{B}_5| = 21$, $|\mathcal{B}| = 49$. It is known that p has minimum $p^* = 0$, attained at $(\pm 1/\sqrt{3}, \pm 1/\sqrt{3})$, and p is not a sum of squares. As Table 5a shows, our algorithm finds a very accurate estimate of p^* and of its minimizers at the relaxation of order 5 when using the perturbation $\lambda = 10^{-4}$.

We have also computed the parameters $\mu^*_{L,k,\lambda}$ from (12) using GloptiPoly. The results are shown in Table 5b. We have $|S_4| = 15$, $|S_5| = 21$, $|S_6| = 28$, $|S_7| = 36$. (At the relaxation of order 7, GloptiPoly does not return the value of the rank of $M_7(y)$, which is indicated by "-" in the table.) At the relaxation of order 6, GloptiPoly does not yet extract a solution since the stronger rank condition (6) does not hold. However, this stronger condition is needed only to be able to claim that the extracted solution does satisfy the constraints $\partial p_\lambda / \partial x_i = 0$ $(i = 1, \dots, n)$. As $\operatorname{rank} M_{k-1}(y) = \operatorname{rank} M_{k-2}(y)$ one could already extract a solution at order 6, which permits us to claim that $\mu^*_{L,6,\lambda} \geq p^*$. Note, however, that our algorithm based on combinatorial moment matrices is able to find an upper bound for p^* at order $k = 5$ already.

Moreover, at a given order k, the parameter $\mu^*_{k,\lambda}$ is a more accurate approximation of p^* than the parameter $\mu^*_{L,k,\lambda}$.

Finally we have computed the bounds $\beta^*_{L,k,\lambda}$ from (13). Here, the radius is $R_\lambda = \frac{24}{\lambda}$. For $\lambda = 1/10$ and $k = 3, 4, 5$, SeDuMi reports that the "dual may be unbounded." For $\lambda = 1$, one finds $\beta^*_{L,3,\lambda} = -3.9722$ (with $(r_3, r_2, r_1) = (8, 6, 3)$, thus no solution extracted) and SeDuMi reports that the "dual may be unbounded" for $k \geq 4$.

When using the radius $R = 20$ (instead of R_λ), the smallest order k for which the rank condition holds for the moment relaxation is $k = 6$, where we find the upper bound $8.5345\ 10^{-12}$ for p^* and GloptiPoly extracts the solution $(\pm 0.5774, \pm 0.5774)$.

If we use a smaller radius $R = 2$, then the rank condition holds already at the moment relaxation of order $k = 3$, where we find the upper bound $1.2561\ 10^{-13}$ for p^* and GloptiPoly extracts the solutions $(\pm 0.5774, \pm 0.5774)$.

Therefore, the approach via optimization on a ball seems to work well only if one knows a priori a small ball containing a global minimizer.

Example 5. Consider the polynomial $p(x_1, x_2) = x_2^2 + (x_1x_2 - 1)^2$. This is a classical example of a polynomial having a finite infimum, which is not attained; $p^* = 0$ as $\lim_{\epsilon \downarrow 0} p(1/\epsilon, \epsilon) = 0$. Here, $n = 2$, $m = 2$, $|\mathcal{B}_2| = 6$, $|\mathcal{B}_3| = 10$, $|\mathcal{B}| = 25$. The Lasserre lower bound is $p^*_{L,2} = 5.4776\ 10^{-5}$ and Table 6a shows the bounds $\mu^*_{k,\lambda}$.

Table 6a

Bounds $\mu^*_{k,\lambda}$ for Example 5.

λ	Order k	(r_k, r_{k-1}, r_{k-2})	$\mu^*_{k,\lambda}$	Extracted solutions
10^{-2}	3	(2,2,2)	0.3385	$\pm(1.3981, 0.4729)$
10^{-3}	3	(2,2,2)	0.2082	$\pm(1.9499, 0.4060)$
10^{-4}	3	(2,2,2)	0.1232	$\pm(2.6674, 0.3287)$
10^{-5}	3	(2,2,2)	0.0713	$\pm(3.6085, 0.2574)$
10^{-6}	3	(2,2,2)	0.0408	$\pm(4.8511, 0.1977)$
10^{-7}	3	(3,2,2)	0.0231	$\pm(6.4986, 0.1503)$
10^{-8}	3	(3,2,2)	0.0131	$\pm(8.6882, 0.1136)$
10^{-9}	3	(8,4,2)	0.0074	none
10^{-10}	3	(7,4,2)	0.0041	none

We have also computed the bounds $\mu^*_{L,k,\lambda}$, shown in Table 6b. When the order k is marked with an asterisk (like 6^*), this means that we have rescaled the problem for SeDuMi (setting pars.scaling = [1 10]). (This is advised when the expected solutions have large entries; see the manual for GloptiPoly [10]. Without rescaling, the solution returned by GloptiPoly is approximatively 1, which is the value of p at the point $(0, 0)$ of V_λ, and thus not the true minimum.) Recall that $|S_3| = 10$, $|S_4| = 15$, $|S_5| = 21$, $|S_6| = 28$.

One can make the following observations regarding the results from Tables 6a and 6b. While our algorithm extracts the correct solutions at order $k = 3$, when using the moment relaxation to program (7) GloptiPoly needs to go to higher orders to be able to extract solutions. We have computed (with *Mathematica*) the points in the gradient variety V_λ; it turns out that there are three real points which are $(0, 0)$ and the two points extracted by the algorithms for the given values of λ in Tables 6a and 6b.

Example 6. Consider the polynomial $q(z_1, z_2, z_3, z_4, z_5) = \sum_{i=1}^{5} \prod_{j \neq i}(z_i - z_j)$, which is again an instance of a nonnegative polynomial which is not a sum of squares, due to Lax–Lax and Schmüdgen. More such examples can be found, e.g., in [27]. Introducing new variables $x_i := z_1 - z_{i+1}$ $(i = 1, \ldots, 4)$, minimizing $q(z)$ is equivalent

TABLE 6B
Bounds $\mu^*_{L,k,\lambda}$ for Example 5.

λ	Order k	$(r_k,\dots,r_1)$	$\mu^*_{L,k,\lambda}$	Extracted solutions
10^{-2}	3	(10,6,2)	0.0096	none
10^{-2}	4	(7,2,2,2)	0.3385	none
10^{-2}	5	(-,2,2,2,2)	0.3385	$\pm(1.3981, 0.4729)$
10^{-3}	3	(10,6,2)	0.0105	none
10^{-3}	4	(7,2,2,2)	0.2082	none
10^{-3}	5	(-,2,2,2,2)	0.2082	$\pm(1.9499, 0.4060)$
10^{-4}	3	(10,6,2)	0.0095	none
10^{-4}	4	(7,2,2,2)	0.1232	none
10^{-4}	5	(-,2,2,2,2)	0.1233	$\pm(2.6674, 0.3287)$
10^{-5}	5	(-,2,2,2,2)	0.0718	$\pm(3.6085, 0.2574)$
10^{-6}	6*	(-,-,2,2,2,2)	0.0408	$\pm(4.8511, 0.1977)$
10^{-7}	6*	(-,-,2,2,2,2)	0.0231	$\pm(6.4986, 0.1503)$
10^{-8}	6*	(-,-,2,2,2,2)	0.0131	$\pm(8.6882, 0.1136)$
10^{-9}	6*	(-,-,2,2,2,2)	0.0074	$\pm(11.6026, 0.0856)$
10^{-10}	6*	(-,-,2,2,2,2)	0.0042	$\pm(15.4849, 0.0643)$

to minimizing a polynomial p in the four variables $x_1,\dots,x_4$. After performing this substitution, we have $n = 4$, $m = 2$, $|\mathcal{B}_2| = 15$, $|\mathcal{B}_3| = 35$, $|\mathcal{B}_4| = 70$, $|\mathcal{B}| = 625$. When computing the lower bound $p^*_{L,2}$, SeDuMi reports that the "primal problem is infeasible" and the "dual problem may be unbounded." Table 7a gives some values of $\mu^*_{k,\lambda}$.

TABLE 7A
Bounds $\mu^*_{k,\lambda}$ for Example 6.

λ	Order k	(r_k, r_{k-1}, r_{k-2})	$\mu^*_{k,\lambda}$	Extracted solutions
10^{-1}	3	(20,10,5)	-0.0575	none
10^{-1}	4	(5,5,5)	$-8.9342\ 10^{-8}$	$\pm(0.0407, 0.0445, 0.0482, 0.0520)$ approx. (0,0,0,0) three times

Table 7b gives some values of the parameter $\mu^*_{L,k,\lambda}$. At order $k = 3$, for $\lambda = 10^{-1}, 10^{-2}$, SeDuMi reports that the "dual problem may be unbounded." On this example the parameter $\mu^*_{L,k,\lambda}$ appears to be a more accurate approximation of p^* than $\mu^*_{k,\lambda}$.

TABLE 7B
Bounds $\mu^*_{L,k,\lambda}$ for Example 6.

λ	Order k	(r_3, r_2, r_1)	$\mu^*_{L,k,\lambda}$	Extracted solutions
10^{-1}	4	(1,1,1)	$6.0249\ 10^{-15}$	$10^{-8}(-0.6138, -0.7014, 0.5825, 0.9606)$
10^{-2}	4	(1,1,1)	$3.9252\ 10^{-14}$	$10^{-8}(0.0602, 0.4502, -0.0416, -0.2084)$

As the polynomial p is homogeneous, i.e., $p(tx) = t^{2m}p(x)$ for all x ($m = 2$ here), there are in fact two possibilities for its infimum: Either $p^* = 0$ if p is nonnegative, or $p^* = -\infty$ otherwise. The parameters $\mu^*_{k,\lambda}$ and $\mu^*_{L,k,\lambda}$ are upper bounds for p^*. Hence, if for some small λ they are close to 0, it is then quite likely that $p^* = 0$ (since μ^*_λ converges to p^* as $\lambda \downarrow 0$), but this cannot be claimed with certitude. On the other hand, such upper bounds will be useful for proving that $p^* = -\infty$. Indeed, if we find a negative upper bound for p^*, then we can conclude that $p^* = -\infty$; moreover, any extracted solution gives a certificate for this. See Example 8 for an illustration.

When p is homogeneous, one can also test its nonnegativity by computing its minimum p^*_B over the unit ball B. Indeed, either $p^*_B = 0$ if p is nonnegative, or $p^*_B < 0$ otherwise. However, if p is nonnegative but not a sum of squares, then the moment relaxation (4) of any order k is never exact; i.e., the inequality $p^*_{L,k} \leq p^*_B$ is always strict (and thus the optimum matrix does not satisfy the rank condition). (Indeed, suppose that $p^*_{L,k} = p^*_B = 0$. For k large enough, there is no duality gap between (4) and (5), and (5) attains its supremum (see [15]). Hence, $\rho^*_k = p^*_{L,k} = p^*_B = 0$, implying that p can be written as $p = u + (1 - \sum_i x_i^2)v$, where u, v are sums of squares. As p is homogeneous, this implies easily that p must be a sum of squares (see [14]), yielding a contradiction.) Let us illustrate this in our current example. Table 7c shows the values $p^*_{L,k}$ obtained for the moment relaxations (4) for the minimum p^*_B of p over the unit ball. Recall that $|S_5| = 126$, $|S_6| = 210$.

Table 7c
Bounds from optimizing over a ball for Example 6.

Order k	$(r_k, r_{k-1}, \ldots, r_1)$	$p^*_{L,k}$	Extracted solutions
2	(10,5)	-0.0375	none
3	(25,15,5)	-0.0035	none
4	(39,29,15,5)	$-7.7935\ 10^{-4}$	none
5	(55,44,29,15,5)	$-2.7268\ 10^{-4}$	none
6	(210,126,70,29,15,5)	$-1.1936\ 10^{-4}$	none

Example 7. Consider the matrix

$$P = \begin{pmatrix} 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \end{pmatrix}$$

and the associated homogeneous polynomial $q(x) = \sum_{i,j=1}^5 x_i^2 x_j^2 P_{ij}$ (Example 5.4 in [22]). The matrix P is said to be *copositive* when q is nonnegative. Testing matrix copositivity is a co-NP-complete problem [21]. Although some necessary and sufficient conditions for the copositivity of a matrix are known (see, e.g., [13]), their algorithmic application is computationally too expensive. An alternative consists therefore of using numerical algorithms for testing (non)copositivity. Parrilo [22, 23] introduced the following criterion, useful for proving copositivity. Namely, if the polynomial $(\sum_{i=1}^n x_i^2)^r q(x)$ is a sum of squares for some integer $r \geq 0$, then q is nonnegative and thus P is copositive. For the matrix P considered in the present example, it is known that this criterion is satisfied for $r = 1$.

Let us nevertheless see the behavior of our method in this example. Due to symmetry, the polynomial q is nonnegative if and only if the (dehomogenized) polynomial $p(x) := q(x_1, x_2, x_3, x_4, 1)$ is nonnegative. Then $n = 4$, $m = 2$, $|\mathcal{B}_2| = 15$, $|\mathcal{B}_3| = 35$, $|\mathcal{B}_4| = 70$, $|\mathcal{B}_5| = 122$, $|\mathcal{B}| = 625$. The Lasserre lower bound is $p^*_{L,2} = -1.4955\ 10^6$ with $(r_1, r_2) = (5, 15)$ and Table 8 gives the parameters $\mu^*_{k,\lambda}$.

Example 8. Let $G = (V, E)$ be a graph with node set $V = \{1, \ldots, n\}$ and let A_G be its adjacency matrix, with $(A_G)_{ij} = 1$ if $ij \in E$ and $(A_G)_{ij} = 0$ otherwise for

Table 8
Bounds $\mu^*_{k,\lambda}$ for Example 7.

λ	Order k	(r_k, r_{k-1}, r_{k-2})	$\mu^*_{k,\lambda}$	Extracted solutions
10^{-2}	3	(18,6,3)	-1.5407	none
10^{-2}	4	(4,4,4)	$1.3854\ 10^{-5}$	$(\pm 0.7058, 0, 0, \pm 0.7058)$
10^{-3}	4	(4,4,4)	$1.3854\ 10^{-5}$	$(\pm 0.7058, 0, 0, \pm 0.7058)$
10^{-4}	4	(9,7,5)	$1.5544\ 10^{-7}$	none

$i, j \in V$. Consider the matrix

$$P := t(I + A_G) - J,$$

where $t \in \mathbb{R}$, I is the identity matrix and J is the all-ones matrix, and the associated homogeneous polynomial $p(x) := \sum_{i,j=1}^n x_i^2 x_j^2 P_{ij}$. By the Motzkin–Straus theorem [20], p is nonnegative (i.e., $p^* = 0$) (equivalently, P is a copositive matrix) if and only if $t \geq \alpha(G)$, where $\alpha(G)$ is the stability number of G, i.e., the largest cardinality of a stable set in G. In Example 7, G is the circuit $(1, 4, 2, 5, 3)$ on 5 nodes with $\alpha(G) = 2$ and $P = 2(I + A_G) - J$, which is therefore copositive. Consider now the case when G is the path $(1, 4, 2, 5, 3)$ on 5 nodes and $t = 2$, giving the matrix

$$P = \begin{pmatrix} 1 & -1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 & 1 \\ -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \end{pmatrix}.$$

Then P is not copositive, as $t < \alpha(G) = 3$ (note also $p(1, 1, 1, 0, 0) = -3$). This is confirmed by the results about p^* from Table 9a, where we have $n = 5$, $m = 2$, $|\mathcal{B}_1| = 6$, $|\mathcal{B}_2| = 21$, $|\mathcal{B}_3| = 56$, $|\mathcal{B}_4| = 126$, $|\mathcal{B}| = 3125$.

Table 9a
Bounds $\mu^*_{k,\lambda}$ for Example 8, when G is the path on 5 nodes and $t = 2$.

λ	Order k	(r_k, r_{k-1}, r_{k-2})	$\mu^*_{k,\lambda}$	Extracted solutions
1	3	(8,7,4)	-1.3333	none
1	4	(8,8,7)	-1.3333	two of the extracted solutions: $\pm(0.8165, 0.8165, 0.8165, 0, 0)$ $(0.8165 \sim \sqrt{2/3})$
10^{-1}	3	(8,7,4)	-133.3333	none
10^{-1}	4	(8,8,7)	-133.3333	two of the extracted solutions: $\pm(2.5820, 2.5820, 2.5820, 0, 0)$ $(2.5820 \sim \sqrt{20/3})$
10^{-2}	3	(8,7,4)	$-1.3333\ 10^4$	none

Consider now the case when G is the circuit $(1, 2, 3, 4, 5, 6)$ on 6 nodes and $t = 2$. Again the corresponding matrix P is not copositive, since $t < \alpha(G) = 3$. This is confirmed by the results about p^* from Table 9b. Because of symmetry, we made the computations for the polynomial $p(x_1, x_2, x_3, x_4, x_5, 1)$. Then $n = 5$, $m = 2$, $|B_2| = 21$, $|\mathcal{B}_3| = 56$, $|\mathcal{B}_4| = 126$.

Table 9b
Bounds $\mu^*_{k,\lambda}$ for Example 8, when G is the circuit on 6 nodes and $t = 2$.

λ	Order k	(r_k, r_{k-1}, r_{k-2})	$\mu^*_{k,\lambda}$	Extracted solutions
1	3	(4,4,3)	-2.2660	$(0, \pm 0.9036, 0, \pm 0.9036, 0)$
10^{-1}	3	(8,7,4)	-106.6640	none
10^{-1}	4	(8,8,7)	-106.6640	-
10^{-2}	3	(8,7,4)	$-1.3067\ 10^4$	none

In both instances we find a point x with $p(x) < 0$ (which certifies that P is not copositive) at the relaxation of order 3 or 4, already for the perturbation $\lambda = 1$. The bounds $\mu^*_{3,\lambda}$ decrease rapidly as λ goes to 0.

Consider finally the case when G is the circuit $(1, 2, 3, 4, 5, 6, 7)$ on 7 nodes and $t = 2$. Again, P is not copositive since $t < \alpha(G) = 3$. Due to symmetry it suffices to consider the polynomial p where we set $x_7 = 1$. Then $n = 6$, $m = 2$, $|\mathcal{B}_2| = 28$, $|\mathcal{B}_3| = 84$ and $|\mathcal{B}_4| = 210$. Table 9c shows some parameters $\mu^*_{3,\lambda}$ which again decrease rapidly as λ becomes small.

Table 9c
Bounds $\mu^*_{k,\lambda}$ for Example 8, when G is the circuit on 7 nodes and $t = 2$.

λ	Order k	(r_k, r_{k-1}, r_{k-2})	$\mu^*_{k,\lambda}$	Extracted solutions
1	3	(62,24,7)	-4.1114	none
10^{-1}	3	(62,24,7)	-304.7340	none
10^{-2}	3	(66,24,7)	$-3.0745\ 10^4$	none
10^{-3}	3	(83,27,7)	$-3.0808\ 10^6$	none

Example 9. Consider the polynomial $p(x) = \sum_{i=1,2,3} x_i^8 + p_0(x)$, where $p_0(x)$ is the Motzkin polynomial $x_1^2 x_2^2(x_1^2 + x_2^2 - 3x_3^2) + x_3^6$. It is known that $p^* = 0$ and that p is not a sum of squares (in fact, p is not a sum of squares modulo its gradient ideal [6]). In view of the form of p, we can apply directly our method for computing p^* without perturbing p. Table 10 shows values of the parameter μ^*_k from (25); as $\mu^*_k \le p^* \le 0$, we can conclude that $p^* \sim 0$ already at the relaxation of order $k = 4$. Here $n = 3$, $m = 3$, $|\mathcal{B}_3| = 20$, $|\mathcal{B}_4| = 35$, $|\mathcal{B}_5| = 56$, $|\mathcal{B}_6| = 84$.

Table 10
Bounds $\mu^*_{k,\lambda}$ for Example 9.

Order k	(r_k, r_{k-1}, r_{k-2})	μ^*_k	Extracted solutions
3	(12,2,1)	-1	none
4	(4,4,4)	$-1.1990\ 10^{-9}$	$\pm(0.0220, 0.0440, 0.0263)$ and approx. (0,0,0) twice
5	(4,4,4)	$-1.9880\ 10^{-10}$	$\pm(0.0160, 0.0319, 0.0274)$ and approx. (0,0,0) twice
6	(4,4,4)	$-8.8465\ 10^{-11}$	$\pm(0.0143, 0.0285, 0.0256)$ and approx. (0,0,0) twice

Example 10. Consider the polynomial $p(x) = (a^T x)^2 + \sum_{i=1}^n (x_i^2 - 1)^2$, where $a_1, \ldots, a_n$ are given positive integers. As mentioned in the introduction, the sequence $a = (a_1, \ldots, a_n)$ can be partitioned if and only if $p^* = 0$, in which case a global minimizer is ±1-valued and thus provides a partition of the sequence. Deciding whether an

Table 11a
The sequence $a = (2, 2, 2, 3, 3)$ is partitionable with $a^T x = 0$ at the returned solutions.

Order k	(r_k, r_{k-1}, r_{k-2})	μ_k^*	Extracted solutions
2	(10,5,1)	2.3994 10^{-9}	none
3	(2,2,2)	2.5072 10^{-10}	$\pm(1, 1, 1, -1, -1)$

Table 11b
The sequence $a = (1, 2, 3, 4, 5)$ is not partitionable as $p^* \geq \mu_3^* \geq \mu_2^* > 0$; its minimum gap is 1, realized at $\pm(1, 1, -1, 1, -1)$, obtained by rounding the extracted solutions.

Order k	(r_k, r_{k-1}, r_{k-2})	μ_k^*	Extracted solutions
2	(12,5,1)	0.0639	none
3	(2,2,2)	0.0657	$\pm(1.0157, 1.0308, -0.9477, 1.0590, -0.9069)$

Table 11c
The sequence $a = (2, 2, 3, 4, 5)$ is partitionable with $a^T x = 0$ at the returned solutions.

Order k	(r_k, r_{k-1}, r_{k-2})	μ_k^*	Extracted solutions
2	(2,2,1)	2.2649 10^{-12}	$\pm(1, 1, -1, 1, -1)$

Table 11d
The sequence $a = (3, 3, 4, 5, 6, 7)$ is partitionable with $a^T x = 0$ at the returned solutions.

Order k	(r_k, r_{k-1}, r_{k-2})	μ_k^*	Extracted solutions
2	(15,6,1)	-1.5649 10^{-8}	none
3	(4,4,3)	1.3816 10^{-8}	$\pm(1, -1, -1, 1, 1, -1)$, $\pm(1, -1, 1, -1, -1, 1)$

integer sequence can be partitioned is an NP-complete problem, and, more generally, computing the parameter $\gamma := \min_{z \in \{\pm 1\}^n} |a^T z|$ (the *minimum gap* of the sequence $a_1, \dots, a_n$) is NP-hard.

It is interesting to note[1] that $\gamma = 0$ (resp., $\gamma = 1$) if $p^* \leq \frac{1}{s^2}$ and $s := \sum_{i=1}^n a_i$ is even (resp., odd); moreover, a partition realizing the minimum gap can be obtained from a real point x with $p(x) \leq \frac{1}{s^2}$ by letting $z := sign(x)$ (with $z_i = 1$ if $x_i > 0$ and $z_i = -1$ otherwise). More generally, a similar argument permits us to show that a partition realizing the minimum gap γ can be derived from a global minimizer x to the polynomial $p_C(x) := (a^T x)^2 + C^2 \sum_{i=1}^n (x_i^2 - 1)^2$ by letting $z := sign(x)$, $C := \frac{1}{2}(\max_i a_i)(\sum_i a_i)$.

Again we can apply directly our method (without perturbation) for computing the minimum p^* of the polynomial p. If we find a positive lower bound μ_k^*, then we can conclude that the sequence cannot be partitioned. Although this approach can be used only for sequences of small length n (where the minimum gap could in fact easily be found directly), we consider below some sequences of length $n = 5, 6, 7, 10, 11$ to see the behavior of the method. We have $m = 1$, $(|\mathcal{B}_1|, |\mathcal{B}_2|, |\mathcal{B}_3|) = (6, 21, 51)$ (resp., $(7, 28, 78)$, $(8, 36, 113)$, $(11, 66, 276)$, $(12, 78, 353)$) if $n = 5$ (resp., $n = 6$, $n = 7$, $n = 10$, $n = 11$) and $|\mathcal{B}| = 3^n$. Results are shown in Tables 11a–11h.

[1] Indeed, let $x \in \mathbb{R}^n$ such that $p(x) \leq \frac{1}{s^2}$; thus $|a^T x|, |x_i^2 - 1| \leq \frac{1}{s}$. Define $z := sgn(x)$, i.e., $z_i := 1$ if $x_i > 0$ and $z_i = -1$ otherwise. Then $|a^T z| \leq |a^T(x - z)| + |a^T x| \leq 1 + \frac{1}{s} < 2$; indeed, $|a^T(x - z)| \leq \sum_i a_i |x_i - z_i| \leq \sum_i a_i |x_i - z_i||x_i + z_i| = \sum_i a_i |1 - x_i^2| \leq \frac{1}{s} \sum_i a_i = 1$. As $|a^T z|$ has the same parity as $s = \sum_i a_i$, $a^T z = 0$ if s is even, and $a^T z = \pm 1$ otherwise, which shows that the ± 1-vector z provides a partition of the sequence $a_1, \dots, a_n$ realizing the minimum gap.

Table 11e

The sequence $a = (1, 1, 2, 2, 3, 3, 13)$ is not partitionable as $p^* \geq \mu_2^* > 0$; its minimum gap is 1, realized at $\pm(1, 1, 1, 1, 1, 1, -1)$, obtained by rounding the extracted solutions.

Order k	(r_k, r_{k-1}, r_{k-2})	μ_k^*	Extracted solutions
2	(2,2,1)	0.0188	$\pm(1.0045, 1.0045, 1.0090, 1.0090, 1.0135, 1.0135, -0.9342)$

Table 11f

The sequence $a = (1, 1, 2, 2, 3, 3, 14)$ is not partitionable, as $p^* \geq \mu_2^* > 0$; its minimum gap is 2, realized at $\pm(1, 1, 1, 1, 1, 1, -1)$, obtained by rounding the extracted solutions.

Order k	(r_k, r_{k-1}, r_{k-2})	μ_k^*	Extracted solutions
2	(2,2,1)	0.0628	$\pm(1.0073, 1.0073, 1.0145, 1.0145, 1.0215, 1.0215, -0.8736)$

Table 11g

The sequence $a = (1, 2, 3, 20, 5, 6, 7, 10, 11, 77)$ is not partitionable as $p^* \geq \mu_2^* > 0$; its minimum gap is 12, realized at $\pm(1, 1, 1, 1, 1, 1, 1, 1, 1, -1)$, obtained by rounding the extracted solutions.

Order k	(r_k, r_{k-1}, r_{k-2})	μ_k^*	Extracted solutions
2	(2,2,1)	0.0758	$\pm(1.0015, 1.0029, 1.0044, 1.0282, 1.0073,$ $1.0087, 1.0101, 1.0144, 1.0158, -0.8580)$

Table 11h

The sequence $a = (1, 2, 3, 20, 5, 6, 7, 10, 11, 77, 3)$ is not partitionable as $p^* \geq \mu_2^* > 0$; its minimum gap is 9, realized at $\pm(1, 1, 1, 1, 1, 1, 1, 1, 1, -1, 1)$, obtained by rounding the extracted solutions.

Order k	(r_k, r_{k-1}, r_{k-2})	μ_k^*	Extracted solutions
2	(2,2,1)	0.0441	$\pm(1.0012, 1.0023, 1.0035, 1.0225, 1.0058,$ $1.0069, 1.0080, 1.0114, 1.0126, -0.8943, 1.0035)$

4. Conclusions. We consider the problem of computing the global infimum p^* of a multivariate polynomial p of degree $2m$. We propose a method for determining upper approximations μ_λ^* (or $\mu_{k,\lambda}^*$ for some integer $k \geq m$) for the infimum that converge to p^* as λ goes to 0. In the examples in which our method was tested, a tight upper bound $\mu_{k,\lambda}^*$ for p^* is very often found for k small ($k = m+1$ or $m+2$) by solving a semidefinite program of reasonable size, together with a real point x whose evaluation $p(x)$ gives a certificate for the upper bound. For small λ, $p(x)$ is in fact very close to the infimum p^* and x is close to a global minimizer (if some exists), which has been confirmed in the examples.

Our method applies to any polynomial; in particular, no assumption about the existence of a minimum is needed. In fact, it works with a perturbation p_λ of p, which has the property of having a minimum as well as a finite set V_λ of critical points. Moreover, the minima μ_λ^* of p over the set V_λ converge to p^* as λ goes to 0. One has two options for computing the minimum μ_λ^*: Either apply the moment relaxations of Lasserre [15] or apply the more compact relaxations via combinatorial moment matrices of Laurent [18] as proposed here. A feature of this second approach is that one has to solve smaller semidefinite programs, and, moreover, one can often extract a solution (giving a certified upper bound for p^*) at an earlier stage than in the approach based on the classical moment relaxation. In fact, our method can be applied directly to polynomials of the form $p = \sum_i c_i x_i^{2m} + p_0$, where $c_i \neq 0$ and $\deg(p_0) \leq 2m - 1$, without perturbing p; then it gives a monotonically nondecreasing hierarchy of lower bounds μ_k^* on the infimum. A limitation for our method is the size of the matrix variable $M_{\mathcal{B}_k}(y)$, which has to be generated and then processed by the semidefinite solver. Thus it applies only to medium-size problems.

Previous methods of Lasserre [15] and Parrilo [23] approximate the infimum of p by giving a hierarchy of lower bounds for p^*. Thus in a sense the various methods complement each other.

Parrilo's method computes for an integer $k \geq 0$ the parameter $\gamma_k^* := \sup\ \gamma$ such that $(\sum_i x_i^2)^k(p(x) - \gamma(\sum_i x_i^2)^m)$ is a sum of squares of polynomials. It is useful for proving that a homogeneous polynomial p is nonnegative, i.e., $p^* = 0$; indeed, if $\gamma_k^* \geq 0$ for some k, then p is nonnegative. On the other hand, our method is useful for proving that a homogeneous polynomial is not nonnegative (e.g., for proving that a matrix is not copositive). Indeed, if one finds an upper bound $\mu_{k,\lambda}^* < 0$ for p^*, then p is not nonnegative; in the examples such certified negative upper bounds on the infimum p^* are (often) found for a small order $k = m + 1$ or $m + 2$.

When applied to the unconstrained minimization of p, Lasserre's approach gives a lower bound $p_{L,m}^*$ for p^*, with equality $p^* = p_{L,m}^*$ if and only if $p - p^*$ is a sum of squares. One can construct a hierarchy of lower bounds converging to p^* by considering the constrained problem of minimizing p over its gradient variety (when p has a minimum) or over a ball (when a ball is known a priori containing a global minimum).

Let us finally mention another method based on perturbations recently introduced by Lasserre [16]. Given $\epsilon > 0$ and an integer $k \geq 0$, define the perturbed polynomial $p_{k,\epsilon} := p + \epsilon \sum_{r=0}^{k} \sum_{i=1}^{n} \frac{x_i^{2r}}{r!}$. Lasserre [16] defines the parameter

$$\ell_{k,\epsilon}^* := \inf\ p_{k,\epsilon}^T y \ \text{ s.t. } M_k(y) \succeq 0,\ y_0 = 1,$$

and shows that, given $\epsilon > 0$, $p^* \leq \ell_{k,\epsilon}^*$ for k large enough, and $\ell_{k,\epsilon}^* \leq p^* + \epsilon \sum_{i=1}^{n} e^{x_i^2}$ if x is a global minimum of p. From the numerical results given in [16], it appears that the bound $\ell_{k,\epsilon}^*$ is sensitive to the parameter ϵ (e.g., $\ell_{k,\epsilon}^*$ does not approximate p^* very well for some values of k and small ϵ) and $\ell_{k,\epsilon}^*$ provides less good approximations of p^* than when solving a constrained program with the first order conditions (which is, however, allowed only when p has a minimum).

REFERENCES

[1] S. Basu, R. Pollack, and M.-F. Roy, *Algorithms in Real Algebraic Geometry*, Springer-Verlag, Berlin, 2003.

[2] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms*, 2nd ed., Springer-Verlag, New York, 1997.

[3] D. Cox, J. Little, and D. O'Shea, *Using Algebraic Geometry*, Grad. Texts in Math. 185, Springer-Verlag, New York, 1998.

[4] R. E. Curto and L. A. Fialkow, *Solution of the truncated complex moment problem for flat data*, Mem. Amer. Math. Soc., 119 (1996).

[5] R. E. Curto and L. A. Fialkow, *The truncated complex K-moment problem*, Trans. Amer. Math. Soc., 352 (2000), pp. 2825–2855.

[6] J. W. Demmel, J. Nie, and B. Sturmfels, *Minimizing polynomials via sum of squares over the gradient ideal*, available online from http://arxiv.org/abs/math.OC/0411342, 2004.

[7] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman, San Francisco, CA, 1979.

[8] K. Hägglöf, P. O. Lindberg, and L. Stenvenson, *Computing global minima to polynomial optimization problems using Gröbner bases,* J. Global Optim., 7 (1995), pp. 115–125.

[9] B. Hanzon and D. Jibetean, *Global minimization of a multivariate polynomial using matrix methods*, J. Global Optim., 27 (2003), pp. 1–23.

[10] D. Henrion and J.-B. Lasserre, *GloptiPoly: Global optimization over polynomials with MATLAB and SeDuMi*, ACM Trans. Math. Software, 29 (2003), pp. 165–194.

[11] D. Henrion and J.-B. Lasserre, *Detecting global optimality and extracting solutions in GloptiPoly*, in Positive Polynomials in Control, Lecture Notes in Control and Inform. Sci. 312, D. Henrion and A. Garulli, eds., Springer-Verlag, Berlin, 2005, pp. 293–310.

[12] D. Jibetean, *Algebraic Optimization with Applications to System Theory*, Ph.D. thesis, Vrije Universiteit, Amsterdam, The Netherlands, 2003.

[13] W. Kaplan, *A test for copositive matrices*, Linear Algebra Appl., 313 (2000), pp. 203–206.

[14] E. De Klerk, M. Laurent, and P. Parrilo, *On the equivalence of algebraic approaches to the minimization of forms over the simplex*, in Positive Polynomials in Control, Lecture Notes in Control and Inform. Sci. 312, D. Henrion and A. Garulli, eds., Springer-Verlag, Berlin, 2005, pp. 121–133.

[15] J. B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM J. Optim., 11 (2001), pp. 796–817.

[16] J. B. Lasserre, *A New Hierarchy of SDP-Relaxations for Polynomial Programming*, Preprint, 2004.

[17] M. Laurent, *Revisiting two theorems of Curto and Fialkow on moment matrices*, Proc. Amer. Math. Soc., 133 (2005), pp. 2965–2976.

[18] M. Laurent, *Semidefinite representations for finite varieties*, Math. Program., to appear.

[19] M. Marshall, *Optimization of polynomial functions*, Canad. Math. Bull., 46 (2003), pp. 575–587.

[20] T. S. Motzkin and E. G. Straus, *Maxima for graphs and a new proof of a theorem of Túran*, Canad. J. Math., 17 (1965), pp. 533–540.

[21] K. G. Murty and S. N. Kabadi, *Some NP-complete problems in quadratic and nonlinear programming*, Math. Program., 39 (1987), pp. 117–129.

[22] P. A. Parrilo, *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*, Ph.D. thesis, California Institute of Technology, Pasadena, CA, 2000.

[23] P. A. Parrilo, *Semidefinite programming relaxations for semialgebraic problems*, Math. Program., 96 (2003), pp. 293–320.

[24] P. Parrilo, *An Explicit Construction of Distinguished Representations of Polynomials Nonnegative over Finite Sets*, Preprint, ETH, Zürich, 2002.

[25] P. A. Parrilo and B. Sturmfels, *Minimizing polynomial functions*, in Algorithmic and Quantitative Real Algebraic Geometry, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 60, S. Basu and L. Gonzales-Vega, eds., AMS, Providence, RI, 2003, pp. 83–99.

[26] M. Putinar, *Positive polynomials on compact semialgebraic sets*, Indiana Univ. Math. J., 42 (1993), pp. 969–984.

[27] B. Reznick, *Uniform denominators in Hilbert's* 17*th problem*, Math. Z., 220 (1995), pp. 75–97.

[28] N. Shor, *Class of global minimum bounds of polynomial functions*, Translated from Kibernetica, 6 (1987), pp. 9–11.

SIAM J. Optim.
Vol. 16, No. 2, pp. 515–530

SECOND-ORDER BEHAVIOR OF PATTERN SEARCH[*]

MARK A. ABRAMSON[†]

Abstract. Previous analyses of pattern search algorithms for unconstrained and linearly constrained minimization have focused on proving convergence of a subsequence of iterates to a limit point satisfying either directional or first-order necessary conditions for optimality, depending on the smoothness of the objective function in a neighborhood of the limit point. Even though pattern search methods require no derivative information, we are able to prove some limited directional second-order results. Although not as strong as classical second-order necessary conditions, these results are stronger than the first-order conditions that many gradient-based methods satisfy. Under fairly mild conditions, we can eliminate from consideration all strict local maximizers and an entire class of saddle points.

Key words. nonlinear programming, pattern search algorithm, derivative-free optimization, convergence analysis, second-order optimality conditions

AMS subject classifications. 90C30, 90C56, 65K05

DOI. 10.1137/04060367X

1. Introduction.

In this paper, we consider the class of generalized pattern search (GPS) algorithms applied to the linearly constrained optimization problem,

$$\min_{x \in X} f(x), \tag{1.1}$$

where the function $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, and $X \subseteq \mathbb{R}^n$ is defined by a finite set of linear inequalities, i.e., $X = \{x \in \mathbb{R}^n : Ax \geq b\}$, where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. We treat the unconstrained, bound constrained, and linearly constrained problems together because in these cases, we apply the algorithm, not to f, but to the "barrier" objective function $f_X = f + \psi_X$, where ψ_X is the indicator function for X; i.e., it is zero on X and infinity elsewhere. If a point x is not in X, then we set $f_X(x) = \infty$, and f is not evaluated. This is important in many practical engineering problems in which f is expensive to evaluate.

The class of derivative-free pattern search algorithms was originally defined and analyzed by Torczon [27] for unconstrained optimization problems with a continuously differentiable objective function f. Torczon's key result is the proof that there exists a subsequence of iterates that converges to a point x^* which satisfies the first-order necessary condition, $\nabla f(x^*) = 0$. Lewis and Torczon [20] add the valuable connection between pattern search methods and positive basis theory [16] (the details of which are ingrained into the description of the algorithm in section 2). They extend the class to solve problems with bound constraints [21] and problems with a finite number of linear

[*] Received by the editors January 31, 2004; accepted for publication (in revised form) April 26, 2005; published electronically December 8, 2005. The views expressed in this paper are those of the author and do not reflect the official policy or position of the U.S. Air Force, Department of Defense, or U.S. Government. This work was performed by an employee of the U.S. Government or under U.S. Government contract. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.

http://www.siam.org/journals/siopt/16-2/60367.html

[†] Department of Mathematics and Statistics, Air Force Institute of Technology, AFIT/ENC, 2950 Hobson Way, Building 641, Wright Patterson AFB, OH 45433-7765 (Mark.Abramson@afit.edu, http://en.afit.edu/enc/Faculty/MAbramson/abramson.html).

constraints [22], showing that if f is continuously differentiable, then a subsequence of iterates converges to a point satisfying the Karush–Kuhn–Tucker (KKT) first-order necessary conditions for optimality.

Audet and Dennis [7] add a hierarchy of convergence results for unconstrained and linearly constrained problems whose strength depends on the local smoothness of the objective function. They apply principles of the Clarke [12] nonsmooth calculus to show convergence to a point having nonnegative generalized directional derivatives in a set of directions that positively span the tangent cone there. They show convergence to a first-order stationary (or KKT) point under the weaker hypothesis of strict differentiability at the limit point and illustrate how the results of [21, 22, 27] are corollaries of their own work.

Audet and Dennis also extend GPS to categorical variables [6], which are discrete variables that cannot be treated by branch and bound techniques. This approach is successfully applied to engineering design problems in [2] and [19]. The theoretical results here can certainly be applied to these *mixed variable problems*, with the caveat that results would be with respect to the continuous variables (i.e., while holding the categorical variables fixed). An adaptation of the results in [6] to more general objective functions using the Clarke [12] calculus can be found in [1].

The purpose of this paper is to provide insight into the second-order behavior of the class of GPS algorithms for unconstrained and linearly constrained optimization. This may seem somewhat counterintuitive, in that, except for the approach described in [3], GPS methods do not even use first derivative information. However, the nature of GPS in evaluating the objective in multiple directions does, in fact, lend itself to some limited discovery of second-order theorems, which are generally stronger than what can be proved for many gradient-based methods. Specifically, while we cannot ensure positive semidefiniteness of the Hessian matrix in all directions (and, in fact, we show a few counterexamples), we can establish this result with respect to a certain subset of the directions, so that the likelihood of convergence to a point that is not a local minimizer is reasonably small.

This paper does not address the question of second-order behavior of GPS algorithms for general nonlinear constraints. Extending convergence results of basic GPS to problems with nonlinear constraints requires augmentation to handle these constraints. Lewis and Torczon [23] do this by approximately solving a series of bound constrained augmented Lagrangian subproblems [14], while Audet and Dennis [9] use a filter-based approach [17]. The results presented here may be extendable to the former but not the latter, since the filter approach given in [9] cannot be guaranteed to converge to a first-order KKT point. The direct search algorithm of Lucidi, Sciandrone, and Tseng [24] applies positive basis theory to handle nonlinear constraints in a way similar to GPS, but it requires constraint derivatives and satisfaction of a sufficient decrease condition to ensure convergence, which [23] and [9] do not. Because of dissatisfaction with these limitations, Audet and Dennis [8] recently introduced the class of mesh-adaptive direct search (MADS) algorithms, a generalization of GPS that achieves first-order convergence for nonlinear constrained problems by generating a set of feasible directions that, in the limit, becomes asymptotically dense in the tangent cone. We plan to study second-order convergence properties of MADS in future work.

The remainder of this paper is organized as follows. In the next section, we briefly describe the basic GPS algorithm, followed by a review of known convergence results for basic GPS algorithms in section 3. In section 4, we show that, while convergence to a local maximizer is possible, it can only happen under some very strong assumptions on both the objective function and the set of directions used by the algorithm. In section 5, we introduce additional theorems to describe second-order behavior of GPS more generally, along with a few examples to illustrate the theory and show that certain hypotheses cannot be relaxed. Section 6 offers some concluding remarks.

Notation. $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{Z}$, and $\mathbb{N}$ denote the set of real numbers, rational numbers, integers, and nonnegative integers, respectively. For any set S, $|S|$ denotes the cardinality of S, and $-S$ is the set defined by $-S = \{-s : s \in S\}$. For any finite set A, we may also refer to the matrix A as the one whose columns are the elements of A. Similarly, for any matrix A, the notation $a \in A$ means that a is a column of A. For $x \in X$, the *tangent cone* to X at x is $T_X(x) = \mathrm{cl}\{\mu(w - x) : \mu \geq 0,\ w \in X\}$, and the *normal cone* $N_X(x)$ to X at x is the polar of the tangent cone; namely, $N_X(x) = \{v \in \mathbb{R}^n : v^T w \leq 0 \quad \forall w \in T_X(x)\}$. It is the nonnegative span of all outwardly pointing constraint normals at x.

2. Generalized pattern search algorithms.

For unconstrained and linearly constrained optimization problems, the basic GPS algorithm generates a sequence of iterates having nonincreasing function values. Each iteration consists of two main steps, an optional SEARCH phase and a local POLL phase, in which the barrier objective function f_X is evaluated at a finite number of points that lie on a mesh, with the goal of finding a point with lower objective function value, which is called an *improved mesh point*.

The mesh is not explicitly constructed; rather, it is conceptual. It is defined primarily through a set of positive spanning directions D in $\mathbb{R}^n$, i.e., where every vector in $\mathbb{R}^n$ may be represented as a nonnegative linear combination of the elements of D. For convenience, we also view D as a real $n \times n_D$ matrix whose n_D columns are its elements. The only other restriction on D is that it must be formed as the product

$$D = GZ, \tag{2.1}$$

where $G \in \mathbb{R}^{n \times n}$ is a nonsingular real generating matrix, and $Z \in \mathbb{Z}^{n \times n_D}$ is an integer matrix of full rank. In this way, each direction $d_j \in D$ may be represented as $d_j = Gz_j$, where $z_j \in \mathbb{Z}^n$ is an integer vector. At iteration k, the mesh is defined by the set

$$M_k = \bigcup_{x \in S_k} \{x + \Delta_k D z : z \in \mathbb{N}^{n_D}\}, \tag{2.2}$$

where $S_k \in \mathbb{R}^n$ is the set of points where the objective function f had been evaluated by the start of iteration k, and $\Delta_k > 0$ is the mesh size parameter that controls the fineness of the mesh. This construction is the same as that of [8] and [9], which generalizes the one given in [7]. It ensures that all previously computed iterates will lie on the current mesh.

The SEARCH step is simply an evaluation of a finite number of mesh points. It retains complete flexibility in choosing the mesh points, with the only caveat being that the points must be finite in number (including none). This could include a few iterations using a heuristic, such as a genetic algorithm, random sampling, etc., or, as is popular among many in industry (see [5, 10, 11, 25]), the approximate optimization on the mesh of a less expensive surrogate function. A related algorithm that does not require the surrogate solution to lie on the mesh (but requires additional assumptions for convergence) is found in [15].

If the SEARCH step fails to generate an improved mesh point, the POLL step is performed. This step is much more rigid in its construction, but this is necessary in order to prove convergence. The POLL step consists of evaluating f_X at points neighboring the current iterate x_k on the mesh. This set of points P_k is called the *poll set* and is defined by

$$P_k = \{x_k + \Delta_k d : d \in D_k \subseteq D\} \subset M_k, \tag{2.3}$$

where D_k is a positive spanning set of directions taken from D. We write $D_k \subset D$ to mean that the columns of D_k are taken from the columns of D. Choosing a subset $D_k \subset D$ of positive spanning directions at each iteration also adds the flexibility that will allow us to handle linear constraints in an efficient fashion.

If either the SEARCH or POLL step is successful in finding an improved mesh point, then the iteration ends immediately, with that point becoming the new iterate x_{k+1}. In this case, the mesh size parameter is either retained or increased (i.e., the mesh is coarsened). If neither step finds an improved mesh point, then the point x_k is said to be a *mesh local optimizer* and is retained as the new iterate $x_{k+1} = x_k$, and the mesh size parameter is reduced (i.e., the mesh is refined).

The rules that govern mesh coarsening and refining are as follows. For a fixed rational number $\tau > 1$ and two fixed integers $w^- \leq -1$ and $w^+ \geq 0$, the mesh size is updated according to the rule

$$\Delta_{k+1} = \tau^{w_k} \Delta_k, \tag{2.4}$$

where $w_k \in \{0, 1, \ldots, w^+\}$ if the mesh is coarsened, or $w_k \in \{w^-, w^- + 1, \ldots, -1\}$ if the mesh is refined.

From (2.4), it follows that, for any $k \geq 0$, there exists an integer r_k such that

$$\Delta_{k+1} = \tau^{r_k} \Delta_0. \tag{2.5}$$

The basic GPS algorithm is given in Figure 2.1.

GENERALIZED PATTERN SEARCH (GPS) ALGORITHM.

Initialization: Let S_0 be a set of initial points, and let $x_0 \in S_0$ satisfy $f_X(x_0) < \infty$ and $f_X(x_0) \leq f_X(y)$ for all $y \in S_0$. Let $\Delta_0 > 0$, and let D be a finite set of n_D positive spanning directions. Define $M_0 \subset X$ according to (2.2).

For $k = 0, 1, 2, \ldots$, perform the following:

1. **SEARCH step:** Optionally employ some finite strategy seeking an improved mesh point; i.e., $x_{k+1} \in M_k$ satisfying $f_X(x_{k+1}) < f_X(x_k)$.
2. **POLL step:** If the SEARCH step was unsuccessful or not performed, evaluate f_X at points in the poll set P_k (see (2.3)) until an improved mesh point x_{k+1} is found, or until all points in P_k have been evaluated.
3. **Update:** If SEARCH or POLL finds an improved mesh point,
   Update x_{k+1}, and set $\Delta_{k+1} \geq \Delta_k$ according to (2.4);
   Otherwise, set $x_{k+1} = x_k$, and set $\Delta_{k+1} < \Delta_k$ according to (2.4).

FIG. 2.1. *Basic GPS algorithm.*

With the addition of linear constraints, in order to retain first-order convergence properties, the set of directions D_k must be chosen to conform to the geometry of the constraints. The following definition, found in [7] (as an abstraction of the ideas and approach of [22]), gives a precise description for what is needed for convergence.

Definition 2.1. *A rule for selecting the positive spanning sets* $D_k = D(k, x_k) \subseteq D$ conforms to X *for some* $\epsilon > 0$ *if at each iteration* k *and for every boundary point* $y \in X$ *satisfying* $\|y - x_k\| < \epsilon$ *the tangent cone* $T_X(y)$ *is generated by nonnegative linear combinations of the columns of* D_k.

Using standard linear algebra tools, Lewis and Torczon [22] provide a clever algorithm to actually construct the sets D_k. If these sets are chosen so that they conform to X, all iterates lie in a compact set, and f is sufficiently smooth, then a subsequence of GPS iterates converges to a first-order stationary point [7, 22].

3. Existing convergence results. Before presenting new results, it is important to state what is currently known about the convergence properties of GPS for linearly constrained problems.

We first make the following assumptions:

A1: All iterates $\{x_k\}$ produced by the GPS algorithm lie in a compact set.

A2: The set of directions $D = GZ$, as defined in (2.1), includes tangent cone generators for every point in X.

A3: The rule for selecting positive spanning sets D_k conforms to X for some $\epsilon > 0$.

Assumption A1, which is already sufficient to guarantee the existence of convergent subsequences of the iteration sequence, is a standard assumption [6, 7, 9, 14, 15, 17, 21, 22, 27]. A sufficient condition for this to hold is that the level set $L(x_0) = \{x \in X : f(x) \leq f(x_0)\}$ is compact. We can assume that $L(x_0)$ is bounded, but not closed, since we allow f to be discontinuous and extended valued. Thus we can assume that the closure of $L(x_0)$ is compact. We should also note that most real engineering optimization problems have simple bounds on the design variables, which is enough to ensure that Assumption A1 is satisfied, since iterates lying outside of X are not evaluated by GPS. In the unconstrained case, note that Assumptions A2 and A3 are automatically satisfied by any positive spanning set constructed from the product in (2.1).

Assumption A2 is automatically satisfied if $G = I$ and the constraint matrix A is rational, as is the case in [22]. Note that the finite number of linear constraints ensures that the set of tangent cone generators for all points in X is finite, which ensures that the finiteness of D is not violated.

If f is lower semicontinuous at any GPS limit point $\bar{x}$, then $f(\bar{x}) \leq \lim_k f(x_k)$, with equality if f is continuous [7]. Of particular interest are limit points of certain subsequences (indexed by some index set K) for which $\lim_{k \in K} \Delta_k = 0$. We know that at least one such subsequence exists because of Torczon's [27] key result, restated here for convenience.

Theorem 3.1. *The mesh size parameters satisfy* $\liminf_{k \to +\infty} \Delta_k = 0$.

From this result, we are interested in subsequences of iterates that converge to a limit point associated with Δ_k converging to zero. The following definitions are due to Audet and Dennis [7].

Definition 3.2. *A subsequence of GPS mesh local optimizers* $\{x_k\}_{k \in K}$ *(for some subset of indices* K*) is said to be a* refining subsequence *if* $\{\Delta_k\}_{k \in K}$ *converges to zero.*

Definition 3.3. *Let* $\hat{x}$ *be a limit point of a refining subsequence* $\{x_k\}_{k \in K}$. *A direction* $d \in D$ *is said to be a* refining direction *of* $\hat{x}$ *if* $x_k + \Delta_k d \in X$ *and* $f(x_k) \leq f(x_k + \Delta_k d)$ *for infinitely many* $k \in K$.

Audet and Dennis [6] prove the existence of at least one convergent refining subsequence. An important point is that, since a refining direction d is one in which $x_k + \Delta_k d \in X$ infinitely often in the subsequence, it must be a feasible direction at the $\hat{x}$, and thus lies in the tangent cone $T_X(\hat{x})$.

The key results of Audet and Dennis are now given. The first shows directional optimality conditions under the assumption of Lipschitz continuity and is obtained by a very short and elegant proof (see [7]) using Clarke's [12] definition of the generalized directional derivative. Audet [4] provides an example to show that Lipschitz continuity (and even differentiability) is not sufficient to ensure convergence to a Clarke stationary point (i.e., where zero belongs to the Clarke generalized gradient). The second result, along with its corollary for unconstrained problems, shows convergence to a point satisfying first-order necessary conditions for optimality. The latter two results were originally proved by Torczon [27] and Lewis and Torczon [21, 22] under the assumption of continuous differentiability of f on the level set containing all of the iterates. Audet and Dennis [7] prove the same results, stated here, requiring only strict differentiability at the limit point.

THEOREM 3.4. *Let $\hat{x}$ be a limit of a refining subsequence, and let $d \in D$ be any refining direction of $\hat{x}$. Under Assumptions* A1–A3*, if f is Lipschitz continuous near $\hat{x}$, then the generalized directional derivative of f at $\hat{x}$ in the direction d is nonnegative, i.e., $f^\circ(\hat{x}; d) \geq 0$.*

THEOREM 3.5. *Under Assumptions* A1–A3*, if f is strictly differentiable at a limit point $\hat{x}$ of a refining subsequence, then $\nabla f(\hat{x})^T w \geq 0$ for all $w \in T_X(\hat{x})$, and $-\nabla f(\hat{x}) \in N_X(\hat{x})$. Thus, $\hat{x}$ satisfies the KKT first-order necessary conditions for optimality.*

COROLLARY 3.6. *Under Assumption* A1*, if f is strictly differentiable at a limit point $\hat{x}$ of a refining subsequence, and if $X = \mathbb{R}^n$ or $\hat{x} \in \operatorname{int}(X)$, then $\nabla f(\hat{x}) = 0$.*

Although GPS is a derivative-free method, its strong dependence on the set of mesh directions presents some advantages in terms of convergence results. For example, if f is only Lipschitz continuous at certain limit points x^*, Theorem 3.4 provides a measure of directional optimality there in terms of the Clarke generalized directional derivatives being nonnegative [7]. In the next two sections, we attempt to prove certain second-order optimality conditions, given sufficient smoothness of the objective function f. Our goal is to quantify our belief that convergence of GPS to a point that is not a local minimizer is very rare.

4. GPS and local maximizers.

We treat the possibility of convergence to a local maximizer separate from other stationary points because what we can prove requires far less stringent assumptions. We begin with an example, provided by Charles Audet, to show that it is indeed possible to converge to a maximizer, even when f is smooth.

EXAMPLE 4.1. *Let $f : \mathbb{R}^2 \to \mathbb{R}$ be the continuously differentiable function defined by*

$$f(x, y) = -x^2 y^2.$$

Choose $(x_0, y_0) = (0, 0)$ as the initial point, and set $D = [e_1, e_2, -e_1, -e_2]$, where e_1 and e_2 are the standard coordinate directions. Now observe that if the SEARCH *phase is empty, then the iteration sequence begins at the global maximizer $(0, 0)$, but can never move off of that point because the directions in D are lines of constant function value. Thus the sequence x_k converges to the global maximizer $(0, 0)$.*

Example 4.1 is clearly pathological. Had we started at *any* other point or polled in any other direction (that is not a scalar multiple of a coordinate direction), the algorithm would not have stalled at the maximizer $(0, 0)$. However, it is clear that the method of steepest descent and even Newton's method would also fail to move away from this point.

From this example, one can envision other cases (also pathological) in which convergence to a local maximizer is possible, but without starting there. However, we can actually characterize these rare situations in which convergence to a maximizer can occur. Lemma 4.2 shows that convergence would be achieved after only a finite number of iterations. Under a slightly stronger assumption, Theorem 4.3 ensures that convergence to a maximizer means that every refining direction is a direction of constant function value. This very restrictive condition is consistent with Example 4.1. Corollary 4.4 establishes the key result that, under appropriate conditions, convergence to a strict local maximizer cannot occur. This result does not hold for gradient-based methods, even when applied to smooth functions.

LEMMA 4.2. *Let $\hat{x}$ be the limit of a refining subsequence. If f is lower semicontinuous at $\hat{x}$, and if $\hat{x}$ is a local maximizer of f in X, then $x_k = \hat{x}$ is achieved in a finite number of iterations.*

Proof. Since $\hat{x} = \lim_{k \in K} x_k$ is a local maximizer for f in X, there exists an open ball $B(\hat{x}, \epsilon)$ of radius ϵ, centered at $\hat{x}$, for some $\epsilon > 0$, such that $f(\hat{x}) \geq f(y)$ for all $y \in B(\hat{x}, \epsilon) \cap X$. Then for all sufficiently large $k \in K$, $x_k \in B(\hat{x}, \epsilon) \cap X$, and thus $f(\hat{x}) \geq f(x_k)$. But since GPS generates a nonincreasing sequence of function values, and since f is lower semicontinuous at $\hat{x}$, it follows that

$$f(x_k) \leq f(\hat{x}) \leq f(x_{k+1}) \leq f(x_k), \tag{4.1}$$

and thus $f(x_k) = f(\hat{x})$, for all sufficiently large $k \in K$. But since GPS iterates satisfy $x_{k+1} \neq x_k$ only when $f(x_{k+1}) < f(x_k)$, it follows that $x_k = \hat{x}$ for all sufficiently large k. □

THEOREM 4.3. *Let $\hat{x}$ be the limit of a refining subsequence. If f is lower semicontinuous in a neighborhood of $\hat{x}$, and if $\hat{x}$ is a local maximizer of f in X, then every refining direction is a direction of constant function value.*

Proof. Let $d \in D(\hat{x})$ be a refining direction. Since $\hat{x}$ is a local maximizer, there exists $\hat{\delta} > 0$ such that $\hat{x} + td \in X$ and $f(\hat{x}) \geq f(\hat{x} + td)$ for all $t \in (0, \hat{\delta})$. Now suppose that there exists $\delta \in (0, \hat{\delta})$ such that f is continuous in $B(\hat{x}, \delta)$ and $f(\hat{x}) > f(\hat{x} + td)$ for all $t \in (0, \delta)$. Then $f(\hat{x}) > f(\hat{x} + \Delta_k d)$ for $\Delta_k \in (0, \delta)$. But since Lemma 4.2 ensures convergence of GPS in a finite number of steps, we have the contradiction, $f(\hat{x}) = f(x_k) \leq f(x_k + \Delta_k d) = f(\hat{x} + \Delta_k d)$ for all sufficiently large k. Therefore there must exist $\delta > 0$ such that $f(\hat{x}) = f(\hat{x} + td)$ for all $t \in (0, \delta)$. □

COROLLARY 4.4. *The GPS algorithm cannot converge to any strict local maximizer of f at which f is lower semicontinuous.*

Proof. If $\hat{x}$ is a strict local maximizer of f in X, then the first inequality of (4.1) is strict, yielding the contradiction, $f(x_k) < f(\hat{x}) \leq f(x_k)$. □

The assumption that f is lower semicontinuous at $\hat{x}$ is necessary for all three of these results to hold. As an example, consider the function $f(x) = 1$ if $x = 0$, and x^2 otherwise. This function has a strict local maximum at 0, and there are clearly no directions of constant function value. It is easy to see that any sequence of GPS iterates will converge to zero, and by choosing an appropriate starting point and mesh size, we can prevent convergence in a finite number of iterations. The theory is not violated because f is not lower semicontinuous there.

The additional assumption in Theorem 4.3 of lower semicontinuity in a neighborhood of the limit point (not just at the limit point) is needed to avoid other pathological examples, such as the function $f(x) = 0$ if $x \in \mathbb{Q}$ and $-x^2$ if $x \notin \mathbb{Q}$. Continuity of f only holds at the local maximizer 0, and there are no directions of constant function value. A typical instance of GPS that uses rational arithmetic would stall at the starting point of 0.

5. Second-order theorems.

An interesting observation about Example 4.1 is that even though $(0, 0)$ is a local (and global) maximizer, the Hessian matrix is equal to the zero matrix there, meaning that it is actually positive semidefinite. This may seem counterintuitive, but it is simply a case where the curvature of the function is described by Taylor series terms of higher than second order.

Thus an important question not yet answered is whether GPS can converge to a stationary point at which the Hessian is not positive semidefinite (given that the objective is twice continuously differentiable near the stationary point). The following simple example demonstrates that it is indeed possible, but once again the algorithm does not move off of the starting point.

EXAMPLE 5.1. *Let $f : \mathbb{R}^2 \to \mathbb{R}$ be the continuously differentiable function defined by*

$$f(x, y) = xy.$$

Choose $(x_0, y_0) = (0, 0)$ as the initial point, and set $D = [e_1, e_2, -e_1, -e_2]$, where e_1 and e_2 are the standard coordinate directions. Now observe that if the SEARCH *step is empty, then the iteration sequence begins at the saddle point $(0, 0)$, but can never move off of that point because the directions in D are lines of constant function value. Thus the sequence x_k converges to the saddle point. Furthermore, the Hessian of f at $(0, 0)$ is given by*

$$\nabla^2 f(0, 0) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

which is indefinite, having eigenvalues of ± 1.

This result is actually not surprising, since many gradient-based methods have this same limitation. However, the results that follow provide conditions by which a pseudo-second-order necessary condition is satisfied—one that is weaker than the traditional second-order necessary condition, but stronger than the first-order condition that is all that can be guaranteed by most gradient-based methods.

We are now ready to present one of the main results of this paper. This will require the use of the Clarke [12] calculus in a manner similar to that of [7], but applied to f' instead of f itself. We will denote by $f^{\circ\circ}(x; d_1, d_2)$ the Clarke generalized directional derivative in the direction d_2 of the directional derivative $f'(x; d_1)$ of f at x in the fixed direction d_1. In other words, if $g(x) = f'(x; d_1)$, then $f^{\circ\circ}(x; d_1, d_2) = g^{\circ}(x; d_2)$. We should note that this is consistent with the concepts and notation given in [13] and [18]; however, we have endeavored to simplify the discussion for clarity. First, we give a general lemma that is independent of the GPS algorithm. The theorem and corollary that follow will be key to establishing a pseudo-second-order result for GPS.

LEMMA 5.2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable at x, and let $f'(\cdot; \pm d)$ be Lipschitz near x. Then*

$$f^{\circ\circ}(x; d, d) = \limsup_{y \to x, t \downarrow 0} \frac{f(y + td) - 2f(y) + f(y - td)}{t^2}.$$

Proof. In general, we can apply the definition of the generalized directional derivative and the backward difference formula for directional derivatives to obtain

$$\begin{aligned} f^{\circ\circ}(x;d,d) &= g^{\circ}(x;d) = \limsup_{y\to x, t\downarrow 0} \frac{g(y+td)-g(y)}{t} \\ &= \limsup_{y\to x, t\downarrow 0} \frac{f'(y+td;d)-f'(y;d)}{t} \\ &= \limsup_{y\to x, t\downarrow 0} \frac{1}{t}\left[\lim_{s\to 0}\frac{f(y+td)-f(y+td-sd)}{s} - \lim_{s\to 0}\frac{f(y)-f(y-sd)}{s}\right] \\ &= \limsup_{y\to x, t\downarrow 0} \left[\lim_{s\to 0}\frac{f(y+td)-f(y+(t-s)d)-f(y)+f(y-sd)}{ts}\right] \\ &= \limsup_{y\to x, t\downarrow 0} \left[\frac{f(y+td)-2f(y)+f(y-td)}{t^2}\right], \end{aligned}$$

where the last equation follows from letting s approach zero as t does (which is allowable, since the limit as $s \to 0$ exists and is independent of how it is approached). □

THEOREM 5.3. *Let $\hat{x}$ be the limit of a refining subsequence, and let $D(\hat{x})$ be the set of refining directions for $\hat{x}$. Under Assumptions* A1–A3*, if f is continuously differentiable in a neighborhood of $\hat{x}$, then for every direction $d \in D(\hat{x})$ such that $\pm d \in D(\hat{x})$ and $f'(\cdot;\pm d)$ is Lipschitz near $\hat{x}$, $f^{\circ\circ}(\hat{x};d,d) \geq 0$.*

Proof. From Lemma 5.2, it follows that

$$\begin{aligned} f^{\circ\circ}(\hat{x};d,d) &= \limsup_{y\to \hat{x}, t\downarrow 0} \frac{f(y+td)-2f(y)+f(y-td)}{t^2} \\ &\geq \lim_{k\in K} \frac{f(x_k+\Delta_k d)-2f(x_k)+f(x_k-\Delta_k d)}{\Delta_k^2} \\ &\geq 0, \end{aligned}$$

since $\pm d \in D(\hat{x})$ means that $f(x_k) \leq f(x_k \pm \Delta_k d)$ for all $k \in K$. □

COROLLARY 5.4. *Let $\hat{x}$ be the limit of a refining subsequence, and let $D(\hat{x})$ be the set of refining directions for $\hat{x}$. Under Assumptions* A1–A3*, if f is twice continuously differentiable at $\hat{x}$, then $d^T\nabla^2 f(\hat{x})d \geq 0$ for every direction d satisfying $\pm d \in D(\hat{x})$.*

Proof. This follows directly from Theorem 5.3 and the fact that, when $\nabla^2 f(\hat{x})$ exists, $d^T\nabla^2 f(\hat{x})d = f^{\circ\circ}(x;d,d)$. □

The following example illustrates how a function can satisfy the hypotheses of Theorem 5.3, but not those of Corollary 5.4.

EXAMPLE 5.5. *Consider the strictly convex function $f : \mathbb{R} \to \mathbb{R}$ defined by*

$$f(x) = \begin{cases} x^2 & \text{if } x \geq 0, \\ -x^3 & \text{if } x < 0. \end{cases}$$

GPS will converge to the global minimizer at $x = 0$ from any starting point. The derivative of f is given by

$$f'(x) = \begin{cases} 2x & \text{if } x \geq 0, \\ -3x^2 & \text{if } x < 0. \end{cases}$$

Clearly, f' is (Lipschitz) continuous at all $x \in \mathbb{R}$, satisfying the hypotheses of Theorem 5.3. The second derivative of f is given by

$$f''(x) = \begin{cases} 2 & \text{if } x > 0, \\ -6x & \text{if } x < 0, \end{cases}$$

and it does not exist at $x = 0$. Thus, the hypotheses of Corollary 5.4 *are violated. The conclusion of Theorem* 5.3 *can be verified by examining the Clarke derivatives of f' at $x = 0$:*

$$\begin{aligned} f^{\circ\circ}(0; d, d) &= \limsup_{y\to 0, t\downarrow 0} \frac{f(y+td) - f(y) + f(y-td)}{t^2} \\ &\geq \limsup_{y\to 0, t\downarrow 0} \frac{(y+td)^2 - (2y)^2 + (y-td)^2}{t^2} \\ &= \limsup_{y\to 0, t\downarrow 0} \frac{y^2 + 2ytd + t^2d^2 - 2y^2 + y^2 - 2ytd + t^2d^2}{t^2} \\ &= 2d^2 \geq 0. \end{aligned}$$

5.1. Results for unconstrained problems. For unconstrained problems, recall that if f is twice continuously differentiable at a stationary point x^*, the second-order necessary condition for optimality is that $\nabla^2 f(x^*)$ is positive semidefinite; that is, $v^T \nabla^2 f(x^*) v \geq 0$ for all $v \in \mathbb{R}^n$. The following definition gives a pseudo-second-order necessary condition that is not as strong as the traditional one.

DEFINITION 5.6. *Suppose that x^* is a stationary point of a function $f : \mathbb{R}^n \to \mathbb{R}^n$ that is twice continuously differentiable at x^*. Then f is said to satisfy a* pseudo-second-order necessary condition *at x for an orthonormal basis $V \subset \mathbb{R}^n$ if*

$$v^T \nabla^2 f(x^*) v \geq 0 \quad \forall v \in V. \tag{5.1}$$

We note that (5.1) holds for $-V$ as well; therefore, satisfying this condition means that it holds for a set of "evenly distributed" vectors in $\mathbb{R}^n$.

Now recall that a symmetric matrix is positive semidefinite if and only if it has nonnegative real eigenvalues. The following theorem gives an analogous result for matrices that are positive semidefinite with respect to only an orthonormal basis. We note that this general linear algebra result is independent of the convergence results presented in this paper.

THEOREM 5.7. *Let $B \in \mathbb{R}^{n\times n}$ be symmetric, and let V be an orthonormal basis for $\mathbb{R}^n$. If B satisfies $v^T Bv \geq 0$ for all $v \in V$, then the sum of its eigenvalues is nonnegative. If B also satisfies $v^T Bv > 0$ for at least one $v \in V$, then this sum is positive.*

Proof. Since B is symmetric, its Schur decomposition can be expressed as $B = Q\Lambda Q^T$, where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ and $Q \in \mathbb{R}^{n\times n}$ is an orthogonal matrix whose columns q_i, $i = 1, 2, \ldots, n$, are the orthonormal eigenvectors corresponding to the real eigenvalues λ_i, $i = 1, 2, \ldots, n$. Then for each $v_i \in V$, $i = 1, 2, \ldots, n$,

$$0 \leq v_i^T B v_i = v_i^T Q\Lambda Q^T v_i = \sum_{j=1}^n \lambda_j (Q^T v_i)_j^2 = \sum_{j=1}^n \lambda_j (q_j^T v_i)^2, \tag{5.2}$$

and, since $\{q_j\}_{j=1}^n$ and $\{v_i\}_{i=1}^n$ are both orthonormal bases for $\mathbb{R}^n$, it follows that

$$0 \leq \sum_{i=1}^n v_i^T B v_i = \sum_{i=1}^n \sum_{j=1}^n \lambda_j (q_j^T v_i)^2 = \sum_{j=1}^n \lambda_j \sum_{i=1}^n (v_i^T q_j)^2 = \sum_{j=1}^n \lambda_j \|q_j\|_2^2 = \sum_{j=1}^n \lambda_j. \tag{5.3}$$

To obtain the final result, observe that making just one of the inequalities in (5.2) strict yields a similar strict inequality in (5.3), and the result is proved. □

It is easy to see from this proof that if V happens to be the set of eigenvectors Q of B, then B is positive (semi)definite, since in this case, (5.2) yields $q_j^T v_i = q_j^T q_i = \delta_{ij}$, which means that $\lambda_i \geq (>) 0$.

We now establish pseudo-second-order results for GPS by the following two theorems. The first theorem requires convergence in a finite number of steps, while the second necessitates the use of a more specific set of positive spanning directions.

THEOREM 5.8. *Let V be an orthonormal basis in $\mathbb{R}^n$. Let $\hat{x}$ be the limit of a refining subsequence, and let $D(\hat{x})$ be the set of refining directions for $\hat{x}$. Under Assumption* A1, *if f is twice continuously differentiable at $\hat{x}$, $D_k \supset V$ infinitely often in the subsequence, and $x_k = \hat{x}$ for all sufficiently large k, then f satisfies a pseudo-second-order necessary condition for V at $\hat{x}$.*

Proof. For all $k \in K$ and $d \in D(\hat{x})$, we have $f(x_k + \Delta_k d) \geq f(x_k)$. Furthermore, for all sufficiently large $k \in K$, since $x_k = \hat{x}$, a simple substitution yields $f(\hat{x} + \Delta_k d) \geq f(\hat{x})$ for all $d \in D(\hat{x})$. For each $d \in D(\hat{x})$, Taylor's theorem yields

$$f(\hat{x} + \Delta_k d) = f(\hat{x}) + \Delta_k d^T \nabla f(\hat{x}) + \frac{1}{2}\Delta_k^2 d^T \nabla^2 f(\hat{x}) d + \mathcal{O}(\Delta_k^3).$$

Since Corollary 3.6 ensures that $\nabla f(\hat{x}) = 0$, we have

$$0 \leq f(\hat{x} + \Delta_k d) - f(\hat{x}) = \frac{1}{2}\Delta_k^2 d^T \nabla^2 f(\hat{x}) d + \mathcal{O}(\Delta_k^3),$$

or $d^T \nabla^2 f(\hat{x}) d \geq \mathcal{O}(\Delta_k)$ for all $d \in D(\hat{x})$ and for all sufficiently large $k \in K$. The result is obtained by taking limits of both sides (in K) and noting that $D(\hat{x})$ must contain V. □

THEOREM 5.9. *Let V be an orthonormal basis in $\mathbb{R}^n$. Let $\hat{x}$ be the limit of a refining subsequence, and let $D(\hat{x})$ be the set of refining directions for $\hat{x}$. Under Assumption* A1, *if f is twice continuously differentiable at $\hat{x}$ and $D_k \supseteq V \cup -V$ infinitely often in the subsequence, then f satisfies a pseudo-second-order necessary condition for V at $\hat{x}$. Furthermore, the sum of the eigenvalues of $\nabla^2 f(\hat{x})$ must be nonnegative.*

Proof. Since $D(\hat{x}) \subset D$ is finite, it must contain $V \cup -V$, and the result follows directly from Corollary 5.4 and Definition 5.6. The final result follows directly from the symmetry of $\nabla^2 f(\hat{x})$ and Theorem 5.7. □

The significance of Theorem 5.9 is that if f is sufficiently smooth, then the choice of orthonormal mesh directions at each iteration will ensure that the pseudo-second-order necessary condition is satisfied, and that the sum of the eigenvalues of $\nabla^2 f(\hat{x})$ will be nonnegative. Thus, under the assumptions, GPS cannot converge to any saddle point whose Hessian has eigenvalues that sum to less than zero.

These saddle points (to which GPS cannot converge) are those which have sufficiently large regions (cones) of negative curvature. To see this, consider the contrapositive of Theorem 5.7 applied to the Hessian at the limit point; namely, if the sum of the eigenvalues of $\nabla^2 f(\hat{x})$ is negative, then for *any* orthonormal basis $V \in \mathbb{R}^n$, at least one vector $v \in V$ must lie in a cone of negative curvature (i.e., $v^T \nabla^2 f(\hat{x}) v < 0$). Since the angle between any two of these orthogonal directions is 90 degrees, there must be a cone of negative curvature with an angle greater than 90 degrees.

The following example shows that even for orthonormal mesh directions, it is still possible to converge to a saddle point—even when not starting there. It also illustrates our assertion about cones of negative curvature.

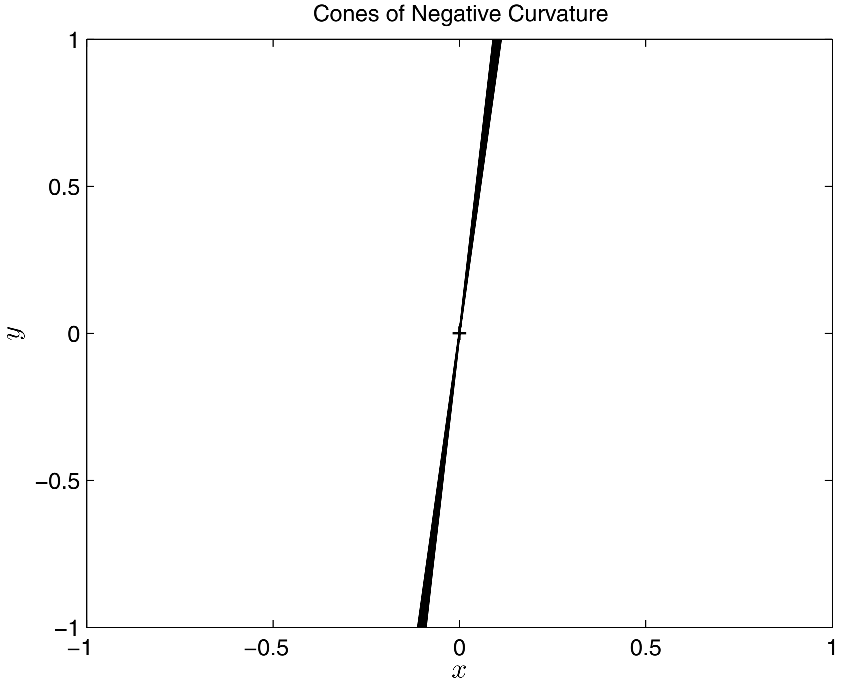


FIG. 5.1. *For $f(x,y) = (9x - y)(11x - y)$, the cones of negative curvature at the saddle point $(0,0)$ are shown in the shaded area between the lines $y = 9x$ and $y = 11x$.*

EXAMPLE 5.10. *Let $f : \mathbb{R}^2 \to \mathbb{R}$ be the twice continuously differentiable function defined by*

$$f(x,y) = 99x^2 - 20xy + y^2 = (9x - y)(11x - y). \tag{5.4}$$

Choose $(x_0, y_0) = (1,1)$ as the initial point, and set $D = \{e_1, e_2, -e_1, -e_2\}$, where e_1 and e_2 are the standard coordinate directions. Now observe that at the saddle point $(0,0)$, directions of negative curvature lie only between the lines $y = 9x$ and $y = 11x$. Thus, to avoid the saddle point, the GPS sequence would have to include a point inside the narrow cone formed by these two lines, when sufficiently close to the origin. If the SEARCH *step is empty, and the polling directions in D are chosen consecutively in the* POLL *step (i.e., we poll in the order $e_1, e_2, -e_1, -e_2$), then the iteration sequence arrives* exactly *at the origin after* 10 *iterations and remains there because none of the directions in D point inside of a cone of negative curvature. Figure* 5.1 *shows the cones of negative curvature for f near the saddle point. Note that these cones, depicted in the shaded areas, are very narrow compared to those of positive curvature. Thus, for the search directions in D, it will be difficult to yield a trial point inside one of these cones.*

On the other hand, if our objective function were $-f$, then the cones of negative curvature would be depicted by the nonshaded areas. In this case, Theorem 5.7 *ensures that GPS cannot converge to the saddle point, since any set of* $2n$ *orthonormal directions would generate a trial point inside one of these cones and thus a lower function value than that of the saddle point.*

5.2. Results for linearly constrained problems. We now treat the linear constrained problem given in (1.1). At this point, we note that there are two equivalent formulations for the classical KKT first-order necessary conditions for optimality, one of which is imbedded in Theorem 3.5. It states that a point x^* satisfies the first-order necessary conditions if $\nabla f(x^*)^T w \geq 0$ for all directions w in the tangent cone $T_X(x^*)$ at x^*, and $-\nabla f(x^*)$ lies in the normal cone $N_X(x^*)$ at x^*. However, since we do not have such a straightforward description of a second-order necessary condition in this form, we now give a more traditional form of the KKT necessary conditions, from which we will be able to establish a sensible pseudo-second-order condition. The following lemma, given without proof, is taken from a well-known textbook [26].

LEMMA 5.11. *If x^* is a local solution of* (1.1)*, then for some vector λ of Lagrange multipliers,*

1. $\nabla f(x^*) = A^T\lambda$*, or, equivalently,* $W^T\nabla f(x^*) = 0$*;*
2. $\lambda \geq 0$*;*
3. $\lambda^T(Ax^* - b) = 0$*;*
4. $W^T\nabla^2 f(x^*)W$ *is positive semidefinite,*

where the columns of W form a basis for the null-space of the active constraints at x^*.

The first three conditions of Lemma 5.11 are generally referred to as first-order necessary conditions, while the last is the second-order necessary condition. Convergence of a subsequence of GPS iterates to a point satisfying first-order conditions has been proved previously [7, 22] and is summarized in Theorem 3.5. Based on the second-order condition, we now provide a pseudo-second-order necessary condition for linearly constrained problems that is analogous to that given in Definition 5.6 for unconstrained problems.

DEFINITION 5.12. *For the optimization problem given in* (1.1)*, let W be an orthonormal basis for the null-space of the binding constraints at x^*, where x^* satisfies the KKT first-order necessary optimality conditions, and f is twice continuously differentiable at x^*. Then f is said to satisfy a* pseudo-second-order necessary condition for W at x^* *if*

$$w^T\nabla^2 f(x^*)w \geq 0 \ \forall w \in W. \tag{5.5}$$

The following theorem shows that the condition given in (5.5) has an equivalent reduced Hessian formulation similar to Definition 5.6. It is formulated to be a general linear algebra result, independent of the GPS algorithm.

THEOREM 5.13. *Let $B \in \mathbb{R}^{n\times n}$ be symmetric, and let $W \in \mathbb{R}^{n\times p}$ be a matrix with orthonormal columns $\{w_i\}_{i=1}^p$, where $p \leq n$. Then the following two statements are equivalent:*

1. $w_i^T B w_i \geq 0,\ i = 1, 2, \ldots, p.$
2. *There exists a matrix Y whose columns $\{y_i\}_{i=1}^p$ form an orthonormal basis for $\mathbb{R}^p$ such that $y_j^T W^T B W y_j \geq 0,\ j = 1, 2, \ldots, p$.*

Proof. Suppose $w_i^T B w_i \geq 0$, $i = 1, 2, \ldots, p$. Then $e_i^T B e_i \geq 0$, $i = 1, 2, \ldots, p$, and the result holds since $\{e_i\}_{i=1}^p$ are orthonormal.

Conversely, suppose there exists $Y \in \mathbb{R}^{p\times p}$ such that $y_j^T W^T B W y_j \geq 0$, $j = 1, 2, \ldots, p$. Let $Z = WY$ with columns $\{z_i\}_{i=1}^p$. Then for $i = 1, 2, \ldots, p$, we have $z_i^T B z_i = y_i^T W^T B W y_i \geq 0$. Furthermore, the columns of Z are orthonormal, since $z_i^T z_j = (Wy_i)^T(Wy_j) = y_i^T W^T W y_j = y_i^T y_j = \delta_{ij}$ (the last step by the orthogonality of Y). □

Assumptions A2–A3 ensure that the GPS algorithm chooses directions that conform to X [7, 21, 22]. This means that the finite set T of all tangent cone generators for all points $x \in X$ must be a subset of D, and that if an iterate x_k is within $\epsilon > 0$ of a constraint boundary, then certain directions in T must be included in D_k. An algorithm that identifies these directions $T_k \subseteq T$, in the nondegenerate case, is given in [22], where it is noted that T_k is chosen so as to contain a (positive) basis for the null-space of the ϵ-active constraints at x_k. Thus, the set of refining directions $D(\hat{x})$ will always contain tangent cone generators at $\hat{x}$, a subset of which forms a basis for null-space of the active constraints at $\hat{x}$. We will denote this null-space by $\mathcal{N}(\hat{A})$, where $\hat{A}$ is the matrix obtained by deleting the rows of A corresponding to the nonactive constraints at $\hat{x}$.

However, in order to exploit the theory presented here we require the following additional assumption so that $D(\hat{x})$ will always contain an orthonormal basis for $\mathcal{N}(\hat{A})$.

A4: The algorithm that computes the tangent cone generators at each iteration includes an orthonormal basis for the null-space of the ϵ-active constraints at each iterate.

Furthermore, since $\mathcal{N}(\hat{A})$ contains the negative of any vector in the space, we can prove convergence to a point satisfying a pseudo-second-order necessary condition by including $T_k \cup -T_k$ in each set of directions D_k.

The next theorem establishes convergence of a subsequence of GPS iterates to a point satisfying a pseudo-second-order necessary condition, similar to that of Theorem 5.8 under the fairly strong condition that convergence occurs in a finite number of steps.

Theorem 5.14. *Let V be an orthonormal basis for $\mathbb{R}^n$. Let $\hat{x}$ be the limit of a refining subsequence, and let $D(\hat{x})$ be the set of refining directions for $\hat{x}$. Under Assumptions* A1–A4*, if f is twice continuously differentiable at $\hat{x}$, and, for all sufficiently large k, $D_k \supset V \cup T_k$ and $x_k = \hat{x}$, then f satisfies a pseudo-second-order necessary condition for some orthonormal basis of $\mathcal{N}(\hat{\mathcal{A}})$ at $\hat{x}$.*

Proof. For all $k \in K$ and $d \in D(\hat{x})$, we have $f(x_k + \Delta_k d) \geq f(x_k)$. Furthermore, for all sufficiently large $k \in K$, since $x_k = \hat{x}$, a simple substitution yields $f(\hat{x}+\Delta_k d) \geq f(\hat{x})$ for all $d \in D(\hat{x})$. For each $d \in D(\hat{x})$, Taylor's theorem yields

$$f(\hat{x} + \Delta_k d) = f(\hat{x}) + \Delta_k d^T \nabla f(\hat{x}) + \frac{1}{2}\Delta_k^2 d^T \nabla^2 f(\hat{x}) d + \mathcal{O}(\Delta_k^3).$$

For $d \in \mathcal{N}(\hat{A})$, Lemma 5.11 ensures that $d^T \nabla f(\hat{x}) = 0$, and thus

$$0 \leq f(\hat{x} + \Delta_k d) - f(\hat{x}) = \frac{1}{2}\Delta_k^2 d^T \nabla^2 f(\hat{x}) d + \mathcal{O}(\Delta_k^3),$$

or $d^T \nabla^2 f(\hat{x}) d \geq \mathcal{O}(\Delta_k)$ for all $d \in D(\hat{x}) \cap \mathcal{N}(\hat{A})$ and for all sufficiently large $k \in K$. The result is obtained by taking limits of both sides (in K), since $D(\hat{x})$ must contain an orthonormal basis for $\mathcal{N}(\hat{A})$. ◻

In the theorem that follows, we show that, given sufficient smoothness of f, if mesh directions are chosen in a fairly standard way, a subsequence of GPS iterates converges to a point satisfying a pseudo-second-order necessary condition. The theorem is similar to Theorem 5.9. Once again, the corollary to this theorem identifies an entire class of saddle points to which GPS cannot converge.

Theorem 5.15. *Let V be an orthonormal basis for $\mathbb{R}^n$. Let $\hat{x}$ be the limit of a refining subsequence, and let $D(\hat{x})$ be the set of refining directions for $\hat{x}$. Under*

Assumptions A1–A4*, if f is twice continuously differentiable at $\hat{x}$ and $D_k \supset V \cup -V \cup T_k \cup -T_k$ infinitely often in the subsequence, then f satisfies a pseudo-second-order necessary condition for some orthonormal basis of $\mathcal{N}(\hat{A})$ at $\hat{x}$.*

Proof. From the discussion following Theorem 5.13, $D(\hat{x})$ contains an orthonormal basis W for $\mathcal{N}(\hat{A})$. Since D is finite, for infinitely many k, we have $-W \subseteq -T_k \subset D_k$, which means that $-W \subseteq D(\hat{x})$. Thus, $D(\hat{x}) \supseteq W \cup -W$, where W is an orthonormal basis for $\mathcal{N}(\hat{A})$, and the result follows from Corollary 5.4 and Definition 5.12. □

COROLLARY 5.16. *If hypotheses of Theorem* 5.15 *hold, then the sum of the eigenvalues of the reduced Hessian $W^T \nabla^2 f(\hat{x}) W$ is nonnegative, where the columns of W form a basis for the null-space of the active constraints at $\hat{x}$.*

Proof. Theorem 5.15 ensures that the pseudo-second-order condition holds; i.e., $w^T \nabla^2 f(\hat{x}) w \geq 0$ for all $w \in W$. Then for $i = 1, 2, \ldots, |W|$, $e_i^T W^T \nabla^2 f(\hat{x}) W e_i \geq 0$, where e_i denotes the ith coordinate vector in $\mathbb{R}^{|W|}$. Since $W^T \nabla^2 f(\hat{x}) W$ is symmetric and $\{e_i\}_{i=1}^{|W|}$ forms an orthonormal basis for $\mathbb{R}^{|W|}$, the result follows from Theorem 5.7. □

6. Concluding remarks.

Clearly, the class of GPS algorithms can never be guaranteed to converge to a point satisfying classical second-order necessary conditions for optimality. However, we have been able to show the following important results, which are surprisingly stronger than what has been proved for many gradient-based (and some Newton-based) methods:

- Under mild assumptions, GPS can converge to a local maximizer only if it does so in a finite number of steps, and if all the directions used infinitely often are directions of constant function value at the maximizer (Lemma 4.2, Theorem 4.3).
- Under mild assumptions, GPS cannot converge to or stall at a strict local maximizer (Corollary 4.4).
- If f is sufficiently smooth and mesh directions contain an orthonormal basis and its negatives, then a subsequence of GPS iterates converges to a point satisfying a pseudo-second-order necessary condition for optimality (Theorems 5.9 and 5.15).
- If f is sufficiently smooth and mesh directions contain an orthonormal basis and its negatives, then GPS cannot converge to a saddle point at which the sum of the eigenvalues of the Hessian (or reduced Hessian) are negative (Theorem 5.9 and Corollary 5.16).

Thus an important characteristic of GPS is that, given reasonable assumptions, the likelihood of converging to a point that does not satisfy second-order necessary conditions is small.

Acknowledgments. The author wishes to thank John Dennis and Charles Audet for several suggestions and useful discussions, along with two anonymous referees for their suggestions, all of which have improved the presentation.

REFERENCES

[1] M. A. Abramson, *Pattern Search Algorithms for Mixed Variable General Constrained Optimization Problems*, Ph.D. thesis, Tech. report TR02-11, Rice University, Department of Computational and Applied Mathematics, Houston, TX, 2002.

[2] M. A. Abramson, *Mixed variable optimization of a load-bearing thermal insulation system using a filter pattern search algorithm*, Optim. Engrg., 5 (2004), pp. 157–177.

[3] M. A. Abramson, C. Audet, and J. E. Dennis, Jr., *Generalized pattern searches with derivative information*, Math. Program., 100 (2004), pp. 3–25.

[4] C. Audet, *Convergence results for pattern search algorithms are tight*, Optim. Engrg., 5 (2004), pp. 101–122.

[5] C. Audet, A. J. Booker, J. E. Dennis, Jr., P. D. Frank, and D. W. Moore, *A Surrogate-Model-Based Method for Constrained Optimization*, presented at the 8th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA, paper 2000-4891, AIAA, 2000.

[6] C. Audet and J. E. Dennis, Jr., *Pattern search algorithms for mixed variable programming*, SIAM J. Optim., 11 (2000), pp. 573–594.

[7] C. Audet and J. E. Dennis, Jr., *Analysis of generalized pattern searches*, SIAM J. Optim., 13 (2003), pp. 889–903.

[8] C. Audet and J. E. Dennis, Jr., *Mesh Adaptive Direct Search Algorithms for Constrained Optimization*, Tech. report TR04-02, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 2004.

[9] C. Audet and J. E. Dennis, Jr., *A pattern search filter method for nonlinear programming without derivatives*, SIAM J. Optim., 14 (2004), pp. 980–1010.

[10] A. J. Booker, J. E. Dennis, Jr., P. D. Frank, D. B. Serafini, and V. Torczon, *Optimization using surrogate objectives on a helicopter test example*, in Optimal Design and Control, J. Borggaard, J. Burns, E. Cliff, and S. Schreck, eds., Progr. Systems Control Theory, Birkhäuser, Cambridge, MA, 1998, pp. 49–58.

[11] A. J. Booker, J. E. Dennis, Jr., P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset, *A rigorous framework for optimization of expensive function by surrogates*, Structural Optim., 17 (1999), pp. 1–13.

[12] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Classics Appl. Math. 5, SIAM, Philadelphia, 1990.

[13] R. Cominetti and R. Correa, *A generalized second-order derivative in nonsmooth optimization*, SIAM J. Control Optim., 28 (1990), pp. 789–809.

[14] A. R. Conn, N. I. M. Gould, and P. L. Toint, *A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds*, SIAM J. Numer. Anal., 28 (1991), pp. 545–572.

[15] I. D. Coope and C. J. Price, *On the convergence of grid-based methods for unconstrained optimization*, SIAM J. Optim., 11 (2001), pp. 859–869.

[16] C. Davis, *Theory of positive linear dependence*, Amer. J. Math., 76 (1954), pp. 733–746.

[17] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Math. Program., 91 (2002), pp. 239–269.

[18] J.-B. Hiriart-Urruty, J.-J. Strodiot, and V. H. Nguyen, *Generalized Hessian matrix and second-order optimality conditions for problems with $C^{1,1}$ data*, Appl. Math. Optim., 11 (1984), pp. 43–56.

[19] M. Kokkolaras, C. Audet, and J. E. Dennis, Jr., *Mixed variable optimization of the number and composition of heat intercepts in a thermal insulation system*, Optim. Engrg., 2 (2001), pp. 5–29.

[20] R. M. Lewis and V. Torczon, *Rank Ordering and Positive Basis in Pattern Search Algorithms*, Tech. report TR 96-71, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA, 1996.

[21] R. M. Lewis and V. Torczon, *Pattern search algorithms for bound constrained minimization*, SIAM J. Optim., 9 (1999), pp. 1082–1099.

[22] R. M. Lewis and V. Torczon, *Pattern search methods for linearly constrained minimization*, SIAM J. Optim., 10 (2000), pp. 917–941.

[23] R. M. Lewis and V. Torczon, *A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds*, SIAM J. Optim., 12 (2002), pp. 1075–1089.

[24] S. Lucidi, M. Sciandrone, and P. Tseng, *Objective-derivative-free methods for constrained optimization*, Math. Program., 92 (2002), pp. 37–59.

[25] A. L. Marsden, M. Wang, J. E. Dennis, Jr., and P. Moin, *Optimal aeroelastic shape design using the surrogate management framework*, Optim. Engrg., 5 (2004), pp. 235–262.

[26] S. G. Nash and A. Sofer, *Linear and Nonlinear Programming*, McGraw-Hill, New York, 1996.

[27] V. Torczon, *On the convergence of pattern search algorithms*, SIAM J. Optim., 7 (1997), pp. 1–25.

SIAM J. Optim.
Vol. 16, No. 2, pp. 531–547

CONVERGENCE OF THE ITERATES OF DESCENT METHODS FOR ANALYTIC COST FUNCTIONS*

P.-A. ABSIL†, R. MAHONY‡, AND B. ANDREWS§

Abstract. In the early eighties Łojasiewicz [in *Seminari di Geometria* 1982-1983, Università di Bologna, Istituto di Geometria, Dipartimento di Matematica, 1984, pp. 115–117] proved that a bounded solution of a gradient flow for an *analytic* cost function converges to a well-defined limit point. In this paper, we show that the iterates of numerical descent algorithms, for an *analytic* cost function, share this convergence property if they satisfy certain natural descent conditions. The results obtained are applicable to a broad class of optimization schemes and strengthen classical "weak convergence" results for descent methods to "strong limit-point convergence" for a large class of cost functions of practical interest. The result does not require that the cost has isolated critical points and requires no assumptions on the convexity of the cost nor any nondegeneracy conditions on the Hessian of the cost at critical points.

Key words. gradient flows, descent methods, real analytic functions, Łojasiewicz gradient inequality, single limit-point convergence, line search, trust region, Mexican hat

AMS subject classifications. 65K05, 90C26, 37N40, 26E05, 40A05

DOI. 10.1137/040605266

1. Introduction.

Unconstrained numerical optimization schemes can be classified into two principal categories: line-search descent methods and trust-region methods. Seminal work by Goldstein [16] and Wolfe [36] on line-search descent methods introduced easily verifiable bounds on step-size selection that led to weak convergence results ($\lim \|\nabla\phi(x_k)\| = 0$) for a wide class of inexact line-search descent algorithms; see, e.g., [15, Theorem 2.5.1] or [30, Theorem 3.2]. For trust-region methods, classical convergence results guarantee weak convergence ($\lim \|\nabla\phi(x_k)\| = 0$) if the total model decrease is at least a fraction of that obtained at the Cauchy point; see, e.g., [30, Theorem 4.8] or [7, Theorem 6.4.6]. Thus, classical convergence results establish that accumulation points of the sequence of iterates are stationary points of the cost function ϕ. Convergence of the whole sequence to a single limit point is not guaranteed. Curry [8, p. 261] first gave the following intuitive counterexample to the existence of such a result for steepest descent methods with line minimization.

> Let $G(x, y) = 0$ on the unit circle and $G(x, y) > 0$ elsewhere. Outside the unit circle let the surface have a spiral gully making infinitely many turns about the circle. The path[1] C will evidently follow the

*Received by the editors March 16, 2004; accepted for publication (in revised form) April 4, 2005; published electronically December 30, 2005.
http://www.siam.org/journals/siopt/16-2/60526.html

†Department of Mathematical Engineering, Université Catholique de Louvain, 1348 Louvain-la-Neuve, Belgium, and Peterhouse, University of Cambridge, Cambridge CB2 1RD, UK (http://www.inma.ucl.ac.be/~absil/). This author's work was partially supported by the School of Computational Science of Florida State University through a postdoctoral fellowship. Part of this work was done while the author was a Research Fellow with the Belgian National Fund for Scientific Research (Aspirant du F.N.R.S.) at the University of Liège.

‡Department of Engineering, Australian National University, Canberra ACT, 0200, Australia (Robert.Mahony@anu.edu.au).

§Center for Mathematical Analysis, IAS, Australian National University, Canberra ACT, 0200, Australia (Ben.Andrews@anu.edu.au).

[1] The path consists of the sequence of estimates of the numerical method.

gully and have all points of the circle as limit points.[2]

It is possible to prove single limit-point convergence for descent algorithms by exploiting additional properties of the cost that ensure critical points are isolated or impose nondegeneracy conditions on the Hessian of the cost on critical sets [17]. Strong convexity of the cost function guarantees that the global minimum is a unique isolated critical point of the cost function and single limit-point convergence is recovered; see, e.g., Byrd and Nocedal [4] for the BFGS algorithm and Kiwiel and Murty [18] for the steepest descent method. Convergence results obtained by Dunn [12, Theorem 4.3] require a uniform growth condition of ϕ and uniqueness of the minimizer within a certain subset. For a class of approximate trust-region methods, Moré and Sorensen [29, Theorem 4.13] show that if the Hessian of ϕ is nonsingular at an accumulation point x^*, then the whole sequence converges to x^*. Conn et al. [6] (or see [7, Theorem 6.5.2]) show that the same result holds for a class of trust-region methods that ensure a fraction of Cauchy decrease. The capture theorem [1], for a class of line-search methods, shows convergence to a single local minimum x^*, provided x^* is an isolated stationary point of ϕ and the iteration comes sufficiently close to x^*; see also [13].

In this paper, we consider the question of convergence given certain regularity conditions on the cost function considered. The motivation for our study is a result in dynamical systems theory that has only recently become widely recognized. For a generic smooth cost function, the ω-limit set [35, p. 42] of a bounded gradient flow is a connected subset of critical points, and not necessarily a single point [17, Prop. C.12.1]. If the cost function is real analytic,[3] then Łojasiewicz's theorem [25] states that the associated gradient flow converges to a single limit point; see section 2 or the introduction of [20] for an overview of Łojasiewicz's argument. A comprehensive treatment of the continuous-time convergence results with applications in optimization theory is contained in the Diploma thesis [22]. The key to the proof lies in showing that the total length of the solution trajectory to the gradient flow is bounded. The proof utilizes the Łojasiewicz gradient inequality (see Lemma 2.1) which gives a lower bound for the norm of the gradient of ϕ in terms of ϕ itself. Due to the importance of this result in the motivation of our work, we provide a review of this result in the early part of the paper, and go on to present an explicit counterexample that shows that single limit-point convergence cannot be proved in general for C^∞ cost functions.

The main contribution of the paper is to adapt these results to iterates of numerical descent algorithms. We define a pair of descent conditions termed the *strong descent conditions* that characterize the key properties of a sequence of iterates that leads to single limit-point convergence. These conditions are deliberately chosen to be as weak as possible in order to apply to the widest possible class of numerical descent algorithms. For line-search methods, it is sufficient to impose an angle condition and the first Wolfe condition (also known as Armijo's condition). For trust-region methods, we give several easily verified conditions involving the Cauchy point that guarantee that the strong descent conditions hold. The main theorem uses these conditions to prove that the whole sequence of iterates $\{x_k\}$ of a numerical descent algorithm, applied to an analytic cost function, either escapes to infinity (i.e., $\|x_k\| \to +\infty$) or converges to a single limit point. An interesting aspect of the development is that

[2] A point x is a *limit point* or *accumulation point* of a sequence $\{x_k\}_{k\in\mathbb{N}}$ if there exists a subsequence $\{x_{k_i}\}_{i\in\mathbb{N}}$ that converges to x.

[3] A real function is said to be analytic if it possesses derivatives of all orders and agrees with its Taylor series in the neighborhood of every point.

the strong descent conditions themselves do not guarantee convergence to a critical point of the cost, $\|\nabla\phi(x_k)\| \not\to 0$. However, combining single limit-point convergence with classical weak convergence results leads to convergence to a single critical point for a wide range of classical numerical descent algorithms applied to analytic cost functions.

Apart from ensuring continuity of ϕ, the only purpose of the analyticity assumption is to guarantee that the Łojasiewicz gradient inequality holds in a neighborhood of every point. Therefore, the domain of application of our results goes beyond the (already large) class of analytic functions to functions that satisfy a simple growth condition (see (2.7)). If moreover it is known that a point x^* is an accumulation point, then in order to have convergence of the whole sequence to x^* it is sufficient to require that this growth condition holds in a neighborhood of x^*.

A preliminary version of the results presented in this paper appeared in the proceedings of the 13th MTNS conference [28]. Generalizations to Riemannian manifolds have been considered in [22].

The paper is organized as follows. The continuous-time case is reviewed in section 2 and the Mexican hat example is presented. The general convergence theory for descent iterations is developed in section 3 and applied to line-search and trust-region methods in section 4. Conclusions are presented in section 5.

2. Convergence of analytic gradient descent flows. In this section, we briefly review Łojasiewicz's argument for the convergence of analytic gradient flows and give an explicit counterexample to show that single limit-point convergence does not hold for certain C^∞ gradient flows. In the past five years, many authors have revisited the original gradient flow convergence results of Łojasiewicz [25]. Our presentation follows the generalization proposed by Lageman [22], where the steepest descent direction was relaxed to an angle condition. The proof is included to provide motivation for the discrete-time analysis in section 3. A concise presentation of the standard argument for Łojasiewicz's theorem is contained in [20].

Let $\mathbb{R}^n$ be the linear space of column vectors with n components, endowed with the usual inner product $\langle x, y\rangle = x^T y$. Let $\nabla\phi(x) := (\partial_1\phi(x), \dots, \partial_n\phi(x))^T$ denote the Euclidean gradient of the differentiable function ϕ. A point x^* where $\nabla\phi(x^*) = 0$ is called a *stationary point* or *critical point* of ϕ.

The proof of Łojasiewicz's theorem is based on the following property of real analytic functions.

LEMMA 2.1 (Łojasiewicz gradient inequality). [4] *Let ϕ be a real analytic function on a neighborhood of x^* in $\mathbb{R}^n$. Then there are constants $c > 0$ and $\mu \in [0, 1)$ such that*

$$\|\nabla\phi(x)\| \geq c|\phi(x) - \phi(x^*)|^\mu \tag{2.1}$$

in some neighborhood U of x^*.

Proof. See [24, p. 92], [2, Prop. 6.8], or the short proof in [21]. □

THEOREM 2.2. *Let ϕ be a real analytic function and let $x(t)$ be a C^1 curve in $\mathbb{R}^n$, with $\dot{x}(t) = \frac{dx}{dt}(t)$ denoting its time derivative. Assume that there exist a $\delta > 0$*

[4] The Łojasiewicz gradient inequality is a special instance of a more general Łojasiewicz inequality [23, 26]. The latter result has been used in the study of error bounds of analytic inequality systems in optimization [27, 9]. In turn, such error bounds have been used in the convergence analysis of optimization algorithms in the same general spirit as in the present paper; see, e.g., [14, 37]. We thank an anonymous reviewer for pointing this out.

and a real τ such that for $t > \tau$, $x(t)$ satisfies the angle condition

$$\frac{d\phi(x(t))}{dt} \equiv \langle \nabla\phi(x(t)), \dot{x}(t)\rangle \leq -\delta\|\nabla\phi(x(t))\|\|\dot{x}(t)\| \tag{2.2}$$

and a weak decrease condition

$$\left[\frac{d}{dt}\phi(x(t)) = 0\right] \Rightarrow [\dot{x}(t) = 0]. \tag{2.3}$$

Then, either $\lim_{t\to+\infty}\|x(t)\| = \infty$ *or there exists* $x^* \in \mathbb{R}^n$ *such that* $\lim_{t\to+\infty} = x^*$.

Proof. Assume that $\|x(t)\| \nrightarrow +\infty$ as $t \to +\infty$. Then $x(t)$ has an accumulation point x^* in $\mathbb{R}^n$. It remains to show that $\lim_{t\to+\infty} x(t) = x^*$ and the proof will be complete.

It follows from (2.2) that $\phi(x(t))$ is nonincreasing. Moreover, since x^* is an accumulation point of $x(t)$, it follows by continuity of ϕ that

$$\phi(x(t)) \downarrow \phi(x^*).$$

We distinguish two cases.

Case (i). There exists a $t_1 > \tau$ such that $\phi(x(t_1)) = \phi(x^*)$. Since $\phi(x(t))$ is nonincreasing then it is straightforward to see that $\phi(x(t)) = \phi(x^*)$ and $\frac{d}{dt}\phi(x(t)) = 0$ for all $t \geq t_1$. From the weak decrease condition (2.3) this implies that $\dot{x}(t) = 0$ for all $t \geq t_1$ and $x(t) = x(t_1) = x^*$.

Case (ii). $\phi(x(t)) > \phi(x^*)$ for all $t > \tau$. In order to simplify the forthcoming equations we assume without loss of generality that $\phi(x^*) = 0$. It follows from the Łojasiewicz gradient inequality (Lemma 2.1) and from (2.2) that

$$\frac{d\phi(x(t))}{dt} \leq -\delta\|\nabla\phi(x(t))\|\|\dot{x}(t)\| \leq -\delta c|\phi(x(t))|^{\mu}\|\dot{x}(t)\| \tag{2.4}$$

holds in a neighborhood U of x^* for some $\mu \in [0, 1)$. Since we have assumed that $\phi(x(t)) > \phi(x^*) = 0$, it follows from (2.4) that

$$c_1\frac{d(\phi(x(t)))^{1-\mu}}{dt} \leq -\|\dot{x}(t)\|, \tag{2.5}$$

where $c_1 := [\delta c(1-\mu)]^{-1} > 0$. Given t_1 and t_2 with $\tau < t_1 < t_2$, if $x(t) \in U$ for all $t \in (t_1, t_2)$, then by integration of (2.5)

$$L_{12} := \int_{t_1}^{t_2} \|\dot{x}(t)\|dt \leq c_1((\phi(x(t_1)))^{1-\mu} - (\phi(x(t_2)))^{1-\mu}) \leq c_1(\phi(x(t_1)))^{1-\mu}. \tag{2.6}$$

Now let r be such that $B_r(x^*) \subset U$, where

$$B_r(x^*) := \{x \in \mathbb{R}^n : \|x - x^*\| < r\}.$$

We show that $x(t)$ eventually enters and remains in $B_r(x^*)$. Since r is arbitrarily small, it follows that $x(t)$ converges to x^* and the theorem will be proven.

Let t_1 be such that $\|x(t_1) - x^*\| < r/2$ and $c_1\phi^{1-\mu}(x(t_1)) < r/2$. Such a t_1 exists by continuity of ϕ since x^* is an accumulation point of $x(t)$ and $\phi(x^*) = 0$. Then we show that the entire trajectory after t_1 lies in $B_r(x^*)$. By contradiction, suppose not, and let t_2 be the smallest $t > t_1$ such that $\|x(t_2) - x^*\| = r$. Then $x(t)$ lies in U for

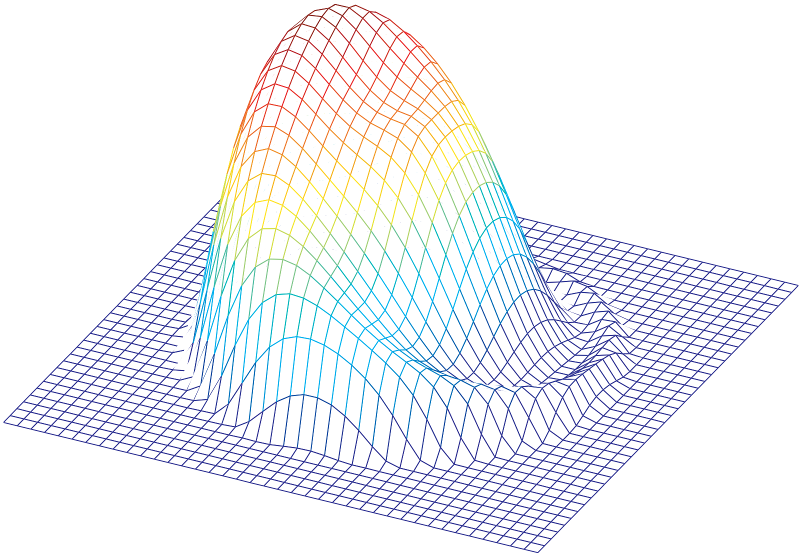


FIG. 2.1. *A plot of the smooth "Mexican hat" function defined in* (2.8).

all $t \in (t_1, t_2)$. Therefore (2.6) holds and it follows that $L_{12} \le c_1(\phi(x(t_1)))^{1-\mu} < r/2$. Then $\|x(t_2) - x^*\| \le \|x(t_2) - x(t_1)\| + \|x(t_1) - x^*\| < L_{12} + r/2 < r$, which is a contradiction. Thus $x(t)$ remains in $B_r(x^*)$ for all $t \in [t_1, +\infty)$, and the proof is complete. □

The role of the weak decrease condition (2.3) is to prevent the trajectory $x(t)$ from wandering endlessly in the critical set $\nabla\phi = 0$. It is possible to weaken this condition somewhat to allow the trajectory to spend finite periods of time wandering in this set as long as it eventually either converges or continues to decrease the cost (see [22]).

Considering Theorem 2.2, a natural question to ask is if it is possible to relax the condition of analyticity on the cost function and retain the convergence results. Clearly, analyticity is principally used to invoke the Łojasiewicz gradient inequality. The rationale goes through if ϕ is continuous at an accumulation point x^* of $x(t)$ and a growth condition of the type

$$\|\nabla\phi(x^*)\| \ge \psi(\phi(x(t)) - \phi(x^*)) \tag{2.7}$$

holds in a neigborhood of x^*, where $1/\psi$ is positive and integrable on an interval $(0, \epsilon)$. In practice, such a growth condition may be difficult to check. This is especially true when no accumulation point is known a priori so that the condition must be verified on a set.

Theorem 2.2 does not hold for the general class of smooth cost functions $\phi \in C^\infty$. It is instructive to provide an explicit counterexample. The following function $f \in C^\infty$ (cf. Figure 2.1) is a smooth example of a "Mexican hat" cost function. Let

$$f(r,\theta) := \begin{cases} e^{-\frac{1}{1-r^2}}\left[1 - \frac{4r^4}{4r^4+(1-r^2)^4}\sin\left(\theta - \frac{1}{1-r^2}\right)\right] & \text{if } r < 1, \\ 0 & \text{if } r \ge 1, \end{cases} \tag{2.8}$$

where (r, θ) denote polar coordinates in $\mathbb{R}^2$.

Since $0 \leq \frac{4r^4}{4r^4+(1-r^2)^4} < 1$ for all $r < 1$, it follows that $f(r,\theta) > 0$ for all $r < 1$. The exponential factor in f ensures that all derivatives at $r = 1$ are well defined (and equal to zero) and it follows that $f \in C^\infty$. The example has been constructed such that, for initial conditions (r_0, θ_0) with $\theta_0(1 - r_0^2) = 1$ and $0 < r_0 < 1$, the solution $(r(t), \theta(t))$ of the gradient descent flow (expressed in polar coordinates) satisfies

$$\theta(t) = \frac{1}{1 - r(t)^2}. \tag{2.9}$$

By inspection, the ω-limit set of the trajectory (2.9) considered is the entire circle $\{(r,\theta) \,|\, r = 1\}$.

The origin of the colloquial name "Mexican hat" function for a counterexample of this form is not clear. Certainly, the structure of the counterexample was known by the time of Curry [8]. Prior examples of Mexican hats were proposed in [38] (mentioned in [1, Exercise 2.18]) and [31, Example 3, p. 13]. The merit of the cost function (2.8) is to provide a closed-form trajectory (2.9) and render the convergence analysis trivial.

3. Convergence of analytic descent iterations.

In this section, a discrete-time analogue of Theorem 2.2 (Łojasiewicz's theorem with an angle condition) is obtained. We propose a pair of "strong descent conditions" that encapsulate the key properties of the iterates of a numerical descent algorithm that lead to single limit-point convergence for an analytic cost function. In later sections we show that the strong descent conditions are satisfied naturally by most numerical descent algorithm iterates.

3.1. Main result.

In the discrete-time case, a solution trajectory is a sequence $\{x_k\}$ instead of a curve $x(t)$. The key to extending the results of section 2 to this case is to adapt the conditions (2.2) and (2.3) to the discrete-time case. For (2.2) we propose a *primary descent condition*:

$$\phi(x_k) - \phi(x_{k+1}) \geq \sigma \|\nabla\phi(x_k)\| \|x_{k+1} - x_k\| \tag{3.1}$$

for all k and for some $\sigma > 0$. Condition (3.1) is satisfied under Armijo's condition (4.4) along with an angle condition (4.2). This fact will be exploited in section 4.1 in the context of line-search methods. Moreover (3.1) is sufficiently general to accomodate the framework of trust-region methods; see section 4.2.

Condition (3.1) itself does not preclude $\{x_k\}$ from endlessly wandering in a critical set of ϕ. To overcome this, we introduce a *complementary descent condition*:

$$[\phi(x_{k+1}) = \phi(x_k)] \Rightarrow [x_{k+1} = x_k]\,. \tag{3.2}$$

This condition simply requires that any nonvanishing update, $x_{k+1} \neq x_k$, produce a change in the cost function. Condition (3.2) adds information to (3.1) only when x_k is a critical point (i.e., $\nabla\phi(x_k) = 0$). Note that conditions (3.1) and (3.2) allow the sequence $\{x_k\}$ to stagnate for arbitrarily many iterations, a behavior observed, e.g., in trust-region methods when the model estimate turns out to be so poor that the proposed update is rejected (see section 4.2). Together, we term conditions (3.1) and (3.2) the *strong descent conditions*.

DEFINITION 3.1 (strong descent conditions). *We say that a sequence* $\{x_k\}$ *in* $\mathbb{R}^n$ *satisfies the* strong descent conditions *if* (3.1) *and* (3.2) *hold for some* $\sigma > 0$ *and for all* k *larger than some* K.

The main result (Theorem 3.2 below) shows that if the iterates $\{x_k\}$ of a numerical descent algorithm satisfy the strong descent conditions (Definition 3.1) and the cost function ϕ is analytic, then $\{x_k\}$ converges to a single point or diverges to infinity. Note that we do not claim that the limit point is a stationary point of ϕ; indeed, the assumptions are not strong enough (in particular, they do not preclude stagnation). For classical descent algorithms, convergence to a stationary point can be obtained by invoking classical weak convergence results ($\nabla\phi \to 0$) in combination with Theorem 3.2.

THEOREM 3.2 (main result). *Let $\phi : \mathbb{R}^n \mapsto \mathbb{R}$ be an analytic cost function. Let the sequence $\{x_k\}_{k=1,2,\dots}$ satisfy the strong descent conditions (Definition* 3.1*). Then, either* $\lim_{k\to\infty} \|x_k\| = +\infty$*, or there exists a single point $x^* \in \mathbb{R}^n$ such that*

$$\lim_{k\to\infty} x_k = x^*.$$

Proof. Without loss of generality, discard all iterates up to the K iterate and relabel the sequence, such that (3.1) and (3.2) hold explicitly on the new sequence. Assume moreover that $\|x_k\| \nrightarrow \infty$; i.e., $\{x_k\}$ has at least one accumulation point x^* in $\mathbb{R}^n$. It is sufficient to show that $\lim_{k\to+\infty} x_k = x^*$ to complete the proof.

For simplicity, we assume without loss of generality that $\phi(x^*) = 0$. If the sequence $\{x_k\}$ is eventually constant (i.e., there exists a K such that $x_k = x_K$ for all $k > K$), then the result follows directly. For the remaining case we remove from the sequence all the x_k's such that $x_{k+1} = x_k$ and we renumber the sequence accordingly. It follows that the new sequence is infinite, never stagnates, and admits the same limit set as the original sequence. By continuity of ϕ, since x^* is an accumulation point of $\{x_k\}$ and $\phi(x_k)$ is strictly decreasing as a consequence of (3.2), it follows that

$$\phi(x_0) > \phi(x_1) > \cdots > 0. \tag{3.3}$$

(Note that this is the only place in this proof where (3.2) is utilized.)

It then follows from the Łojasiewicz gradient inequality (Lemma 2.1) and the primary descent condition (3.1) that, in some neighborhood U of x^*,

$$\phi(x_k) - \phi(x_{k+1}) \geq \sigma\|\nabla\phi(x_k)\|\|x_{k+1} - x_k\| \geq \sigma c|\phi(x_k)|^\mu \|x_{k+1} - x_k\|.$$

That is, since we have shown that $\phi(x_k) > 0$ for all k,

$$\|x_{k+1} - x_k\| \leq \frac{\phi(x_k) - \phi(x_{k+1})}{\sigma c\,(\phi(x_k))^\mu}, \tag{3.4}$$

provided x_k belongs to U.

Since $\mu \in [0, 1)$, it follows from (3.3) that $\frac{1}{(\phi(x_k))^\mu} \leq \frac{1}{\phi^\mu}$ for all ϕ in the interval $[\phi(x_{k+1}), \phi(x_k)]$, and therefore

$$\begin{aligned}\frac{\phi(x_k) - \phi(x_{k+1})}{(\phi(x_k))^\mu} &= \int_{\phi(x_{k+1})}^{\phi(x_k)} \frac{1}{(\phi(x_k))^\mu} d\phi \leq \int_{\phi(x_{k+1})}^{\phi(x_k)} \frac{1}{\phi^\mu} d\phi \\ &= \frac{1}{1-\mu}\left((\phi(x_k))^{1-\mu} - (\phi(x_{k+1}))^{1-\mu}\right).\end{aligned} \tag{3.5}$$

Substituting (3.5) into (3.4) yields

$$\|x_{k+1} - x_k\| \leq \frac{1}{\sigma c(1-\mu)}\left((\phi(x_k))^{1-\mu} - (\phi(x_{k+1}))^{1-\mu}\right). \tag{3.6}$$

This bound plays a role similar to the bound (2.5) on the exact derivative obtained in the continuous-time case.

Given $k_2 > k_1$ such that the iterates x_{k_1} up to x_{k_2-1} belong to U, we have

$$\sum_{k=k_1}^{k_2-1} \|x_{k+1} - x_k\| \leq c_1 \left((\phi(x_{k_1}))^{1-\mu} - (\phi(x_{k_2}))^{1-\mu}\right) \leq c_1(\phi(x_{k_1}))^{1-\mu}, \tag{3.7}$$

where $c_1 = [\sigma c(1-\mu)]^{-1}$. This bound plays the same role as (2.6).

Now the conclusion comes much as in the proof of Theorem 2.2. Let $r > 0$ be such that $B_r(x^*) \subset U$, where

$$B_r(x^*) = \{x \in \mathbb{R}^n : \|x - x^*\| < r\}$$

is the open ball of radius r centered at x^*. Let k_1 be such that $\|x_{k_1} - x^*\| < r/2$ and $c_1(\phi(x_{k_1}))^{1-\mu} < r/2$. Such a k_1 exists since x^* is an accumulation point and $\phi(x^*) = 0$. Then we show that $x_{k_2} \in B_r(x^*)$ for all $k_2 > k_1$. By contradiction, suppose not, and let K be the smallest $k > k_1$ such that $\|x_K - x^*\| \geq r$. Then x_k remains in U for $k_1 \leq k < K$, so it follows from (3.7) that $\sum_{k=k_1}^{K-1} \|x_{k+1} - x_k\| \leq c_1(\phi(x_{k_1}))^{1-\mu} < r/2$. It then follows that $\|x_K - x^*\| \leq \|x_K - x_{k_1}\| + \|x_{k_1} - x^*\| \leq \sum_{k=k_1}^{K-1} \|x_{k+1} - x_k\| + \|x_{k_1} - x^*\| < \frac{r}{2} + \frac{r}{2} \leq r$. But we have supposed that $\|x_K - x^*\| \geq r$, which is a contradiction.

We have thus shown that, given r sufficiently small, there exists k_1 such that $\|x_{k_2} - x^*\| < r$ for all $k_2 > k_1$. Since $r > 0$ is arbitrary (subject to $B_r(x^*) \subset U$), this means that the whole sequence $\{x_k\}$ converges to x^*, and the proof is complete. (The same conclusion follows by noting that the "length" $\sum_{k=1}^{+\infty} \|x_{k+1} - x_k\|$ is finite.) □

3.2. Discussion. We now comment on Theorem 3.2 and propose a few variations and extensions to this result.

3.2.1. C^∞ is not sufficient to guarantee single limit-point convergence. Similar to the continuous-time case, it is natural to wonder whether the analyticity assumption on ϕ can be relaxed to indefinite differentiability ($\phi \in C^\infty$). The answer is again negative: as we now show, there exist a sequence $\{x_k\}$ in $\mathbb{R}^n$ and a function ϕ in C^∞ such that $\{x_k\}$ satisfies the strong descent conditions (Definition 3.1) and nevertheless the limit set of $\{x_k\}$ contains more than one point of $\mathbb{R}^n$.

Consider the Mexican hat function (2.8) and let $x_k = (r_k \cos\theta_k, r_k \sin\theta_k)^T$ with $\theta_k = k\omega$ and $r_k = \sqrt{(\theta_k - 1)/\theta_k}$, so that x_k belongs to the trajectory given by (2.9). Choose $\omega > 0$ such that ω/π is an irrational number. Then the limit set of $\{x_k\}$ is the unit circle in $\mathbb{R}^2$. However, f is C^∞ and the primary descent condition (3.1) is satisfied for $\sigma = \frac{1-e^{-\omega}}{4}$. Indeed, simple manipulations yield

$$\begin{aligned}
\partial_r f(r_k, \theta_k) &= -e^{-\frac{1}{1-r_k^2}} \tfrac{2r_k(1-r_k^2)^2}{4r_k^4+(1-r_k^2)^4}, \\
\tfrac{1}{r}\partial_\theta f(r_k, \theta_k) &= -e^{-\frac{1}{1-r_k^2}} \tfrac{4r_k^3}{4r_k^4+(1-r_k^2)^4}, \\
\|\nabla_x f(x_k)\| &= e^{-\theta_k} \tfrac{2r_k}{4r_k^4+(1-r_k^2)^4} \sqrt{(1-r_k^2)^4 + 4r_k^4}.
\end{aligned}$$

Thus $\|\nabla_x f(x_k)\| \leq 2e^{-k\omega}$ when r_k is sufficiently close to 1, i.e., when k is sufficiently large. Thus

$$\begin{aligned} f(x_k) - f(x_{k+1}) = e^{-k\omega} - e^{-(k+1)\omega} = e^{-k\omega}(1 - e^{-\omega}) &\geq \tfrac{1-e^{-\omega}}{4} 2e^{-k\omega}\, 2 \\ &\geq \tfrac{1-e^{-\omega}}{4} \|\nabla f(x_k)\| \|x_k - x_{k+1}\|. \end{aligned}$$

3.2.2. Ruling out escape to infinity. There are several ways to rule out the case $\lim_{k\to+\infty} \|x_k\| = \infty$ in Theorem 3.2. Convergence results often assume that ϕ has compact sublevel sets, in which case $\{x_k\}$ is bounded. Note also that $\lim_{k\to\infty} \|x_k\| = +\infty$ occurs if and only if $\{x_k\}$ has no accumulation point in $\mathbb{R}^n$.

It is interesting to consider what happens in the close vicinity of a critical point. Proposition 3.3 guarantees that if the iteration starts close enough to a local minimum x^* of ϕ, and if the complementary descent condition (3.2) is replaced by a termination condition, then the sequence of iterates stays in a neighborhood of x^*. Strengthening the weak descent condition to condition (3.9) is required since it is not possible to center the analysis at an accumulation point as was done in the proof of Theorem 3.2.

PROPOSITION 3.3 (Lyapunov stability of minima). *Let x^* be a (possibly nonstrict) local minimum of the analytic cost function ϕ. Let*

$$x_{k+1} = F(x_k) \tag{3.8}$$

be a discrete-time dynamical system satisfying the primary descent condition (3.1) *and the termination condition*

$$\nabla\phi(x_k) = 0 \Rightarrow \textit{terminate.} \tag{3.9}$$

Then x^* is Lyapunov-stable for (3.8). *That is, given $\epsilon > 0$, there exists $\delta > 0$ such that*

$$\|x_0 - x^*\| \leq \delta \Rightarrow \|x_k - x^*\| \leq \epsilon \quad \textit{for all } k.$$

Proof. Without loss of generality, we again assume that $\phi(x^*) = 0$. Let U_m be a neighborhood of x^* such that $\phi(x) \geq \phi(x^*)$ for all $x \in U_m$. Let U_{L} be a neighborhood of x^* where the Łojasiewicz inequality (Lemma 2.1) holds. Let ϵ be such that $B_\epsilon(x^*) \subset U_m \cap U_{\text{L}}$. Let $\delta < \epsilon/2$ be such that $c_1(\phi(x))^{1-\mu} < \epsilon/2$ for all $x \in B_\delta(x^*)$, where $c_1 = [\sigma c(1-\mu)]^{-1}$ and c, μ, and σ are the constants appearing in the Łojasiewicz inequality and the primary descent condition (3.1). Then we show that x_k belongs to $B_\epsilon(x^*)$ and the proof is complete. By contradiction, suppose that x_k eventually leaves $B_\epsilon(x^*)$. Let K be the smallest k such that x_k is not in $B_\epsilon(x^*)$. We dismiss the trivial case where the algorithm terminates. Thus $\nabla\phi(x_k) \neq 0$ for all $k < K$. It follows that $\phi(x_k) > 0$ for all $k < K$; otherwise the assumption on U_m would not hold. The rationale given in the proof of Theorem 3.2 yields that

$$\|x_k - x_0\| \leq c_1(\phi(x_0))^{1-\mu} < \epsilon/2$$

for all $k \leq K$, and it follows from the triangle inequality that $\|x_K - x^*\| < \epsilon$, which is a contradiction. □

Note that Proposition 3.3 is a Lyapunov stability result and one must prove local attractivity of x^* in addition to Proposition 3.3 to prove asymptotic stability of x^*. It would be sufficient to additionally require weak convergence of the iterates (i.e., $\nabla\phi(x_k) \to 0$) and that x^* is an isolated stationary point of ϕ. A similar result is given by the capture theorem [1].

3.2.3. A stronger result. The proof of Theorem 3.2 does not use the analyticity of ϕ to its full extent. Instead, the proof requires only that ϕ be continuous at the accumulation point x^* and that the Łojasiewicz gradient inequality hold in a neighborhood of x^*.

There is a large class of functions that are not real analytic but nevertheless satisfy the Łojasiewicz gradient inequality; see, e.g., [19, 3]. As an illustration, consider $\phi(x) = f(g(x))$, where g is real analytic and f is C^1. Assume for simplicity that $g(x^*) = 0$ and $f(0) = 0$, and so $\phi(x^*) = 0$. Assume moreover that $f'(0) = c_1 > 0$, where f' denotes the first derivative of f. Since $f' \circ g$ is continuous, it follows that there exists a neighborhood U of x^* such that $f'(g(x)) > \frac{c_1}{2}$ for all $x \in U$. Shrinking U if necessary, it follows from the Łojasiewicz gradient inequality on g that there are constants $c > 0$ and $\mu \in [0, 1)$ such that $\|\nabla\phi(x)\| = |f'(g(x))| \cdot \|\nabla g(x)\| \geq \frac{c_1}{2}\|\nabla g(x)\| \geq \frac{c_1}{2}c|g(x)|^\mu$ for all $x \in U$. Shrinking U further if necessary, since $f'(0) = c_1 > 0$, $f(0) = 0$, and $f \in C^1$, we have $|g(x)| \geq |f(g(x))|/(2c_1)$ for all $x \in U$. Consequently, $\|\nabla\phi(x)\| \geq \frac{c_1 c}{2(2c_1)^\mu}|\phi(x)|^\mu$ for all $x \in U$, and this is a Łojasiewicz inequality.

It is interesting to consider what would be the weakest general condition on the cost function that would ensure single limit-point convergence of a descent iteration under the strong descent conditions (Definition 3.1). In general, this question is difficult to answer; however, the following result provides the weakest condition on the cost function such that the proof given for Theorem 3.2 applies. Note that the class of functions covered is again larger than those satisfying the Łojasiewicz gradient inequality, shown earlier to be a superset of analytic functions.

THEOREM 3.4. *Let x^* be a point of $\mathbb{R}^n$ and let ϕ be a cost function on $\mathbb{R}^n$ continuous at x^*. Assume that there exist a neighborhood U of x^*, an $\epsilon > 0$, and a nondecreasing strictly positive function $\psi : (0, \epsilon) \to \mathbb{R}$ such that $1/\psi$ is integrable over $(0, \epsilon)$ and*

$$\|\nabla\phi(x)\| \geq \psi(\phi(x) - \phi(x^*))$$

for all x in $\{x \in U : 0 < \phi(x) - \phi(x^*) < \epsilon\}$. Consider a sequence $\{x_k\}$ satisfying the strong descent conditions (Definition 3.1*) and assume that x^* is an accumulation point of $\{x_k\}$. Then* $\lim_{k\to\infty} x_k = x^*$.

4. Application to classical optimization schemes.

In this section, we show that the strong descent conditions (Definition 3.1) hold for a wide variety of numerical optimization methods. Consequently, these methods have single limit-point convergence when the cost function is analytic, or more generally when the conditions of Theorem 3.4 are satisfied. We will successively consider methods of the line-search type and of the trust-region type. References on numerical optimization include [10, 15, 1, 30, 7].

4.1. Convergence of line-search methods.

Any line-search method proceeds in two steps. First, the algorithm chooses a search direction p_k from the current iterate x_k. Then the algorithm searches along this direction for a new iterate

$$x_{k+1} = x_k + \alpha_k p_k \tag{4.1}$$

satisfying some criteria.

We first consider the choice of the search direction p_k. An obvious choice is the steepest descent direction $p_k = -\nabla\phi(x_k)$, which is often relaxed to a direction p_k

satisfying an angle condition

$$\frac{\langle p_k, \nabla\phi(x_k)\rangle}{\|p_k\|\|\nabla\phi(x_k)\|} = \cos\theta_k \leq -\delta < 0; \tag{4.2}$$

i.e., the angle between p_k and $-\nabla\phi(x_k)$ is bounded away from 90°. A wide variety of optimization schemes obtain the search direction by solving an equation of the form

$$B_k p_k = -\nabla\phi(x_k). \tag{4.3}$$

In particular, the choice $B_k = \nabla^2\phi(x_k)$, the Hessian of ϕ at x_k, yields the Newton direction. When some approximation of the Hessian is used, p_k is called a quasi-Newton search direction. From (4.2) and (4.3), standard manipulations (see, e.g., [30, p. 45]) yield $\cos\theta_k \geq 1/\kappa(B_k)$, where $\kappa(B_k) = \|B_k\|\,\|B_k^{-1}\|$ is the condition number of B_k. Therefore, for the angle condition (4.2) to hold true with (4.3), it is sufficient that the condition number of B_k be bounded.

Now consider the choice of α_k in (4.1). A very usual condition on α is the first Wolfe condition, also known as the *Armijo condition* (see, e.g., [30]):

$$\phi(x_k) - \phi(x_{k+1}) \geq -c_1\langle\nabla\phi(x_k), x_{k+1} - x_k\rangle, \tag{4.4}$$

where $c_1 \in (0,1)$ is a constant. The Armijo condition is satisfied for all sufficiently small values of α_k. Therefore, in order to ensure that the algorithm makes sufficient progress, it is usual to require moreover that, for some constant $c_2 \in (c_1, 1)$,

$$\langle\nabla\phi(x_{k+1}), x_{k+1} - x_k\rangle \geq c_2\langle\nabla\phi(x_k), x_{k+1} - x_k\rangle, \tag{4.5}$$

known as the *curvature condition.* Conditions (4.4) and (4.5) are known collectively as the *Wolfe conditions.* Several schemes exist that compute an α_k such that the Wolfe conditions hold; see, e.g., [1, 30].

THEOREM 4.1. (i) *Consider the line-search descent algorithm given by* (4.1). *Let the algorithm terminate if* $\nabla\phi(x_k) = 0$. *Assume that the search direction* p_k *satisfies the angle condition* (4.2). *Let the step-size be selected such that the Armijo condition* (4.4) *holds. Then the strong descent conditions (Definition* 3.1*) hold.*

(ii) *Assume moreover that the cost function* ϕ *is analytic. Then either* $\lim_{k\to\infty}\|x_k\| = +\infty$*, or there exists a single point* $x^* \in \mathbb{R}^n$ *such that*

$$\lim_{k\to\infty} x_k = x^*.$$

(iii) *In the latter case, if moreover the curvature condition* (4.5) *holds, then* x^* *is a stationary point of* ϕ*, i.e.,*

$$\nabla\phi(x^*) = 0.$$

Proof. (i) Combining the angle condition (4.2) and the Armijo condition (4.4) yields $\phi(x_k) - \phi(x_{k+1}) \geq c_1\delta\|\nabla\phi(x_k)\|\|x_{k+1} - x_k\|$, i.e., the primary descent condition (3.1) with $\sigma = c_1\delta$. The complementary descent condition (3.2) is also satisfied: if $\nabla\phi(x_k) = 0$, then the algorithm terminates and if $\nabla\phi(x_k) \neq 0$, then (3.2) follows from (3.1).

(ii) The proof is direct from (i) and Theorem 3.2.

(iii) The proof is a direct consequence of (ii) and a classical convergence result (proven, e.g., in [15, Theorem 2.5.1] and [30, section 3.2]). □

4.2. Convergence of trust-region methods. Most trust-region methods compute the trust-region step such that the model decrease is at least a fraction of that obtained at the so-called Cauchy point. This condition alone is not sufficient to guarantee that the primary descent condition (3.1) holds. However, we show in this section that the strong descent conditions (Definition 3.1) hold under a mild modification of the Cauchy decrease condition.

Before stating the results in Theorem 4.4, we briefly review the underlying principles of trust-region methods. The vast majority of trust-region methods proceed along the following lines. At each iterate x_k, a *model* $m_k(p)$ is built that agrees with $\phi(x_k + p)$ to the first order, that is,

$$m_k(p) = \phi(x_k) + \nabla\phi(x_k)^T p + \frac{1}{2} p^T B_k p, \tag{4.6}$$

where B_k is some symmetric matrix. Then the problem

$$\min_{p \in \mathbb{R}^n} m_k(p) \quad \text{s.t.} \quad \|p\| \le \Delta_k, \tag{4.7}$$

where $\Delta_k > 0$ is the trust-region radius, is solved within some approximation, yielding an update vector p_k. Finally the actual decrease of ϕ is compared with the decrease predicted by m_k in the ratio

$$\rho_k = \frac{\phi(x_k) - \phi(x_k + p_k)}{m_k(0) - m_k(p_k)}. \tag{4.8}$$

If ρ is exceedingly small, then the model is very bad: the step must be rejected and the trust-region radius must be reduced. If ρ is small but less dramatically so, then the step is accepted but the trust-region radius is reduced. If ρ is close to 1, then there is a good agreement between the model and the function over the step, and the trust region can be expanded. This can be formalized into the following algorithm (similar formulations are given, e.g., in [29, 7]).

Algorithm 4.2 (trust region; see, e.g., [30]). *Given $\bar{\Delta} > 0$, $\Delta_0 \in (0, \bar{\Delta})$, and $\eta \in (0, \frac{1}{4})$:*

```
for $k = 0, 1, 2, \ldots$
  Obtain $p_k$, $\|p_k\| < \Delta_k$, by (approximately) solving (4.7);
  evaluate $\rho_k$ from (4.8);
  if $\rho_k < \frac{1}{4}$
    $\Delta_{k+1} = \frac{1}{4}\|p_k\|$
  else if $\rho_k > \frac{3}{4}$ and $\|p_k\| = \Delta_k$
    $\Delta_{k+1} = \min(2\Delta_k, \bar{\Delta})$
  else
    $\Delta_{k+1} = \Delta_k$;
  if $\rho_k > \eta$
    $x_{k+1} = x_k + p_k$
  else
    $x_{k+1} = x_k$;
end (for).
```

Trust-region methods essentially differ in the way they approximately solve the trust-region subproblem (4.7). Most of the algorithms compute a step such that the model decrease is at least a fraction of that obtained at the Cauchy point. By definition, the Cauchy point is the solution p_k^C of the one-dimensional problem

$$p_k^C = \arg\min\{m_k(p) : p = \alpha\nabla\phi(x_k),\ \|p\| \le \Delta_k\}. \tag{4.9}$$

The class of methods that ensure a fraction of the Cauchy decrease includes the dogleg method of Powell [32], the double-dogleg method of Dennis and Mei [11], the truncated conjugate-gradient method of Steihaug [33] and Toint [34], and the two-dimensional subspace minimization strategy of Byrd, Schnabel, and Shultz [5]. These methods have weak convergence properties ($\|\nabla\phi(x_k)\| \to 0$) in general; see, e.g., [30, Theorem 4.8]. Other methods, including the one of Moré and Sorensen [29], do even better as they attempt to find a nearly exact solution of the trust-region subproblem (4.7). In this case a strong limit-point convergence result is available [29, Theorem 4.13] under some additional hypotheses, including nonsingularity of the Hessian of ϕ at an accumulation point.

Assuming that the cost function ϕ is analytic, we have to check that the strong descent conditions (Definition 3.1) hold in order to apply our main result (Theorem 3.2) and conclude to single limit-point convergence.

The following technical lemma will prove to be useful.

LEMMA 4.3. *If p_k^C is the Cauchy point defined in* (4.9), *then*

$$m_k(0) - m_k(p_k^C) \geq \frac{1}{2}\|\nabla\phi(x_k)\|\|p_k^C\|.$$

Proof. The Cauchy point p_k^C is given explicitly by (see, e.g., [30, eq. (4.8)])

$$p_k^C = -\tau_k \frac{\Delta_k}{\|\nabla\phi(x_k)\|}\nabla\phi(x_k), \tag{4.10a}$$

where

$$\tau_k = \begin{cases} 1 & \text{if } \nabla\phi(x_k)^T B_k \nabla\phi(x_k) \leq 0; \\ \min\left(\frac{\|\nabla\phi(x_k)\|^3}{\Delta_k \nabla\phi(x_k)^T B_k \nabla\phi(x_k)}, 1\right) & \text{otherwise.} \end{cases} \tag{4.10b}$$

We have

$$m_k(0) - m_k(p_k^C) - \frac{1}{2}\|\nabla\phi(x_k)\|\,\|p_k^C\| = \beta_k\left(1 - \frac{\tau_k\Delta_k}{\|\nabla\phi(x_k)\|^3}\nabla\phi(x_k)^T B_k \nabla\phi(x_k)\right)$$

with $\beta_k := \frac{1}{2}\tau_k\Delta_k\|\nabla\phi(x_k)\|$; thus the claim is equivalent to

$$1 - \frac{\tau_k\Delta_k}{\|\nabla\phi(x_k)\|^3}\nabla\phi(x_k)^T B_k \nabla\phi(x_k) \geq 0,$$

which follows from the definition of τ_k. □

Due to the variety of trust-region methods and the flexibility in the choice of the update direction, it is not possible to prove a generic convergence result of the nature of Theorem 4.1. Instead, Theorem 4.4 provides several easily verified conditions for the iterates of Algorithm 4.2 in order that its iterates satisfy the strong descent conditions (Definition 3.1). Once this is verified then the results of Theorem 3.2 apply. Convergence to a critical point again depends on additional weak convergence results for the algorithm considered.

The conditions given in Theorem 4.4 are progressively more restrictive on the iterates of Algorithm 4.2. Condition (B) imposes condition (3.1) on the model m_k. We show that this in turn implies condition (3.1) on the cost function ϕ. Condition (C) imposes a fraction of the Cauchy decrease that becomes more restrictive as the ratio $\|p_k\|/\|p_k^C\|$ grows. Condition (D) simply states that the model decrease is at least a fraction of that obtained at the Cauchy point. This condition holds for most of the standard trust-region algorithms. However, (D) alone is not sufficient to guarantee single limit-point convergence for analytic ϕ. To see this, consider, for example, the function in $\mathbb{R}^3$ given by

$$f(x) = \left(\sqrt{x_1^2 + x_2^2} - 1\right)^2 + x_3^2$$

which has a symmetry of revolution around the third axis. If B_k is chosen to be singular along the θ direction, then a sequence $\{x_k\}$ can be constructed that satisfies (D) but nevertheless loops endlessly toward the set $\{x : x_1^2 + x_2^2 = 1,\ x_3 = 0\}$. Condition (D) becomes sufficient with complementary conditions, like (E) which imposes a bound on $\|p_k\|/\|p_k^C\|$ or like (F) which imposes that B_k remains positive definite and does not become ill-conditioned.

Theorem 4.4. *Let $\{x_k\}$, $\{\Delta_k\}$, $\{p_k\}$, $\{\phi(x_k)\}$, $\{\nabla\phi(x_k)\}$, and $\{B_k\}$ be infinite sequences generated by Algorithm* 4.2 *(trust region). Let m_k, ρ_k, and p_k^C be defined as in* (4.6)*,* (4.8)*, and* (4.9)*, respectively. Consider the following conditions.*

(A) *The strong descent conditions (Definition* 3.1*) hold.*

(B) *There exists $\sigma_1 > 0$ such that for all k with $\nabla\phi(x_k) \neq 0$,*

$$m_k(0) - m_k(p_k) \geq \sigma_1 \|\nabla\phi(x_k)\|\|p_k\|. \tag{4.11}$$

(C) *There exists $\sigma_2 > 0$ such that for all k with $\nabla\phi(x_k) \neq 0$,*

$$\frac{m_k(0) - m_k(p_k)}{m_k(0) - m_k(p_k^C)} \geq \sigma_2 \frac{\|p_k\|}{\|p_k^C\|}. \tag{4.12}$$

(D) *There exists $c_2 > 0$ such that for all k with $\nabla\phi(x_k) \neq 0$,*

$$m_k(0) - m_k(p_k) \geq c_2(m_k(0) - m_k(p_k^C)). \tag{4.13}$$

(E) *There exists $\kappa_1 > 0$ such that for all k with $\nabla\phi(x_k) \neq 0$,*

$$\|p_k\| \leq \kappa_1 \|p_k^C\|. \tag{4.14}$$

(F) *B_k is positive definite for all k and there exists a $\kappa_2 \geq 1$ such that* $\mathrm{cond}(B_k) := \|B_k\|\|B_k^{-1}\| \leq \kappa_2$ *for all k (where the matrix norms are 2-norms).*

Then (D) *and* (F) $\Rightarrow$ (D) *and* (E) $\Rightarrow$ (C) $\Rightarrow$ (B) $\Rightarrow$ (A). *Furthermore, if* (A) *holds and the cost function ϕ is analytic, then either $\lim_{k\to\infty} \|x_k\| = +\infty$ or there exists a single point $x^* \in \mathbb{R}^n$ such that $\lim_{k\to\infty} x_k = x^*$.*

Proof. First note that the condition $\nabla\phi(x_k) \neq 0$ guarantees that $p_k^C \neq 0$ and $m_k(0) - m_k(p_k^C) > 0$.

(D) and (F) $\Rightarrow$ (D) and (E). If $\|p_k^C\| = \Delta_k$, then (E) holds with $\kappa_1 = 1$. Assume then that $\|p_k^C\| < \Delta_k$. Let $\lambda_{\max}(B_k)$, respectively, $\lambda_{\min}(B_k)$, denote the largest, respectively, smallest, eigenvalue of the positive definite matrix B_k. Then

$$\frac{\|\nabla\phi(x_k)\|}{\lambda_{\max}(B_k)} \leq \frac{\|\nabla\phi(x_k)\|^3}{\nabla\phi(x_k)^T B_k \nabla\phi(x_k)} = \|p_k^C\|, \tag{4.15}$$

where the equality follows from (4.10) and $\|p_k^C\| < \Delta_k$. In view of (D), one has $m_k(0) - m_k(p_k) \geq 0$ and thus $-\nabla\phi(x_k)^T p_k - \frac{1}{2} p_k^T B_k p_k \geq 0$. Therefore

$$\frac{1}{2}\lambda_{\min}(B_k)\|p_k\|^2 \leq \frac{1}{2} p_k^T B_k p_k \leq -\nabla\phi(x_k)^T p_k \leq \|\nabla\phi(x_k)\|\,\|p_k\|. \tag{4.16}$$

It follows from (4.15) and (4.16) that

$$\|p_k\| \leq 2\frac{\|\nabla\phi(x_k)\|}{\lambda_{\min}(B_k)} \leq 2\frac{\lambda_{\max}(B_k)}{\lambda_{\min}(B_k)}\|p_k^C\| = 2\mathrm{cond}(B_k)\|p_k^C\| \leq 2\kappa_2\|p_k^C\|;$$

i.e., (E) holds with $\kappa_1 := 2\kappa_2$.

(D) and (E) $\Rightarrow$ (C). The proof is direct, with $\sigma_2 = c_2/\kappa_1$.

(C) $\Rightarrow$ (B). The proof directly follows from Lemma 4.3, with $\sigma_1 = \sigma_2/2$.

(B) $\Rightarrow$ (A). If $x_{k+1} = x_k$, then the strong descent conditions trivially hold. Assume then that $x_{k+1} \neq x_k$, in which case the complementary descent condition (3.2) holds by the definition of Algorithm 4.2. If $\nabla\phi(x_k) = 0$, then the primary descent condition (3.1) trivially holds. On the other hand, if $\nabla\phi(x_k) \neq 0$, then it follows from (B) that (3.1) holds with $\sigma = \eta\sigma_1$, where η is defined in Algorithm 4.2.

The final claim follows directly from Theorem 3.2. □

Convergence of the iterates of Algorithm 4.2 to a critical point depends on additional weak convergence ($\|\nabla\phi(x_k)\| \to 0$) results for the particular algorithm considered. For example, if assumptions (D) and (E) hold, ϕ is analytic, and $\|B_k\| \leq \beta$ for some constant β, then either $\lim_{k\to\infty}\|x_k\| = +\infty$ or there exists a single point $x^* \in \mathbb{R}^n$ such that

$$\lim_{k\to\infty} x_k = x^* \text{ and } \nabla\phi(x^*) = 0.$$

This follows from the above result along with a classical convergence result for trust-region methods (see [30, Theorem 4.8]).

5. Conclusion.

We have shown strong limit-point convergence results that do not rely on the usual requirement that critical points are isolated. Instead, we require two conditions: the Łojasiewicz gradient inequality (2.1), i.e., a lower bound on the norm of the gradient of the cost function in terms of the cost function itself, and some "strong descent conditions" stated in Definition 3.1. The Łojasiewicz gradient inequality is satisfied in particular for analytic cost functions. The strong descent conditions are satisfied for a wide variety of optimization schemes; they include line-search methods with an angle condition on the search direction and Armijo's condition on the step length, and trust-region methods under the condition that the length of the update vector is bounded by a multiple of the length of the Cauchy update.

Acknowledgments. The authors would like to thank all reviewers who have provided comments on this paper during its various revisions. They would particularly like to thank both J.-C. Gilbert and E. W. Sachs for handling the paper as associate editors and for providing excellent suggestions and recommendations to improve the material. In addition, the authors wish to thank K. Kurdyka and A. L. Tits for useful discussions related to this topic.

REFERENCES

[1] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995.

[2] E. Bierstone and P. D. Milman, *Semianalytic and subanalytic sets*, Inst. Hautes Études Sci. Publ. Math., 67 (1988), pp. 5–42.

[3] J. Bolte, A. Daniilidis, and A. S. Lewis, *The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems generalized eigenvalue problems*, submitted.

[4] R. H. Byrd and J. Nocedal, *A tool for the analysis of quasi-Newton methods with application to unconstrained minimization*, SIAM J. Numer. Anal., 26 (1989), pp. 727–739.

[5] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, *Approximate solution of the trust region problem by minimization over two-dimensional subspaces*, Math. Programming, 40 (1988), pp. 247–263.

[6] A. R. Conn, N. Gould, A. Sartenaer, and Ph. L. Toint, *Global convergence of a class of trust region algorithms for optimization using inexact projections on convex constraints*, SIAM J. Optim., 3 (1993), pp. 164–221.

[7] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust-Region Methods*, MPS/SIAM Ser. Optim. 1, SIAM, Philadelphia, 2000.

[8] H. B. Curry, *The method of steepest descent for non-linear minimization problems*, Quart. Appl. Math., 2 (1944), pp. 258–261.

[9] J.-P. Dedieu, *Approximate solutions of analytic inequality systems*, SIAM J. Optim., 11 (2000), pp. 411–425.

[10] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall Series in Computational Mathematics, Prentice Hall, Englewood Cliffs, NJ, 1983.

[11] J. E. Dennis, Jr. and H. H. W. Mei, *Two new unconstrained optimization algorithms which use function and gradient values*, J. Optim. Theory Appl., 28 (1979), pp. 453–482.

[12] J. C. Dunn, *Global and asymptotic convergence rate estimates for a class of projected gradient processes*, SIAM J. Control Optim., 19 (1981), pp. 368–400.

[13] J. C. Dunn, *On the convergence of projected gradient processes to singular critical points*, J. Optim. Theory Appl., 55 (1987), pp. 203–216.

[14] F. Facchinei, A. Fischer, and C. Kanzow, *On the identification of zero variables in an interior-point framework*, SIAM J. Optim., 10 (2000), pp. 1058–1078.

[15] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., John Wiley & Sons, Chichester, UK, 1987.

[16] A. A. Goldstein, *On steepest descent*, J. Soc. Indust. Appl. Math. Ser. A Control, 3 (1965), pp. 147–151.

[17] U. Helmke and J. B. Moore, *Optimization and Dynamical Systems*, Springer, London, 1994.

[18] K. C. Kiwiel and K. Murty, *Convergence of the steepest descent method for minimizing quasiconvex functions*, J. Optim. Theory Appl., 89 (1996), pp. 221–226.

[19] K. Kurdyka, *On gradients of functions definable in o-minimal structures*, Ann. Inst. Fourier (Grenoble), 48 (1998), pp. 769–783.

[20] K. Kurdyka, T. Mostowski, and A. Parusiński, *Proof of the gradient conjecture of R. Thom*, Ann. of Math. (2), 152 (2000), pp. 763–792.

[21] K. Kurdyka and A. Parusiński, *$\mathbf{w}_f$-stratification of subanalytic functions and the Łojasiewicz inequality*, C. R. Acad. Sci. Paris Sér. I Math., 318 (1994), pp. 129–133.

[22] C. Lageman, *Konvergenz reell-analytischer gradientenähnlicher Systeme*, Diplomarbeit im Fach Mathematik, Mathematisches Institut, Universität Würzburg, Wüzburg, Germany, 2002.

[23] S. Łojasiewicz, *Sur le problème de la division*, Studia Math., 18 (1959), pp. 87–136.

[24] S. Łojasiewicz, *Ensembles semi-analytiques*, Inst. Hautes Études Sci., Bures-sur-Yvette, France, 1965.

[25] S. Łojasiewicz, *Sur les trajectoires du gradient d'une fonction analytique*, in Seminari di Geometria 1982-1983, Istituto di Geometria, Dipartimento di Matematica, Università di Bologna, Bologna, Italy, 1984, pp. 115–117.

[26] S. Łojasiewicz, *Sur la géométrie semi- et sous-analytique*, Ann. Inst. Fourier (Grenoble), 43 (1993), pp. 1575–1595.

[27] Z.-Q. Luo and J.-S. Pang, *Error bounds for analytic systems and their applications*, Math. Programming, 67 (1994), pp. 1–28.

[28] R. E. Mahony, *Convergence of gradient flows and gradient descent algorithms for analytic cost functions*, in Proceedings of the Thirteenth International Symposium on the Mathematical Theory of Networks and Systems (MTNS98), Padova, Italy, 1998, pp. 653–656.

[29] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[30] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Ser. Oper. Res., Springer-Verlag, New York, 1999.

[31] J. Palis, Jr. and W. de Melo, *Geometric Theory of Dynamical Systems*, Springer-Verlag, New York, 1982.

[32] M. J. D. Powell, *A new algorithm for unconstrained optimization*, in Nonlinear Programming (Proc. Sympos., Univ. of Wisconsin, Madison, Wis., 1970), Academic Press, New York, 1970, pp. 31–65.

[33] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[34] Ph. L. Toint, *Towards an efficient sparsity exploiting Newton method for minimization*, in Sparse Matrices and Their Uses, I. S. Duff, ed., Academic Press, London, 1981, pp. 57–88.

[35] S. Wiggins, *Introduction to Applied Nonlinear Dynamical Systems and Chaos*, in Texts in Applied Mathematics 2, Springer-Verlag, New York, 1990.

[36] P. Wolfe, *Convergence conditions for ascent methods*, SIAM Rev., 11 (1969), pp. 226–235.

[37] N. Yamashita, H. Dan, and M. Fukushima, *On the identification of degenerate indices in the nonlinear complementarity problem with the proximal point algorithm*, Math. Program., 99 (2004), pp. 377–397.

[38] G. Zoutendijk, *Mathematical Programming Methods*, North–Holland, Amsterdam, 1976.

SIAM J. Optim.
Vol. 16, No. 2, pp. 548–562

A NOTE ON TRUST-REGION RADIUS UPDATE[*]

JÉRÔME M. B. WALMAG[†] AND ÉRIC J. M. DELHEZ[‡]

Abstract. In classical trust-region optimization algorithms, the radius of the trust region is reduced, kept constant, or enlarged after, respectively, unsuccessful, successful, and very successful iterations. We propose here to refine the empirical rules used for this update by the definition of a new set of iterations that we call "too successful iterations." At such iterations, a large reduction of the objective function is obtained despite a crude local approximation of the objective function; the trust region is thus kept nearly constant instead of being enlarged.

The new update rules preserve the strong convergence property of traditional trust-region methods. They can also be generalized to define a self-adaptive trust-region algorithm along the lines introduced by Hei [*J. Comput. Math.*, 21 (2003), pp. 229–236].

Numerical experiments carried out on 70 unconstrained problems from the CUTEr collection demonstrate the positive impact of the modified update strategy on the efficiency and robustness of quasi-Newton variants of a trust-region solver, when BFGS or SR1 updates of the approximation of the Hessian matrix are carried at all iterations.

Key words. unconstrained optimization, nonlinear optimization, trust region, radius update

AMS subject classifications. 90C26, 90C53, 65K05

DOI. 10.1137/030602563

1. Introduction.

Trust-region methods are increasingly used in applied mathematics and engineering to tackle optimization problems. They indeed provide an efficient alternative to the usual line-search methods that demand repeated and costly evaluations of the objective function. A very general framework for trust-region methods can be found in Conn, Gould, and Toint [3].

In this paper, we consider a trust-region method applied to the unconstrained optimization problem

$$\text{find} \quad x^* = \arg\min_{x \in \mathbb{R}^n} f(x), \tag{1.1}$$

where $f(x)$ is a real-valued twice-continuously differentiable function. This formulation is typical of many parameter identification problems and is therefore of very general use. The ideas developed here are, however, directly applicable to constrained optimization problems as well.

Trust-region methods are iterative methods whose key ideas are as follows. At each iteration, an analytical local model $m^{(k)}(x^{(k)} + s)$ of the true objective function is built around the current iterate $x^{(k)}$. A trial point

$$\tilde{x}^{(k+1)} = x^{(k)} + s^{(k+1)} \tag{1.2}$$

is then generated using a solution $s^{(k+1)}$ of the subproblem

$$\text{find} \quad s^{(k+1)} = \arg\min_{s \in \mathcal{B}^{(k)}} m^{(k)}(x^{(k)} + s), \tag{1.3}$$

[*] Received by the editors December 19, 2003; accepted for publication (in revised form) May 20, 2005; published electronically December 30, 2005.

http://www.siam.org/journals/siopt/16-2/60256.html

[†] Corresponding author. Université de Liège, Modélisation et Méthodes Mathématiques, Sart-Tilman B37, B-4000 Liège, Belgium (J.Walmag@ulg.ac.be).

[‡] Université de Liège, Modélisation et Méthodes Mathématiques, Sart-Tilman B37, B-4000 Liège, Belgium (e.delhez@ulg.ac.be).

where $\mathcal{B}^{(k)}$ is defined as

$$\mathcal{B}^{(k)} = \left\{ s \in \mathbb{R}^n, \|s\| \leq \Delta^{(k)} \right\} \tag{1.4}$$

with $\|\cdot\|$ being the ℓ_2-norm and $\Delta^{(k)} > 0$.

The model function $m^{(k)}$ is deemed to provide a valid approximation of the objective function f in $\mathcal{B}^{(k)}$; hence the name of trust region for $\mathcal{B}^{(k)}$ and the name of trust-region radius for $\Delta^{(k)}$. As a model function $m^{(k)}(x)$ is often available in closed form, appropriate techniques can be applied to find an approximate minimizer (any step giving a decrease of the model function that is lower than or equal to a fraction of the reduction obtained at the so-called Cauchy point is appropriate; see Conn, Gould, and Toint [3] for more details) of subproblem (1.3). Adequacy of the predicted reduction and true variation of the objective function is measured by means of the ratio

$$\rho^{(k)} = \frac{f(x^{(k)}) - f(\tilde{x}^{(k+1)})}{m^{(k)}(x^{(k)}) - m^{(k)}(\tilde{x}^{(k+1)})}, \tag{1.5}$$

where $\tilde{x}^{(k+1)}$ is the trial point. This trial point is accepted as new iterate $x^{(k+1)}$ if a sufficient reduction of the true objective function is achieved, i.e., if $\rho^{(k)} \geq \eta_1$, where η_1 is a predefined positive threshold. The iteration is then said to be *successful*. If not, the iteration is *unsuccessful*, the trial point is rejected, and $x^{(k+1)} = x^{(k)}$.

The ratio $\rho^{(k)}$ defined by (1.5) provides a measure of the fidelity of the model $m^{(k)}$ to the true objective function f in the neighborhood of the current iterate. It is then used to update the radius $\Delta^{(k)}$ of the trust region from one iteration to the other. The usual empirical rules for this update can be summarized as follows (see, e.g., Gould et al. [7]):

$$\Delta^{(k+1)} = \begin{cases} \alpha_1 \Delta^{(k)} & \text{if } \rho^{(k)} < \eta_1, \\ \Delta^{(k)} & \text{if } \eta_1 \leq \rho^{(k)} < \eta_2, \\ \alpha_2 \Delta^{(k)} & \text{if } \rho^{(k)} \geq \eta_2, \end{cases} \tag{1.6}$$

where α_1, α_2, η_1, and η_2 are predefined constants such that

$$0 < \eta_1 \leq \eta_2 < 1 \quad \text{and} \quad \alpha_1 < 1 < \alpha_2. \tag{1.7}$$

In other words, the radius of the trust region is reduced after unsuccessful iterations and kept constant or increased after successful iterations.

The update strategy defined by (1.6) is likely to have a strong influence on the performance of the algorithm. On the one hand, if the radius of the trust region is too small, the successive iterates will remain close to each other, and the algorithm will converge slowly. On the other hand, if the trust region is too large, the algorithm will perform a large number of successive unsuccessful iterations. The update strategy is therefore critical for the efficiency of the algorithm but has received only little attention so far. Various values for parameters (1.7) are used by different authors (e.g., Dennis and Mei [4], Gould et al. [8]), but the general formula (1.6) is seldom questioned. Hei [10] generalizes (1.6) to allow for a more continuous dependency of the trust-region radius on $\rho^{(k)}$. Byrd, Khalfan, and Schnabel [2] suggest further refinement when $\rho^{(k)} < 0$, i.e., when the radius is too large or the approximation of the objective function by the model function is so bad that some drastic action should be taken. See section 4 for further refinements.

In this paper, we introduce a refinement of (1.6) that is applicable when $\rho^{(k)}$ is much larger than unity and show, on a subset of problems from the CUTEr set (see

Gould, Orban, and Toint [9]), that this modification is beneficial to the performance of the algorithm. The generalization of this new update strategy is also developed along the lines defined by Hei [10].

2. The too successful iterations.

Iterations of the third category defined in (1.6) are called *very successful* because they produce reductions of the true objective function that are similar to or larger than the reductions predicted by the model $m^{(k)}$. As suggested by (1.6), the usual approach in such a case is to enlarge the trust region. The rationale for this increase of $\Delta^{(k)}$ is that we are confident that the model is accurate in a large region about the current iterate and the algorithm should therefore be allowed to take bigger steps if required.

This is, however, only part of the story. While very successful iterations deserve their name because of the decrease of the objective function that they produce, they may rely on inaccurate local models $m^{(k)}$ if $\rho^{(k)}$ is significantly larger than unity. In this case, the decrease of the objective function appears rather fortunate, and there is no reason to be overconfident in the model $m^{(k)}$. This suggests the definition of the set of *too successful iterations* characterized by $\rho^{(k)} > \eta_3$, where $\eta_3 > 1$ is a predetermined constant, and the replacement of (1.6) by

$$\Delta^{(k+1)} = \begin{cases} \alpha_1\,\Delta^{(k)} & \text{if } \rho^{(k)} < \eta_1, \\ \Delta^{(k)} & \text{if } \eta_1 \le \rho^{(k)} < \eta_2, \\ \alpha_2\,\Delta^{(k)} & \text{if } \eta_2 \le \rho^{(k)} \le \eta_3, \\ \alpha_3\,\Delta^{(k)} & \text{if } \rho^{(k)} > \eta_3, \end{cases} \tag{2.1}$$

where

$$0 < \eta_1 \le \eta_2 < 1 < \eta_3 \tag{2.2}$$

and

$$\alpha_1 < 1 < \alpha_3 < \alpha_2. \tag{2.3}$$

The usual update rules (1.6) appear as a particular case of the new rules (2.1), where $\eta_3 = +\infty$.

According to (2.1), the maximum increase of the radius of the trust region occurs when $\rho^{(k)}$ is close to one, i.e., when the model function $m^{(k)}$ provides an accurate local approximation of the objective function. At too successful iterations, the reduction of the objective function obtained at iteration k is significantly larger than the reduction expected from $m^{(k)}$. While this iteration allows the algorithm to progress towards the optimum, there is no reason to believe that the next step will be as fortunate as the current one since $m^{(k+1)}$ is likely to be as inaccurate as $m^{(k)}$. It therefore seems safer to avoid increasing the size of the trust region too rapidly, and we take $\alpha_3 < \alpha_2$.

One could conclude that the trust region must shrink after too successful iterations. We take, however, $\alpha_3 > 1$—but close to unity—to match the convergence criteria presented by Conn, Gould, and Toint [3] for the general trust-region algorithm. Indeed, these authors proved global convergence, at least to a first-order critical point, for the general update strategy defined by the constants η_1, η_2, γ_1, γ_2, γ_3, and γ_4 such that

$$0 < \eta_1 \le \eta_2 < 1 \quad \text{and} \quad 0 < \gamma_1 \le \gamma_2 < 1 < \gamma_3 \le \gamma_4 \tag{2.4}$$

and the update rules

$$\Delta^{(k+1)} \in \begin{cases} [\ \gamma_1\Delta^{(k)}\ ,\ \gamma_2\Delta^{(k)}\ ] & \text{if } \rho^{(k)} < \eta_1, \\ [\ \gamma_2\Delta^{(k)}\ ,\ \Delta^{(k)}\ ] & \text{if } \rho^{(k)} \in [\eta_1\ ,\ \eta_2[, \\ [\ \gamma_3\Delta^{(k)}\ ,\ \gamma_4\Delta^{(k)}\ ] & \text{if } \rho^{(k)} \geq \eta_2. \end{cases} \tag{2.5}$$

Obviously, the modified update rules (2.1) satisfy the convergence criteria given by Conn, Gould, and Toint [3], and the strong general convergence properties of trust-region methods are therefore retained (at least if the model and objective function share the same value and gradient at the current iterate and if the other convergence conditions discussed in Chapter 6 of [3] are met).

3. Self-adaptive trust-region algorithm. The concept of too successful iterations can also be used in the context of the self-adaptive trust-region method introduced by Hei [10]. The idea presented in Hei [10] is simply to allow the updated trust-region radius $\Delta^{(k+1)}$ to vary more or less continuously with the ratio $\rho^{(k)}$ according to

$$\Delta^{(k+1)} = R(\rho^{(k)})\ \Delta^{(k)}, \tag{3.1}$$

where R is some appropriate function such that the convergence conditions (2.5) are satisfied. Obviously, the usual update rules (1.6) are a particular case of (3.1) corresponding to a staircase function R_1 (Figure 3.1, top left). Hei [10] suggests using nondecreasing R-functions such as

$$R_2(\rho^{(k)}) = \begin{cases} \alpha_1 & \text{if } \rho^{(k)} \leq 0, \\ \alpha_1 + (1-\alpha_1)\left(\frac{\rho^{(k)}}{\eta_2}\right)^2 & \text{if } 0 < \rho^{(k)} < \eta_2, \\ \alpha_3 + (\alpha_2 - \alpha_3)\exp\left\{-\left(\frac{\rho^{(k)}-1}{\eta_2-1}\right)^2\right\} & \text{if } \eta_2 \leq \rho^{(k)} < 1, \\ 2\alpha_2 - \alpha_2\exp(1-\rho^{(k)}) & \text{if } \rho^{(k)} \geq 1, \end{cases} \tag{3.2}$$

where $\alpha_1 < 1 < \alpha_3 < \alpha_2$ and $\eta_2 < 1$ are appropriate constants (Figure 3.1, bottom left). This R_2 function is qualitatively similar to the original update rule R_1 since it allows the trust region to grow after too successful iterations.

To generalize the modified update rules (2.1), we define Λ-functions as one-dimensional functions $\Lambda(t)$ defined in $\mathbb{R}$ such that

1. $\Lambda(t)$ is nondecreasing in $]-\infty, 1]$ and nonincreasing in $[1, +\infty[$,
2. $\lim_{t\to-\infty}\Lambda(t) = \alpha_1$,
3. $\lim_{t\to\eta_2^-}\Lambda(t) = 1$,
4. $\Lambda(\eta_2) > \alpha_3 > 1$,
5. $\Lambda(1) = \alpha_2$,
6. $\lim_{t\to+\infty}\Lambda(t) = \alpha_3$,

where the constants α_1, α_2, α_3 satisfy condition (2.3) and η_2 is the usual threshold used for the definition of very successful iterations.

The update rule

$$\Delta^{(k+1)} = \Lambda(\rho^{(k)})\ \Delta^{(k)} \tag{3.3}$$

is then a special case of (2.5) with $\gamma_1 = \alpha_1$, $\gamma_3 = \alpha_3$, $\gamma_4 = \alpha_2$, and $\gamma_2 = \Lambda(\eta_1)$, so that the convergence properties hold. The modified update rules (2.1) can be described by

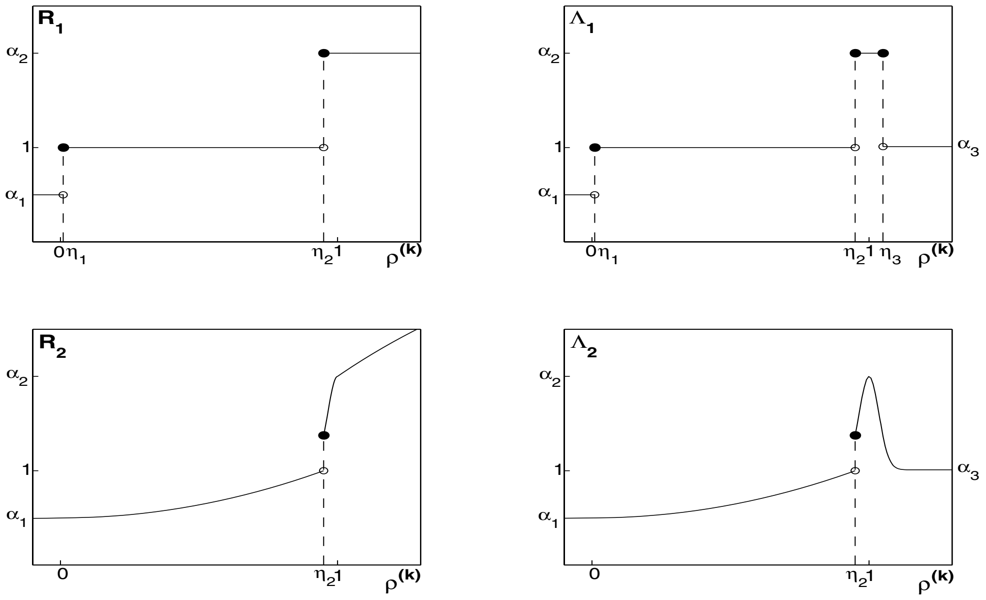


FIG. 3.1. *Self-adaptive functions used in the numerical experiments.*

a staircase function Λ_1 (Figure 3.1, top right) of the form (3.3). As a generalization of (3.2), we propose to use the Λ-function defined by

$$\Lambda_2(\rho^{(k)}) = \begin{cases} \alpha_1 & \text{if } \rho^{(k)} \leq 0, \\ \alpha_1 + (1-\alpha_1)\left(\frac{\rho^{(k)}}{\eta_2}\right)^2 & \text{if } 0 < \rho^{(k)} < \eta_2, \\ \alpha_3 + (\alpha_2 - \alpha_3)\exp\left\{-\left(\frac{\rho^{(k)}-1}{\eta_2-1}\right)^2\right\} & \text{if } \rho^{(k)} \geq \eta_2. \end{cases} \tag{3.4}$$

This Λ_2-function is qualitatively similar to the update rule Λ_1 since it allows the trust region to grow only if $\rho^{(k)}$ is about unity (Figure 3.1, bottom right).

4. Refinements. Some refinements for trust-region radius update have been introduced in the literature (see in particular Conn, Gould, and Toint [3]). They usually appear as empirical "tricks" introduced to improve efficiency. None of these tricks appears similar to the too successful iterations defined in this paper.

The first convenient rule is quite natural as soon as numerical experiments are made: a user-defined maximum trust-region radius $\Delta_{\max}$ is simply introduced to prevent too large trust regions. In some cases, it is also used to prove convergence to second-order critical points (e.g., [3]).

Another refinement is to base the trust-region radius update on the step length. For example, the following quite cumbersome rule has been proposed (Conn, Gould, and Toint [3]):

$$\Delta^{(k+1)} = \begin{cases} \max(\alpha_2 \|s^{(k)}\|, \Delta^{(k)}) & \text{if } \rho^{(k)} \geq \eta_2, \\ \Delta^{(k)} & \text{if } \rho^{(k)} \in [\eta_1, \eta_2[, \\ \alpha_1 \|s^{(k)}\| & \text{if } \rho^{(k)} \in [0, \eta_1[, \\ \min[\alpha_1 \|s^{(k)}\|, \max(\gamma_1, \gamma_{\text{bad}}^{(k)})\,\Delta^{(k)}] & \text{if } \rho^{(k)} < 0, \end{cases} \tag{4.1}$$

where γ_1 is a given constant and

$$\gamma_{\text{bad}}^{(k)} = \frac{(1-\eta_2)\, s^{(k)T} \nabla_x f(x^{(k)})}{(1-\eta_2)[f(x^{(k)}) + s^{(k)T} \nabla_x f(x^{(k)})] + \eta_2 m^{(k)}(\tilde{x}^{(k)}) - f(\tilde{x}^{(k)})}. \tag{4.2}$$

These rules may still be applied in the same way even if we introduce too successful iterations. However, for the sake of simplicity, they are not used in the following numerical experiments.

5. Numerical experiments.

The ideas introduced in sections 2 and 3 are first illustrated on a variant of the classical Rosenbrock banana function. The influence of the modified update rules on the robustness and efficiency of a trust-region solver is then assessed on 70 small-scale problems of the CUTEr test set.

5.1. Particular implementation of the trust-region algorithm.

The particular version of the trust-region algorithm used in this study is based on quadratic models

$$m^{(k)}(x^{(k)} + s) = f(x^{(k)}) + s^T \nabla_x f(x^{(k)}) + \frac{1}{2}\, s^T H^{(k)} s, \tag{5.1}$$

where $f(x^{(k)})$ and $\nabla_x f(x^{(k)})$ are readily available at each iteration while $H^{(k)}$ approximates the Hessian matrix. Two strategies are used for the approximation of the Hessian matrix: the symmetric rank one (SR1) update formula and the Broyden–Fletcher–Goldfarb–Shanno (BFGS) formula (see, e.g., Fletcher [6]).

In a first step, these quasi-Newton update formulas are used after every iteration (successful or not). Another quasi-Newton strategy is discussed in section 5.4. The versions of the algorithm using the SR1 and BFGS approaches are referred to as, respectively, Trust-SR1 and Trust-BFGS. Global convergence of the Trust-BFGS algorithm towards first-order critical points can be proved, while Trust-SR1 converges towards second-order critical points (Conn, Gould, and Toint [3]). The quadratic subproblems (1.3) are solved using the GQT routines of Moré and Sorensen [11]. The values of the parameters of the different update rules are listed in Table 5.1.

TABLE 5.1
Parameters of the update rule used in the numerical experiments.

Parameter	Value	Parameter	Value
α_1	0.5	η_1	0.01
α_2	2	η_2	0.95
α_3	1.01	η_3	1.05

5.2. Rosenbrock's banana function.

A first test of the ideas presented in the previous sections is provided by the minimization of a logarithmic variant of the well-known Rosenbrock function (see Fletcher [6])

$$f(x_1, x_2) = \ln\left[1 + 10000(x_2 - x_1^2)^2 + (1 - x_1)^2\right]. \tag{5.2}$$

This function has a deep, curved valley following the parabola $x_2 = x_1^2$, and its minimizer is $(x_1, x_2) = (1, 1)$. Figure 5.1 shows the evolution of the objective function with both algorithms Trust-SR1 and Trust-BFGS associated with the four update rules R_1, R_2, Λ_1, and Λ_2, while the numbers of iterations required to reach convergence are listed in Table 5.2. For completeness, the behavior of a Newton version of the algorithm using the exact Hessian matrix is also shown.

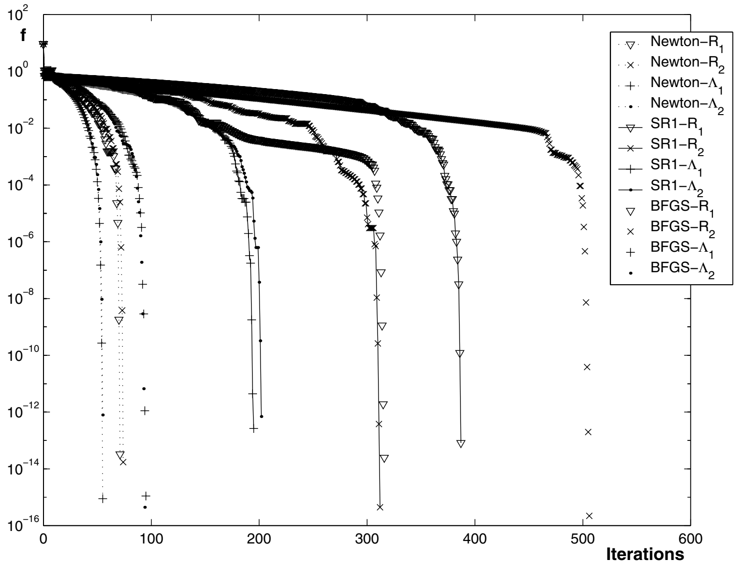


FIG. 5.1. *The value of the objective function on successive iterations. The starting point is* $(x_1, x_2) = (1, 0)$ *and the initial trust-region radius is* $\Delta^{(0)} = 1$. *The stopping criterion is* $\|\nabla_x f(x^{(k)})\| \leq 5 \times 10^{-6}$.

TABLE 5.2
Number of iterations for the logarithmic Rosenbrock problem. In parenthesis: number of successful iterations.

	Trust-SR1		Trust-BFGS		Trust-Newton	
R_1	388	(211)	317	(176)	72	(46)
R_2	312	(153)	506	(260)	74	(46)
Λ_1	195	(152)	96	(87)	55	(48)
Λ_2	203	(164)	94	(86)	56	(49)

With all the Newton, SR1, and BFGS versions of the algorithm, the Λ-functions appear much more efficient than the R-functions. The suggested algorithmic modifications decrease the number of iterations and then the number of evaluations of the objective function at no additional cost. It can also be seen that the proportion of successful iterations is greater with Λ-functions than with R-functions.

The lower number of iterations obtained with Λ-functions results from the combination of two effects. The first is a reduction of the number of iterations induced by the conservative update rule; this prevents the algorithm from wasting time with too large steps that produce unsuccessful iterations. The second effect is related to the SR1 and BFGS updates of the Hessian matrix; it is therefore irrelevant for the Newton version of the algorithm. The many unsuccessful iterations carried out with the R-function versions of the algorithm produce large steps $s^{(k)}$ and inaccurate quasi-Newton updates of the Hessian matrix. On the contrary, the Λ-function approach tends to give shorter trial steps $s^{(k)}$ when the model does not fit the objective function, and hence provides a more accurate quasi-Newton update.

5.3. Performances on a test set. A systematic comparison between the different trust-region radius updates is carried out for the 70 twice-continuously dif-

Table 5.3
Names and sizes of the selected CUTEr problems.

Name	n	Name	n	Name	n	Name	n
3PK	30	DENSCHNF	2	HYDC20LS	99	PFIT1LS	3
AKIVA	2	DJTL	2	JENSMP	2	PFIT2LS	3
ALLINITU	4	ENGVAL2	3	KOWOSB	4	PFIT3LS	3
BARD	3	EXPFIT	2	LOGHAIRY	2	PFIT4LS	3
BEALE	2	GROWTHLS	3	MARATOSB	2	ROSENBR	2
BIGGS6	6	GULF	3	MEXHAT	2	S308	2
BOX3	3	HAIRY	2	MEYER3	3	SINEVAL	2
BRKMCC	2	HATFLDD	3	OSBORNEA	5	SISSER	2
BROWNBS	2	HATFLDE	3	OSBORNEB	11	SNAIL	2
BROWNDEN	4	HEART6LS	6	PALMER1C	8	STRATEC	10
CLIFF	2	HEART8LS	8	PALMER1D	7	TOINTGOR	50
CUBE	2	HELIX	3	PALMER2C	8	TOINTPSP	50
DECONVU	61	HIELOW	3	PALMER3C	8	TOINTQOR	50
DENSCHNA	2	HIMMELBB	2	PALMER4C	8	VIBRBEAM	8
DENSCHNB	2	HIMMELBF	4	PALMER5C	6	YFITU	3
DENSCHNC	2	HIMMELBG	2	PALMER6C	8	ZANGWIL2	2
DENSCHND	3	HIMMELBH	2	PALMER7C	8		
DENSCHNE	3	HUMPS	2	PALMER8C	8		

ferentiable small-scale unconstrained problems ($n \leq 100$; see Table 5.3) with first derivatives available contained in the CUTEr test set (see Bongartz et al. [1] and Gould, Orban, and Toint [9]). Results are analyzed by means of performance profiles proposed by Dolan and Moré [5]. Separate profiles are computed for the SR1 and BFGS update rules.

For both the SR1 and BFGS update rules, we define the set $\mathcal{P}$ of $n_p\,(= 70)$ test problems and the set $\mathcal{S}$ of the four solvers implementing the different update rules for the radius of the trust region (R_1, R_2, Λ_1, Λ_2). For each problem $p \in \mathcal{P}$ and solver $s \in \mathcal{S}$, the number of iterations $N_{p,s}$ needed to solve problem p with solver s is evaluated. A *performance ratio*

$$r_{p,s} = \frac{N_{p,s}}{\min\{N_{p,s} : s \in \mathcal{S}\}}$$

is then built by comparing the number of iterations, i.e., the number of objective function evaluations, required by solver s to solve problem p with the best performance obtained by any solver on this problem. An arbitrarily large ($r_M = 100$) performance ratio is assigned to a solver s when it is unable to solve a given problem. The *performance profile* of a solver s is defined as the cumulative distribution function for the performance ratio

$$P_s(\tau) = \frac{1}{n_p}\,|\mathcal{J}_s(\tau)|\,, \tag{5.3}$$

where $\mathcal{J}_s(\tau) = \{p \in \mathcal{P} : r_{p,s} \leq \tau\}$. With such definitions, $P_s(1)$ appears as the probability that the solver s will win over the rest of the solvers and can therefore be used to compare the average speed of the algorithms. The limit

$$P_s^* = \lim_{\tau \to r_M^-} P_s(\tau)$$

is the probability for the solver s to solve a problem and can therefore be used to compare the robustness of the algorithms. These values are shown in Table 5.4.

TABLE 5.4
Speed and robustness of the different algorithms.

	Trust-SR1		Trust-BFGS	
	$P_s(1)$	P_s^*	$P_s(1)$	P_s^*
R_1	0.30	0.74	0.43	0.83
R_2	0.46	0.80	0.36	0.84
Λ_1	0.54	0.94	0.53	0.96
Λ_2	0.50	0.96	0.46	0.93

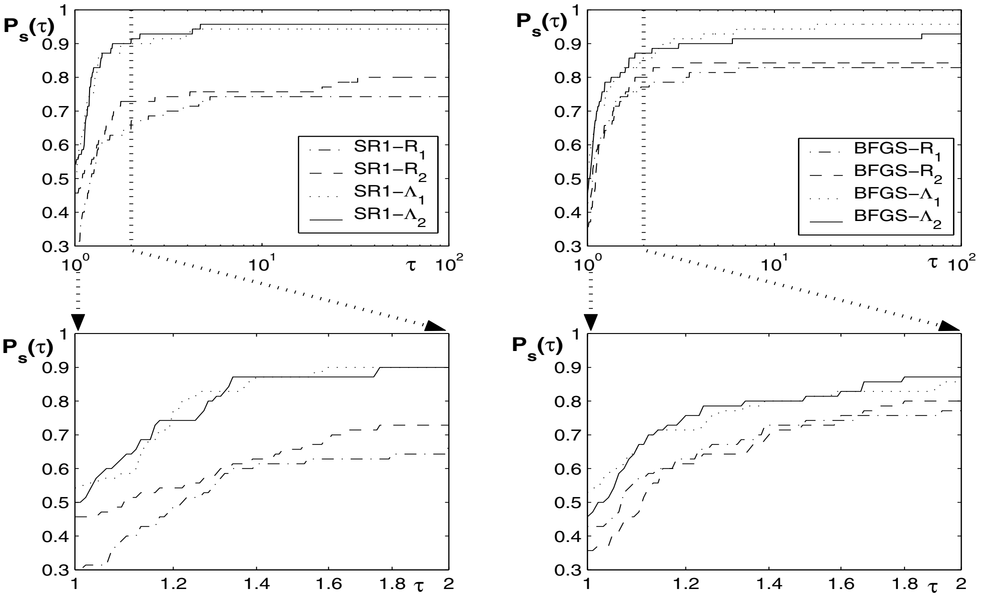


FIG. 5.2. *Complete performance profiles of different versions of the algorithm for* 70 *problems of the* **CUTEr** *test set. The numerical experiments use values of Table* 5.1*, an initial trust-region radius of* $\Delta^{(0)} = 1$*, and a stopping criterion* $\|\nabla_x f(x^{(k)})\| \,/\, \|\nabla_x f(x^{(0)})\| \leq 10^{-6}$*. The two bottom figures are zooms of the two top ones.*

Overall, the variants using Λ-functions perform substantially better from the points of view of both efficiency and robustness than do those based on the usual R-functions (see Tables 5.5 and 5.6 for details). This larger efficiency of the Λ versions is clearly demonstrated by the plots of the complete performance profiles (Figure 5.2) with corresponding curves lying above the curves of the R versions for all values of τ.

5.4. Interaction between the quasi-Newton update rule and the trust-region radius update strategy. As mentioned above in the analysis of the Rosenbrock problem, there is a clear interaction between the update rule of the Hessian matrix and the update rule of the radius of the trust region. The results discussed so far were obtained using an *unconditional* quasi-Newton update; i.e., the approximation of the Hessian matrix was updated at each iteration, whether successful or not. Such an update strategy can be justified by the hope of improving convergence by the use of all available information at successive trial points. However, this approach turns out to be detrimental to the performance of the algorithm in the case of long trial steps, resulting in the “pollution” of the Hessian matrix by bad updates which are

Table 5.5

*Detailed results for Trust-SR*1 *with unconditional quasi-Newton update: number of iterations for each selected problem. In parenthesis: number of successful iterations. Symbols "–" and "**" mean, respectively, that a trial point produces "out of range values" and that the number of iterations exceeds* 10,000. *An exponent star means that convergence occurs towards a different local minimizer with a greater objective function value; this is not regarded as a failure here.*

Name	R_1		R_2		Λ_1		Λ_2	
3PK	66	(50)	71	(49)	68	(50)	67	(46)
AKIVA	20	(13)	19	(12)	15	(11)	15	(11)
ALLINITU	10	(9)	14	(11)	10	(9)	12	(10)
BARD	19	(14)	13	(13)	16	(14)	12	(12)
BEALE	26	(19)	17	(15)	17	(15)	17	(16)
BIGGS6	–		57	(38)	46	(38)	44	(39)
BOX3	9	(9)	18	(11)	9	(9)	14	(12)
BRKMCC	5	(4)	5	(4)	5	(4)	5	(4)
BROWNBS	50	(40)	50	(41)	43	(37)	49	(44)
BROWNDEN	24	(21)	25	(21)	19	(18)	23	(21)
CLIFF	1	(1)	1	(1)	1	(1)	1	(1)
CUBE	**		**		58	(42)	58	(51)
DECONVU	62	(42)	68	(50)	63	(43)	71	(58)
DENSCHNA	9	(9)	9	(9)	9	(9)	9	(9)
DENSCHNB	11	(10)	10	(10)	10	(10)	10	(10)
DENSCHNC	14	(14)	13	(12)	14	(14)	13	(12)
DENSCHND	23	(23)	22	(22)	28	(26)	23	(23)
DENSCHNE	–		**		22	(21)	30	(29)
DENSCHNF	9	(9)	11	(10)	9	(9)	10	(9)
DJTL	5369	(2695)	**		256	(171)	**	
ENGVAL2	53	(40)	**		33	(28)	32	(27)
EXPFIT	**		**		21	(16)	16	(14)
GROWTHLS	29*	(19*)	–		51	(37)	46	(39)
GULF	63	(41)	–		43	(37)	53	(45)
HAIRY	44	(30)	52	(34)	56	(50)	206	(194)
HATFLDD	24	(20)	**		27	(22)	28	(26)
HATFLDE	18	(15)	32	(23)	13	(10)	26	(21)
HEART6LS	**		**		**		5070	(4059)
HEART8LS	**		**		**		1492	(1196)
HELIX	35	(26)	80	(49)	34	(26)	27	(23)
HIELOW	15	(11)	16	(12)	20	(16)	17	(13)
HIMMELBB	3	(3)	3	(3)	3	(3)	3	(3)
HIMMELBF	20	(18)	36	(26)	35	(27)	28	(26)
HIMMELBG	11	(8)	12	(9)	10	(7)	9	(7)
HIMMELBH	7	(7)	7	(7)	7	(7)	7	(7)
HUMPS	221	(138)	178	(118)	138	(108)	307	(266)
HYDC20LS	228	(166)	160	(131)	209	(165)	158	(136)
JENSMP	**		**		38	(31)	33	(29)
KOWOSB	83	(49)	**		31	(26)	31	(22)
LOGHAIRY	276	(187)	1267	(912)	**		**	
MARATOSB	9	(7)	8	(6)	9	(7)	8	(6)
MEXHAT	19	(17)	19	(17)	19	(17)	19	(17)
MEYER3	33*	(21*)	35*	(24*)	38	(32)	40	(32)
OSBORNEA	–		–		–		–	
OSBORNEB	62	(41)	81	(57)	80	(61)	78	(61)
PALMER1C	7	(7)	7	(7)	7	(7)	7	(7)
PALMER1D	8	(8)	9	(9)	8	(8)	9	(9)
PALMER2C	7	(7)	7	(7)	7	(7)	7	(7)
PALMER3C	7	(7)	7	(7)	7	(7)	7	(7)
PALMER4C	7	(7)	7	(7)	7	(7)	7	(7)
PALMER5C	7	(7)	7	(7)	7	(7)	7	(7)
PALMER6C	8	(8)	9	(9)	8	(8)	9	(9)
PALMER7C	5	(5)	6	(6)	5	(5)	6	(6)
PALMER8C	6	(6)	7	(7)	7	(7)	7	(7)
PFIT1LS	–		–		651	(516)	255	(209)
PFIT2LS	–		–		198	(155)	48	(40)
PFIT3LS	–		–		515	(408)	502	(406)
PFIT4LS	–		–		816	(622)	755	(590)
ROSENBR	1399	(706)	**		43	(33)	59	(49)
S308	11	(11)	10	(10)	11	(11)	12	(12)
SINEVAL	3613	(1814)	535	(246)	141	(108)	159	(129)
SISSER	9	(9)	9	(9)	9	(9)	9	(9)
SNAIL	185	(98)	45	(26)	177	(149)	191	(173)
STRATEC	**		91	(59)	87	(63)	75	(58)
TOINTGOR	45	(32)	48	(40)	46	(35)	49	(41)
TOINTPSP	86	(49)	75	(46)	62	(44)	49	(37)
TOINTQOR	25	(24)	25	(22)	25	(24)	25	(22)
VIBRBEAM	72	(33)	110	(68)	58	(29)	55	(31)
YFITU	**		860	(433)	163	(122)	194	(150)
ZANGWIL2	2	(2)	2	(2)	2	(2)	2	(2)

Table 5.6

*Detailed results for Trust-BFGS with unconditional quasi-Newton update: number of iterations for each selected problem. In parenthesis: number of successful iterations. Symbols “−” and “**” mean, respectively, that a trial point produces “out of range values” and that the number of iterations exceeds* 10, 000. *An exponent star means that convergence occurs towards a different local minimizer with a greater objective function value; this is not regarded as a failure here.*

Name	R_1		R_2		Λ_1		Λ_2	
3PK	154	(117)	116	(96)	183	(157)	117	(99)
AKIVA	18	(11)	18	(11)	16	(12)	13	(9)
ALLINITU	11	(10)	10	(9)	11	(10)	10	(9)
BARD	16	(15)	16	(16)	16	(15)	16	(16)
BEALE	13	(13)	16	(16)	13	(13)	13	(13)
BIGGS6	41	(38)	42	(38)	38	(36)	41	(40)
BOX3	9	(9)	15	(15)	9	(9)	15	(15)
BRKMCC	6	(5)	6	(5)	6	(5)	6	(5)
BROWNBS	35	(34)	52	(45)	35	(34)	42	(38)
BROWNDEN	24	(19)	24	(19)	23	(19)	24	(20)
CLIFF	1	(1)	1	(1)	1	(1)	1	(1)
CUBE	**		67	(59)	49	(44)	39	(36)
DECONVU	100	(81)	105	(103)	101	(80)	102	(99)
DENSCHNA	9	(9)	9	(9)	9	(9)	9	(9)
DENSCHNB	9	(9)	9	(9)	9	(9)	9	(9)
DENSCHNC	13	(13)	14	(14)	13	(13)	14	(14)
DENSCHND	15	(14)	21	(20)	20	(19)	25	(24)
DENSCHNE	−		−		34	(33)	37	(35)
DENSCHNF	8	(8)	8	(8)	8	(8)	8	(8)
DJTL	**		**		**		**	
ENGVAL2	33	(29)	31	(28)	29	(26)	27	(25)
EXPFIT	17	(16)	16	(14)	17	(15)	15	(14)
GROWTHLS	−		−		196	(172)	48	(46)
GULF	**		23	(17)	51	(40)	51	(44)
HAIRY	52	(34)	70	(45)	99	(87)	160	(150)
HATFLDD	22	(21)	23	(23)	22	(21)	25	(25)
HATFLDE	29	(27)	21	(21)	29	(27)	22	(22)
HEART6LS	**		**		1634	(1457)	**	
HEART8LS	**		**		876	(801)	346	(324)
HELIX	21	(18)	24	(23)	22	(20)	26	(24)
HIELOW	20	(15)	17	(13)	19	(14)	18	(14)
HIMMELBB	3	(3)	3	(3)	3	(3)	3	(3)
HIMMELBF	34	(33)	36	(35)	35	(34)	32	(31)
HIMMELBG	11	(9)	11	(9)	9	(7)	9	(7)
HIMMELBH	8	(7)	8	(7)	8	(7)	8	(7)
HUMPS	428	(262)	258	(157)	122	(90)	7496	(7441)
HYDC20LS	347	(286)	369	(341)	347	(286)	369	(341)
JENSMP	**		−		45	(39)	36	(32)
KOWOSB	30	(28)	30	(28)	30	(28)	33	(32)
LOGHAIRY	449	(300)	491	(331)	**		**	
MARATOSB	27	(20)	19	(12)	27	(20)	14	(8)
MEXHAT	14	(12)	14	(12)	14	(12)	14	(12)
MEYER3	83^*	(66^*)	66^*	(57^*)	415	(387)	394	(386)
OSBORNEA	−		−		−		−	
OSBORNEB	62	(53)	63	(58)	61	(54)	57	(53)
PALMER1C	20	(17)	32	(29)	20	(17)	32	(29)
PALMER1D	20	(19)	21	(19)	19	(18)	22	(20)
PALMER2C	16	(15)	15	(14)	16	(15)	15	(14)
PALMER3C	16	(15)	27	(26)	15	(14)	27	(26)
PALMER4C	12	(11)	19	(18)	12	(11)	18	(17)
PALMER5C	18	(13)	20	(17)	18	(13)	20	(17)
PALMER6C	18	(17)	12	(11)	18	(17)	12	(11)
PALMER7C	15	(14)	11	(10)	14	(13)	11	(10)
PALMER8C	17	(16)	19	(18)	16	(15)	19	(18)
PFIT1LS	−		−		449	(402)	27	(23)
PFIT2LS	154	(124)	−		357	(326)	**	
PFIT3LS	−		−		921	(846)	332	(315)
PFIT4LS	−		−		470	(432)	522	(498)
ROSENBR	118	(68)	40	(35)	35	(31)	35	(31)
S308	13	(13)	13	(13)	13	(13)	16	(14)
SINEVAL	187	(128)	195	(124)	89	(76)	87	(80)
SISSER	9	(9)	9	(9)	9	(9)	9	(9)
SNAIL	632	(328)	343	(186)	98	(92)	103	(98)
STRATEC	72	(63)	80	(71)	72	(63)	79	(71)
TOINTGOR	75	(61)	76	(73)	75	(61)	76	(74)
TOINTPSP	62	(44)	60	(50)	62	(44)	62	(51)
TOINTQOR	47	(29)	42	(40)	47	(29)	42	(40)
VIBRBEAM	112	(81)	68	(50)	70	(52)	91	(71)
YFITU	89	(75)	107	(82)	76	(67)	80	(72)
ZANGWIL2	2	(2)	2	(2)	2	(2)	2	(2)

difficult to compensate. This is particularly true when the initial trust-region radius is too big for the problem at stake. Thanks to their conservative nature, Λ-functions suppress such long steps and therefore prevent the associated pollution of the Hessian matrix.

Empirical solutions have been proposed to tackle the problem of inaccurate quasi-Newton updates produced by long steps. Byrd, Khalfan, and Schnabel [2] suggest skipping the update when the variation of f at the current iteration is too large, i.e., when

$$f(x^{(k)} + s^{(k)}) - f(x^{(k)}) > 0.5 \left[f(x^{(0)}) - f(x^{(k)}) \right]. \tag{5.4}$$

Implementing condition (5.4) with R-functions does not bring any improvement in the Rosenbrock problem; the decrease of the objective function at the first iteration is so large that (5.4) is never activated. The empirical rule (5.4) is very sensitive to the initial value $f(x^{(0)})$ of the objective function. Also, as used by Byrd, Khalfan, and Schnabel [2], equation (5.4) transforms the first iterations into a pure backtracking line search along the steepest descent direction. Indeed, starting with the identity matrix as the initial guess of the Hessian matrix, the trial points remain along the steepest descent direction as long as the update of $H^{(k)}$ is skipped. If $f(x^{(1)})$ is used rather than $f(x^{(0)})$ in (5.4), the update of the Hessian matrix is skipped at iterations 2–8, and convergence is achieved after 345 iterations with the Trust-BFGS-R_2 algorithm (against 506 without (5.4); cf. Table 5.2). While the initial convergence is largely improved, the positive effect disappears after iteration 8; the accumulated decrease $f(x^{(1)}) - f(x^{(k)})$ of the objective function is too large for (5.4) to be activated before convergence.

In essence, the introduction of Λ-functions achieves the same goal as the empirical rule (5.4) but in a more efficient way. In Rosenbrock's problem, condition (5.4) is never met when Λ-functions are used to update the radius of the trust region. As a result, convergence is achieved in 94 iterations with the Λ_2-function. Also, the new update rules of the trust-region radius do not show the same critical dependency on the initial guess as for (5.4).

Another widely used approach for avoiding bad quasi-Newton updates consists of skipping the update of $H^{(k)}$ when the iteration is unsuccessful (see the discussion in Byrd, Khalfan, and Schnabel [2]). This approach, hereafter referred to as the *conditional update* approach, is easily implemented. As shown by the results of its application to the 70 problems of the test set introduced above (Table 5.7), this conditional update strategy is both robust and efficient: used in combination with the usual R-functions, it induces a drastic decrease of the number of iterations in comparison with the corresponding unconditional update. In this respect, it offers an alternative to the Λ-functions.

Now, the different update rules of the approximation of the Hessian matrix can be combined with the different update strategies of the trust-region radius. The detailed results of the combinations using the conditional quasi-Newton update rule are also listed in Table 5.7 and can be compared with Tables 5.5 and 5.6. In order to compare the different combinations, new performance profiles are computed separately for the SR1 and BFGS update rules. To simplify the analysis, only the staircase R_1- and Λ_1-functions are considered in combination with the unconditional and conditional update strategies. The performance profiles (Figure 5.3) confirm the better performances of Λ-functions with respect to R-functions in the unconditional case. With the usual R_1 versions, the conditional approach also behaves better than the uncon-

Table 5.7

*Detailed results for the conditional Hessian update approach, BFGS, and SR*1 *with quasi-Newton update, and* R_1 *and* Λ_1 *trust-region radius update: number of iterations for each selected problem. Symbols "–" and "∗∗" mean, respectively, that a trial point produces "out of range values" and that the number of iterations exceeds* 10,000.

Name	SR1-R_1		SR1-Λ_1		BFGS-R_1		BFGS-Λ_1	
3PK	57	(35)	57	(35)	166	(143)	147	(123)
AKIVA	18	(10)	18	(11)	18	(10)	18	(11)
ALLINITU	12	(10)	12	(10)	14	(11)	14	(11)
BARD	14	(13)	13	(12)	17	(16)	17	(16)
BEALE	23	(15)	23	(16)	13	(13)	13	(13)
BIGGS6	57	(31)	55	(34)	50	(36)	45	(40)
BOX3	9	(9)	9	(9)	9	(9)	9	(9)
BRKMCC	5	(4)	5	(4)	6	(5)	6	(5)
BROWNBS	159	(96)	114	(72)	69	(49)	69	(48)
BROWNDEN	22	(16)	21	(18)	40	(25)	28	(21)
CLIFF	1	(1)	1	(1)	1	(1)	1	(1)
CUBE	90	(53)	96	(60)	64	(43)	45	(37)
DECONVU	171	(103)	128	(83)	124	(101)	124	(101)
DENSCHNA	9	(9)	9	(9)	9	(9)	9	(9)
DENSCHNB	11	(10)	10	(10)	9	(9)	9	(9)
DENSCHNC	14	(14)	14	(14)	13	(13)	13	(13)
DENSCHND	23	(23)	24	(22)	21	(18)	21	(20)
DENSCHNE	28	(19)	27	(22)	–		34	(29)
DENSCHNF	9	(9)	9	(9)	8	(8)	8	(8)
DJTL	189	(104)	180	(98)	188	(100)	188	(100)
ENGVAL2	79	(42)	60	(38)	29	(26)	31	(27)
EXPFIT	15	(13)	15	(13)	13	(11)	13	(11)
GROWTHLS	55	(32)	70	(44)	42	(26)	161	(141)
GULF	94	(53)	242	(129)	45	(33)	46	(41)
HAIRY	79	(57)	141	(127)	102	(70)	115	(101)
HATFLDD	97	(47)	93	(47)	24	(23)	24	(23)
HATFLDE	42	(24)	35	(23)	22	(21)	22	(21)
HEART6LS	9552	(5247)	∗∗		2518	(1589)	∗∗	
HEART8LS	593	(365)	414	(276)	685	(456)	386	(323)
HELIX	27	(24)	34	(27)	34	(22)	24	(21)
HIELOW	43	(15)	23	(14)	21	(16)	21	(16)
HIMMELBB	3	(3)	3	(3)	3	(3)	3	(3)
HIMMELBF	22	(20)	37	(29)	32	(31)	34	(33)
HIMMELBG	14	(7)	13	(7)	12	(7)	11	(7)
HIMMELBH	7	(7)	7	(7)	10	(8)	10	(8)
HUMPS	274	(170)	227	(173)	143	(90)	247	(185)
HYDC20LS	112	(91)	112	(91)	296	(237)	296	(237)
JENSMP	60	(28)	43	(31)	46	(32)	50	(39)
KOWOSB	36	(22)	60	(36)	51	(30)	38	(33)
LOGHAIRY	∗∗		∗∗		2027	(1360)	∗∗	
MARATOSB	13	(6)	12	(6)	122	(78)	149	(125)
MEXHAT	31	(18)	27	(18)	23	(16)	23	(16)
MEYER3	222	(108)	274	(146)	124	(75)	64	(49)
OSBORNEA	36	(17)	33	(18)	88	(55)	73	(57)
OSBORNEB	142	(86)	100	(68)	73	(58)	68	(58)
PALMER1C	7	(7)	7	(7)	34	(23)	34	(23)
PALMER1D	8	(8)	8	(8)	42	(29)	42	(29)
PALMER2C	7	(7)	7	(7)	34	(22)	34	(22)
PALMER3C	7	(7)	7	(7)	39	(26)	40	(27)
PALMER4C	7	(7)	7	(7)	30	(20)	30	(20)
PALMER5C	7	(7)	7	(7)	26	(19)	26	(19)
PALMER6C	8	(8)	8	(8)	14	(13)	14	(13)
PALMER7C	5	(5)	5	(5)	19	(17)	16	(15)
PALMER8C	6	(6)	7	(7)	21	(19)	22	(20)
PFIT1LS	77	(40)	62	(35)	449	(299)	274	(246)
PFIT2LS	900	(506)	897	(532)	108	(69)	81	(66)
PFIT3LS	1246	(705)	1430	(837)	479	(300)	315	(272)
PFIT4LS	1932	(1095)	2147	(1261)	683	(432)	509	(416)
ROSENBR	70	(46)	96	(64)	52	(33)	38	(32)
S308	11	(11)	11	(11)	13	(13)	13	(13)
SINEVAL	238	(147)	268	(163)	106	(71)	76	(68)
SISSER	9	(9)	9	(9)	9	(9)	9	(9)
SNAIL	222	(148)	256	(190)	152	(96)	106	(95)
STRATEC	149	(85)	172	(103)	84	(70)	84	(69)
TOINTGOR	78	(51)	87	(56)	89	(73)	89	(73)
TOINTPSP	71	(42)	62	(41)	75	(54)	75	(54)
TOINTQOR	27	(25)	27	(25)	56	(37)	56	(37)
VIBRBEAM	134	(59)	109	(58)	125	(78)	114	(79)
YFITU	499	(262)	700	(377)	133	(85)	106	(83)
ZANGWIL2	2	(2)	2	(2)	2	(2)	2	(2)

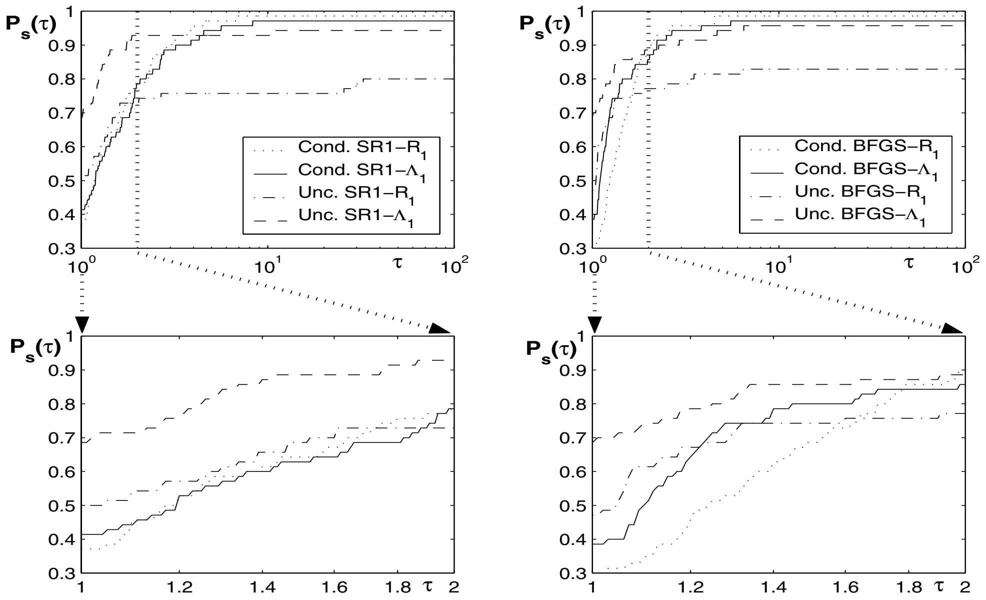


FIG. 5.3. *Complete performance profiles of different versions of the algorithm for* 70 *problems of the* CUTEr *test set. The numerical experiments use values of Table* 5.1, *an initial trust-region radius of* $\Delta^{(0)} = 1$, *and a stopping criterion* $\|\nabla_x f(x^{(k)})\| \,/\, \|\nabla_x f(x^{(0)})\| \le 10^{-6}$. *The two bottom figures are zooms of the two top ones.*

ditional update strategy: the former appears much more robust than the latter. The average speed of the unconditional approach is, however, significantly higher in the BFGS case. Among the four algorithms, the combination of the Λ_1-function for the radius of the trust region with the unconditional quasi-Newton update can be recommended. Although it is slightly less robust than the two conditional variants, it is significantly faster than the other three algorithms.

6. Conclusion. Trust-region methods are increasingly used in applied mathematics and engineering to tackle optimization problems. The update strategy of the radius is likely to have a strong influence on the convergence properties of the algorithm. This paper provides a rather new empirical trust-region radius update strategy based on the idea that very successful iterations may in fact be too successful and negatively affect the subsequent iterations. This occurs when the actual reduction of the objective function is significantly larger than expected reduction from the analysis of the local model function. In this case, unlike the usual approach, we suggest keeping the trust-region radius nearly unchanged.

This strategy is very intuitive and widely applicable. The general convergence properties of trust-region algorithms are preserved by the proposed self-adaptive radius update. Close to convergence, most of the iterations are indeed very successful, and the trust-region constraint becomes irrelevant in the local subproblem. The convergence rate is therefore unaffected by the radius update rule.

Numerical experiments with a quasi-Newton trust-region algorithm using several update rules show that the new update strategy improves the convergence speed by preventing the pollution of the Hessian matrix by inaccurate quasi-Newton updates. While only slightly affecting robustness, the combination of this new strategy with an unconditional quasi-Newton update of the approximation of the Hessian matrix results in the most efficient algorithm.

Acknowledgments. We are grateful to Moré and Sorensen for providing the MINPACK routine GQT, and to Gould, Orban, and Toint for providing the CUTEr interface and test set. We also thank both referees for their useful and detailed comments.

REFERENCES

[1] I. Bongartz, A. R. Conn, N. Gould, and P. L. Toint, *Constrained and unconstrained testing environment*, ACM Trans. Math. Software, 21 (1995), pp. 123–160.

[2] R. H. Byrd, H. F. Khalfan, and R. B. Schnabel, *Analysis of a symmetric rank-one trust region method*, SIAM J. Optim., 6 (1996), pp. 1025–1039.

[3] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust Region Methods*, MPS/SIAM Ser. Optim. 1, SIAM, Philadelphia, 2000.

[4] J. E. Dennis and H. H. W. Mei, *Two new unconstrained optimization algorithms which use function and gradient values*, J. Optim. Theory Appl., 28 (1979), pp. 453–482.

[5] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program., Series A, 91 (2002), pp. 201–213.

[6] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., John Wiley & Sons, New York, 1987.

[7] N. I. M. Gould, S. Lucidi, M. Roma, and P. L. Toint, *Solving the trust-region subproblem using the Lanczos method*, SIAM J. Optim., 9 (1999), pp. 504–525.

[8] N. I. M. Gould, D. Orban, A. Sartenaer, and P. L. Toint, *Sensitivity of trust-region algorithms to their parameters*, 4OR, 3 (2005), pp. 227–241.

[9] N. I. M. Gould, D. Orban, and P. L. Toint, *CUTEr (and SifDec), A constrained and unconstrained testing environment, revisited*, Trans. Amer. Math. Soc. Math. Softw., 29 (2003), pp. 373–394.

[10] L. Hei, *A self-adaptive trust region algorithm*, J. Comput. Math., 21 (2003), pp. 229–236.

[11] J. M. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

SIAM J. Optim.
Vol. 16, No. 2, pp. 563–586

REVISITING ASYNCHRONOUS PARALLEL PATTERN SEARCH FOR NONLINEAR OPTIMIZATION*

TAMARA G. KOLDA†

Abstract. We present a new asynchronous parallel pattern search (APPS) method which is different from that developed previously by Hough, Kolda, and Torczon. APPS efficiently uses parallel and distributed computing platforms to solve science and engineering design optimization problems where derivatives are unavailable and cannot be approximated. The original APPS was designed to be fault-tolerant as well as asynchronous and was based on a peer-to-peer design. Each process was in charge of a single, fixed search direction. Our new version is based instead on a manager-worker paradigm. Though less fault-tolerant, the resulting algorithm is more flexible in its use of distributed computing resources. We further describe how to incorporate a zero-order sufficient decrease condition and handle bound constraints. Convergence theory for all situations (unconstrained and bound constrained as well as simple and sufficient decrease) is developed. We close with a discussion of how the new APPS will better facilitate the future incorporation of linear and nonlinear constraints.

Key words. asynchronous parallel optimization, pattern search, direct search, distributed computing, generating set search

AMS subject classifications. 65K05, 90C56, 65Y05, 68W15, 90C90

DOI. 10.1137/040603589

1. Introduction.

Asynchronous parallel pattern search (APPS) is a variation on parallel pattern search that uses parallel resources more efficiently by eliminating synchronization [12]. Pattern search methods [18, 20, 21, 26] and, more generally, generating set search (GSS) methods [14, 15] are geared toward solving science and engineering optimization problems that lack explicit derivative information. These problems are typically characterized by objective functions based on complex and expensive computer simulations. GSS methods are provably convergent to a stationary point if the underlying objective function is suitably smooth; further, GSS methods often work well in practice (with some theoretical justification; see, e.g., [1]) even on nonsmooth problems; see, e.g., [8] and references therein.

The original APPS algorithm is described in [12], and analysis follows in [16, 17]. The motivation for an asynchronous version of parallel pattern search has not changed from that described in [12]:

> A single synchronization step at the end of every iteration... is neither appropriate nor effective when any of the following factors holds: function evaluations finish in varying amounts of time (even on equivalent processors), the processors employed in the computation possess different performance characteristics, or the processors have varying loads.

*Received by the editors January 27, 2004; accepted for publication (in revised form) July 17, 2005; published electronically December 30, 2005. This research was sponsored by the Mathematical, Information, and Computational Sciences Division at the United States Department of Energy and by Sandia National Laboratory, a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.
http://www.siam.org/journals/siopt/16-2/60358.html

†Computational Sciences and Mathematics Research Department, Sandia National Laboratories, Livermore, CA 94551-9217 (tgkolda@sandia.gov)

However, another driving motivation for the original APPS was the need for a method that was tolerant to various types of failures that might cause synchronous parallel pattern search to completely fail or be extremely slow to converge. To facilitate fault-tolerance, the original APPS algorithm was based on a peer-to-peer model and used PVM [7] as the communication architecture. The new APPS is based instead on a manager-worker paradigm, sacrificing some fault-tolerance in exchange for greater simplicity and flexibility. Further, the new version is based on MPI [25], which many users seem to prefer to PVM. (Is should be noted that some fault-tolerant versions of MPI do exist [4] but such functionality is still rare.)

The sacrifices in terms of fault-tolerance are minimal since checkpointing to disk in the manager-worker version can be used in lieu of a peer-to-peer design. The checkpoint data is small, consisting of only the current best point and corresponding function value. The primary difference between peer-to-peer and checkpointing manager-worker implementations is that the checkpoint version requires some mechanism for restarting (either manual or automated) after a failure, whereas the peer-to-peer continues without any intervention.

In the original APPS, there were multiple agents (i.e., the peers), each of which owned part of the logic of the search. These agents had to correspond with one another regarding algorithmic events (a better point, single direction convergence, and overall convergence), not to mention different types of faults; see [12] for more details. With a single manager process controlling all the logic of the search, these complexities are eliminated. Since the number of worker processes is typically very small (1 to 100 workers) and each communicates infrequently and asynchronously with the manager, it is unlikely that there will be any sort of communication bottleneck at the manager process.

In our description of the new APPS, we present additional modifications for a zero-order (i.e., does not use gradient information) sufficient decrease condition and for bound constraints. The adaptation of a zero-order sufficient decrease condition to pattern search has been discussed in several contexts [14, 23], including a different take on peer-to-peer asynchronous parallel pattern search [6]. In particular, the generalization of pattern search to GSS in [14] was motivated by the desire to incorporate generic globalization strategies, including sufficient decrease, into the framework. The use of a sufficient decrease condition yields greater flexibility in the selection of search directions at each iteration. Handling bound constraints for pattern search has also been the subject of several papers [19, 22, 14]. Some problematic numerical results in the original APPS paper [12, Table 5.6] are the result of not appropriately handling the bound constraints.

The organization of this paper is as follows. In section 2, we review the parallel pattern search algorithm and variants that can be used for sufficient decrease and/or bound constraints; known convergence results are summarized in section 2.4. The new APPS algorithm is presented in section 3, along with its own corresponding variants. An illustrative example of the new APPS algorithm is presented in section 4. Convergence theory follows in section 5. Numerical results comparing the synchronous and asynchronous versions are presented in section 6. We conclude with commentary on the algorithm and associated theory, pointers to its implementation and more numerical results, and ideas for future work in section 7. Table 1.1 summarizes the differences between the new and original versions of APPS. For those familiar with the original APPS, a discussion of the evolution from that one to this one is discussed in Appendix A.

For the purposes of this text, we consider both the unconstrained and bound-

Table 1.1
Comparison of new and original APPS.

	New APPS	**Original APPS**
Parallel model	Manager-worker	Peer-to-peer
Load Balancing	Each evaluation is assigned dynamically to a worker process.	Each evaluation is assigned to the agent process that owns the corresponding direction.
Communication architecture	MPI (or PVM)	PVM, or any *fault-tolerant* architecture
Fault-tolerance achieved by. . .	Checkpointing to disk	Automatic run-time recovery
Provably convergent in unconstrained case	Yes	Yes [17]
. . . in bound constrained case	Yes	Possible, but has not been published
. . . using zero-order sufficient decrease condition	Yes	Possible, but has not been published
Can be modified to run in synchronous mode?	Yes, very easily	Not easily
Can the directions change?	Yes, at successful iterations. Also possible at unsuccessful iterations, but more difficult to implement.	Possible at successful iterations, but very difficult to implement.

constrained nonlinear optimizations problems. The unconstrained problem is given by

$$\min_{x \in \mathbb{R}^n} \quad f(x). \tag{1.1}$$

Here $f : \mathbb{R}^n \to \mathbb{R}$ and $x \in \mathbb{R}^n$. The bound constrained problem is given by

$$\begin{array}{ll} \min & f(x) \\ \text{subject to} & \ell \le x \le u. \end{array} \tag{1.2}$$

The function f is the same as for the unconstrained problem. The upper and/or lower bounds are optional on an element-by-element basis; specifically, ℓ is an n-vector with entries in $\mathbb{R} \cup \{-\infty\}$ and u is an n-vector with entries in $\mathbb{R} \cup \{+\infty\}$. The set Ω denotes the feasible region; i.e.,

$$\Omega = \{x \in \mathbb{R}^n : \ell \le x \le u\}.$$

The unconstrained problem can be thought of as a special case of the bound constrained problem; in other words, $\Omega = \mathbb{R}^n$ in the unconstrained problem.

2. Review of parallel pattern search. We briefly review parallel pattern search (PPS) with simple decrease and its extensions for sufficient decrease and bound constraints. We refer throughout to pattern search, although it might be more accurate to refer to GSS; recall that the generalization of pattern search to GSS was motivated by the desire to bring in different globalization strategies, including sufficient decrease [14]. We conclude by presenting unified convergence results. This review lays the groundwork for the description of the asynchronous methods.

```
Initialization.
Let $\Delta_{\text{tol}} > 0$ be the step length convergence tolerance.
Set $x_0 \in \Omega$ to be some feasible initial guess.
Set $\mathcal{D}_0 = \left\{ d_0^{(1)}, \dots, d_0^{(p_0)} \right\}$ to be the initial set of search directions.
Set $\Delta_0 > \Delta_{\text{tol}}$ to be the initial value of the step length.

Iteration. For $k = 0, 1, \dots$
  STEP 1. Generate a set of trial points corresponding to the search
          directions; i.e.,
              $\mathbf{X}_k = \left\{ x_k + \tilde{\Delta}_k^{(i)} d_k^{(i)} : 1 \le i \le p_k \text{ and } \tilde{\Delta}_k^{(i)} \ge \Delta_{\text{tol}} \right\}.$
          Send all points in $\mathbf{X}_k$ to the evaluation queue.

  STEP 2. Wait until all trial points in the evaluation queue have been
          evaluated. Collect those points in the set $\mathbf{Y}_k$.

  STEP 3. If there exists a trial point in $y_k \in \mathbf{Y}_k$ such that
          $f(y_k) < f(x_k) - \rho(\Delta_k)$, then goto Step 4; else goto Step 5.

  STEP 4. The iteration is successful:
          – Set $x_{k+1} = y_k$.
          – Choose a new $\mathcal{D}_{k+1} = \left\{ d_{k+1}^{(1)}, \dots, d_{k+1}^{(p_{k+1})} \right\}$.
          – Set $\Delta_{k+1} = \Delta_k$.
          – Go to Step 1.

  STEP 5. The iteration is unsuccessful:
          – Set $x_{k+1} = x_k$.
          – Set $\mathcal{D}_{k+1} = \mathcal{D}_k$ (and $p_{k+1} = p_k$).
          – Set $\Delta_{k+1} = \frac{1}{2}\Delta_k$
          – If $\Delta_{k+1} < \Delta_{\text{tol}}$, terminate; else, goto Step 1.
```

FIG. 2.1. *PPS algorithm (with synchronization).*

The generic algorithm is presented in Figure 2.1, and the notation used is as follows. Subscripts denote the iteration index. The vector $x_k \in \mathbb{R}^n$ denotes the *best point* (i.e., the point with the smallest known function value) at the beginning of iteration k. The set

$$\mathcal{D}_k = \left\{ d_k^{(1)}, \dots, d_k^{(p_k)} \right\}$$

denotes the set of *search directions* at iteration k, and the number of search directions in $\mathcal{D}_k$ is denoted by p_k. Superscripts denote the *direction index*, which ranges between 1 and p_k at iteration k. The value Δ_k denotes the *step length* at iteration k, and the values $\tilde{\Delta}_k^{(i)} \in [0, \Delta_k]$, for $i = 1, \dots, p_k$, denote the corresponding *pseudo step lengths*. The function $\rho(\cdot)$ denotes the *forcing function*.

In Step 1 of Figure 2.1, a set of trial points is generated, denoted by $\mathbf{X}_k$. The method for choosing the pseudo step lengths is discussed in detail in the subsections that follow. In general, $\tilde{\Delta}_k^{(i)} = \Delta_k$ unless bound constraints are involved.

In Step 2 of Figure 2.1, the trial points are evaluated, and the results are collected in $\mathbf{Y}_k$. For PPS, $\mathbf{Y}_k = \mathbf{X}_k$ for all k; however, this will not be the case for the asynchronous version in section 3. Step 2 is where parallelism may be employed, in which case the p_k function evaluations are executed in parallel. The algorithm does not go on to the next step until all evaluations have completed, so this is the point of synchronization. Furthermore, this is typically the most computationally expensive step because p_k function evaluations must be computed.

In Step 3 of Figure 2.1, the decrease condition is evaluated. The choice of the function $\rho(\cdot)$ is discussed in detail in section 2.1 and section 2.2. If the decrease condition is satisfied, then the iteration is termed *successful*; otherwise, it is *unsuccessful*.

As an aside, we make the following remark. If multiple points in $\mathbf{Y}_k$ produce decrease, any one can be chosen as y_k without impacting the convergence theory in section 2.4. However, from a practical perspective, a point that yields the smallest function value should be selected.

The algorithm executes Step 4 of Figure 2.1 if the iteration is successful. The next iterate x_{k+1} is updated to be the trial point that produced decrease in the function, y_k. A new set of search directions may also be selected at this point. The search directions must be chosen in a particular way to guarantee that the algorithm will converge. The criteria are detailed in the subsections that follow. For simplicity, a choice that always works is the set of plus and minus unit vectors, i.e., $\mathcal{D}_k = \{\pm e_1, \pm e_2, \ldots, \pm e_n\}$ and $p_k = 2n$ for $k = 1, 2, \ldots$.

The algorithm executes Step 5 of Figure 2.1 if the iteration is unsuccessful. In this case, the step length Δ_k is reduced by a factor of two. In practice, termination occurs when the step length is less then $\Delta_{\text{tol}} > 0$. However, for the purposes of studying the asymptotic behavior of the algorithm, $\Delta_{\text{tol}} = 0$.

2.1. PPS with simple decrease.

2.1. PPS with simple decrease. Let us consider PPS with simple decrease for the unconstrained optimization problem (1.1). The term *simple decrease* means that only $f(y_k) < f(x_k)$ is required in Step 3. In other words, the function $\rho(\cdot)$ is assumed to be identically zero.

Below, we describe the four conditions that specialize the algorithm in Figure 2.1 to be PPS with simple decrease (in the unconstrained case).

The first two conditions have to do with the selection of the search directions, $\mathcal{D}_k$. It is useful to decompose the set of search directions as $\mathcal{D}_k = \mathcal{G}_k \cup \mathcal{H}_k$. The subset $\mathcal{G}_k$ is the core set of search directions (the poll set), while the subset $\mathcal{H}_k$ is a possibly empty set of additional search directions (the search set) [3, 14]. The two subsets play different roles in the analysis and are constructed according to different rules. The subset $\mathcal{G}_k$ is key to the convergence analysis and must satisfy very specific properties as outlined below. On the other hand, the subset $\mathcal{H}_k$ is subject to only those requirements that ensures it does not interfere with convergence. This means the subset $\mathcal{H}_k$ can be populated with directions that might accelerate the search by, for example, allowing very long steps.

Condition 1 requires that the *cosine measure* of the subset $\mathcal{G}_k$ be uniformly bounded; see [14] for a discussion of cosine measure.

Condition 1. Every $\mathcal{G}_k$ positively spans $\mathbb{R}^n$. Furthermore, there exists a constant $c_{\min} > 0$, both independent of k, such that $\kappa(\mathcal{G}_k) \geq c_{\min}$ for all k, where

$$\kappa(\mathcal{G}_k) \equiv \min_{v \in \mathbb{R}^n} \max_{d \in \mathcal{G}_k} \frac{v^T d}{\| v \| \, \| d \|}.$$

Condition 2 requires that the search directions in $\mathcal{G}_k$ be uniformly bounded in length.

Condition 2. There exist $\beta_{\min} > 0$ and $\beta_{\max} > 0$, independent of k, such that for all k the following holds:

$$\beta_{\min} \leq \| d \| \leq \beta_{\max} \qquad \text{for all } d \in \mathcal{G}_k.$$

Parts (a)–(c) of Condition 3 set more specific conditions for selecting the search directions; these conditions are important in the simple decrease case. Essentially, all search directions must be derived from a fixed, finite set $\mathbf{G}$. Part (c) explains how the optional set $\mathcal{H}_k$ must be formed. Condition 3 also requires that the forcing function is identically zero in part (d) and that the pseudo step lengths are chosen appropriately in part (e).

Condition 3 *(rational lattice).*
(a) There exists a finite set $\mathbf{G} = \{d^{(1)}, \ldots, d^{(p)}\}$ such that every vector $d^{(i)} \in \mathbf{G}$ is of the form $d^{(i)} = Bc^{(i)}$, where $B \in \mathbb{R}^{n \times n}$ is a nonsingular matrix and $c^{(i)} \in \mathbb{Q}^n$.
(b) All search directions in $\mathcal{G}_k$ are chosen from $\mathbf{G}$; i.e., $\mathcal{G}_k \subseteq \mathbf{G}$ for all k.
(c) All search directions in $\mathcal{H}_k$ are nonnegative integer combinations of the elements of $\mathbf{G}$;. i.e., $\mathcal{H}_k \subset \left\{ \sum_{i=0}^{p} \xi^{(i)} d^{(i)} \mid \xi^{(i)} \in \{0, 1, 2, \ldots\} \right\}$ for all k.
(d) The forcing function is identically zero, i.e., $\rho(t) \equiv 0$.
(e) All pseudo step lengths satisfy $\tilde{\Delta}_k^{(i)} \in \{0, \Delta_k\}$.

Conditions 1–3 are not difficult to satisfy; consider, for example,

$$\mathcal{D}_k = \{\pm e_1, \pm e_2, \ldots, \pm e_n\} \text{ for all } k.$$

Since we are only considering the unconstrained problem in this subsection, we further assume that the pseudo step lengths are always equal to the step length, i.e.,

$$\tilde{\Delta}_k^{(i)} = \Delta_k \text{ for } i = 1, \ldots, p_k, \; k = 1, 2, \ldots.$$

This is Condition 6, formalized in the discussion of bound constraints.

2.2. PPS with sufficient decrease. Let us consider PPS with sufficient decrease for the unconstrained optimization problem (1.1). In this case, $\rho(\cdot)$ is a nonzero function, in contrast to the simple decrease case.

There are four conditions that specialize the algorithm in Figure 2.1 to be PPS with sufficient decrease (in the unconstrained case). As before, Conditions 1, 2, and 6 are imposed. Condition 3 is replaced instead by the following.

Condition 4 *(forcing function).*
(a) The function $\rho(\Delta)$ is a nonnegative continuous function on $\Delta \in [0, +\infty)$.
(b) The function $\rho(\Delta)/\Delta$ monotonically decreases to zero as $\Delta \downarrow 0$.

A common choice that satisfies Condition 4 is

$$\rho(\Delta) = \alpha \Delta^2,$$

where α is some fixed, positive constant. For a complete discussion of forcing functions for GSS, see [14] and references therein.

2.3. PPS with bound constraints. Let us consider PPS for the bound constrained optimization problem (1.2). Adapting PPS for bound constraints affects the choice of search directions and the choice of the pseudo step lengths. The adaptation is largely independent of the choice of simple or sufficient decrease, except for the particulars of choosing the pseudo step lengths.

Three conditions specialize the algorithm in Figure 2.1 to be PPS with bound constraints.

In the bound constrained case, the search directions must conform to the geometry of the nearby boundary, so Condition 5 requires that $\mathcal{G}_k$ be the coordinate search directions [19]. Condition 5 replaces Condition 1 and Condition 2 since these conditions are trivially satisfied by this choice of $\mathcal{G}_k$. Condition 5 is more restrictive than absolutely necessary and more general selection criteria may be employed; e.g., see the requirements on choosing search directions for general linear constraints in [14, 15].

Condition 5. For all k, we have $\mathcal{G}_k = \{\pm e_1, \ldots, \pm e_n\}$.

The second condition is either Condition 3 or Condition 4, depending on the choice of simple or sufficient decrease.

The final condition is the one we have already referred to, having to do with the choice for pseudo step lengths. Special choices for these values are required in the case of bound constraints. There are several ways that $\tilde{\Delta}_k^{(i)}$ can be chosen so long as Condition 6, which states that the full step is used if possible, is satisfied.

Condition 6. If $x_k + \Delta_k d_k^{(i)} \in \Omega$, then $\tilde{\Delta}_k^{(i)} = \Delta_k$.

Three possible strategies for choosing admissible values for $\tilde{\Delta}_k^{(i)}$ are described in [15] for the case of general linear constraints; we present two here. The simplest choice is the following:

$$\tilde{\Delta}_k^{(i)} = \begin{cases} \Delta_k & \text{if } x_k + \Delta_k d_k^{(i)} \in \Omega, \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

A more sophisticated choice may be employed in the sufficient decrease case: taking the longest possible feasible step. Define $\tilde{\Delta}_k^{(i)}$ as the solution to

$$\begin{array}{ll} \max & \tilde{\Delta} \\ \text{subject to} & 0 \le \tilde{\Delta} \le \Delta_k, \\ & x_k + \tilde{\Delta}\, d_k^{(i)} \in \Omega. \end{array} \tag{2.2}$$

Note that only elementary algebra is required to solve (3.2)

2.4. PPS convergence theory. Before discussing convergence theory, we present some useful definitions and assumptions.

In any practical situation, $\Delta_{\text{tol}} > 0$. However, for the purposes of studying the asymptotic behavior of the algorithm, $\Delta_{\text{tol}} = 0$.

The following assumptions on the function are employed later theorems.

Assumption 1. The set $\mathcal{L}_f(x_0) = \{x \in \Omega : f(x) \leq f(x_0)\}$ is bounded.

Assumption 2. The function f is bounded below on Ω.

Assumption 3. The function f is continuously differentiable on $\mathcal{L}_f(x_0)$.

Assumption 4. The gradient ∇f is Lipschitz continuous with constant M on $\mathcal{L}_f(x_0)$.

Assumption 4 is stronger than necessary and can be replaced by assuming only continuous differentiability. (See the note at the end of section 3.6 in [14].) In that case, we let $M = \omega(x, r)$, where ω denotes the modulus of continuity, i.e.,

$$\omega(x, r) = \max\{\|\nabla f(y) - \nabla f(x)\| \mid \|y - x\| \leq r\}.$$

In constrained optimization, we can measure progress to a KKT point using the following analogue of $\| \nabla f(x) \|$. For $x \in \Omega$, define

$$\chi(x) = \max_{\substack{x+w\in\Omega \\ \| w \| \leq 1}} -\nabla f(x)^T w.$$

The function χ is particularly suitable for the analysis of pattern search (and GSS) methods [14, 15]. It has the following three properties [2]: $\chi(x)$ is continuous, $\chi(x) \geq 0$, and $\chi(x) = 0$ if and only if x is a KKT point. Note that $\chi(x) \equiv \| \nabla f(x) \|$ if $\Omega = \mathbb{R}^n$.

Now that the assumptions and notation have been established, we can present the convergence results for PPS.

Theorem 2.1 (see [14] and references therein). *Consider the optimization problem* (1.1)*, satisfying Assumptions* 1–4*. Let the PPS algorithm in Figure* 2.1 *satisfy Conditions* 1, 2, *either* 3 *or* 4, *and* 6. *Then*

$$\liminf_{k\to+\infty} \| \nabla f(x_k) \| = 0.$$

Theorem 2.2 (see [14, 15] and references therein). *Consider the optimization problem* (1.2)*, satisfying Assumptions* 1–4*. Let the PPS algorithm in Figure* 2.1 *satisfy Conditions either* 3 *or* 4, 5, *and* 6. *Then*

$$\liminf_{k\to+\infty} \chi(x_k) = 0.$$

The next sections describe an asynchronous version of parallel pattern search and its convergence theory.

3. APPS. The premise of APPS is that greater efficiency in parallel processor utilization will enable faster solution in comparison with synchronous pattern search. The original peer-to-peer version has indeed demonstrated faster execution times [12]. The new version is based on a manager-worker design, and that it also demonstrates faster execution times in section 6. A comparison between the manager-worker and peer-to-peer approaches is presented in Table 1.1 and Appendix A.

As mentioned in section 2, the synchronization point in pattern search occurs in Step 2 of Figure 2.1, where the algorithm is required to wait until the evaluation of every trial point is complete before continuing. The difference between the synchronous and asynchronous versions is that the asynchronous version need not wait until all function evaluations complete before moving on to the decision step (Step 3). Instead, the points with incomplete function evaluations are stored in a queue, and the algorithm moves ahead based on the best information available to it.

The flexibility of APPS necessitates a small amount of additional bookkeeping, as observed in [12]. Each trial point must "remember" how it was generated. More specifically, let y be a trial point generated at Step 1 in iteration k using direction i; then the following information is also stored:

- Parent(y) = its parent, x_k,
- ParentFx(y) = its parent's function value, $f(x_k)$,
- Dir(y) = its direction index, i, and
- Step(y) = its step length, $\Delta_k^{(i)}$ (defined below).

It is not necessary that the actual parent be stored; instead, a unique identifier is sufficient. In terms of implementation, the additional storage is for this bookkeeping negligible.

The manager-worker APPS algorithm, presented in in Figure 3.1, has the same structure as PPS in Figure 2.1. We discuss the major differences.

The notation is the same as for PPS, with the following exceptions and additions. There is no longer a single step length Δ_k at step k; instead, there is a step length associated with each direction, denoted by $\Delta_k^{(i)}$. As before, we assume that $\tilde{\Delta}_k^{(i)} = \Delta_k^{(i)}$ in the unconstrained case. We introduce a minimum step length, $\Delta_{\min}$, defined by the integer $\Gamma_{\min}$. There is an evaluation queue which may not be completely emptied in each iteration. Correspondingly, we introduce the set $\mathcal{A}_k$ containing the indices of the search directions that, at the start of iteration k, are "active"; in other words, those directions that have an associated trial point in the evaluation queue. Further, we define $q_{\max}$ to be the maximum number of points the queue holds.

In Step 1 of Figure 3.1, the trial points are generated. The selection criteria for generating new trial points have changed slightly and now take into account whether a given search direction is active. The set $\mathcal{A}_{k+1}$ is set during this step, and it may be reset or modified in Step 4 or Step 5.

In Step 2 of Figure 3.1, a set of evaluated trial points, denoted $\mathbf{Y}_k$, is collected. In contrast to Step 2 of Figure 2.1, this step does not wait until all trial points have been evaluated before moving on. Thus, it may be the case that $\mathbf{Y}_k \neq \mathbf{X}_k$ and further that $\mathbf{Y}_k \not\subseteq \mathbf{X}_k$. Note that if it is always the case that $\mathbf{Y}_k = \mathbf{X}_k$, then the APPS algorithm in Figure 3.1 is equivalent to the PPS algorithm in Figure 2.1 with the exception of how Δ_{k+1} is reset in Step 4, which is inconsequential in practice as discussed below.

Step 3 of Figure 3.1 now selects a *subset* of the trial points to consider for a simple decrease comparison with respect to the current best point. The subset includes those points that satisfy a sufficient decrease condition *with respect to their corresponding*

```
Initialization.
Set $x_0 \in \Omega$ be some feasible initial guess.
Set $\mathcal{D}_0 = \left\{ d_0^{(1)}, \ldots, d_0^{(p_0)} \right\}$ to be the initial set of search directions.
Let $\Delta_{\text{tol}} > 0$ be the step length convergence tolerance.
Set $\Delta_0^{(i)} = \Delta_0 > \Delta_{\text{tol}}$ for $i = 1, \ldots, p_0$ to be the initial step lengths.
Let $\Gamma_{\min} \in \mathbb{Z}$ with $\Gamma_{\min} \geq 0$. Set $\Delta_{\min} = 2^{-\Gamma_{\min}} \Delta_0$.
Set $\mathcal{A}_0 = \emptyset$. Let $q_{\max}$ be the evaluation queue size.

Iteration. For $k = 0, 1, \ldots$
  STEP 1. Generate a (possibly empty) set of trial points
            $$\mathbf{X}_k = \left\{ x_k + \tilde{\Delta}_k^{(i)} d_k^{(i)} \ : \ 1 \leq i \leq p_k, \ \ i \notin \mathcal{A}_k, \ \text{and} \ \tilde{\Delta}_k^{(i)} > \Delta_{\text{tol}} \right\}.$$
          Then, send the set of points $\mathbf{X}_k$ to the evaluation queue.
          Set $\mathcal{A}_{k+1} = \{ i \ : \ \tilde{\Delta}_k^{(i)} > \Delta_{\text{tol}} \}$.

  STEP 2. Collect a nonempty set $\mathbf{Y}_k$ of evaluated trial points.

  STEP 3. Let $\bar{\mathbf{Y}}_k \subseteq \mathbf{Y}_k$ be the subset of trial points that satisfy the sufficient
          decrease condition (see Figure 3.2). If there exists a trial point
          $y_k \in \bar{\mathbf{Y}}_k$ such that $f(y_k) < f(x_k)$, then goto Step 4; else goto Step 5.

  STEP 4. The iteration is successful.
          – Set $x_{k+1} = y_k$.
          – Choose a new $\mathcal{D}_{k+1} = \left\{ d_{k+1}^{(1)}, \ldots, d_{k+1}^{(p_{k+1})} \right\}$.
          – Let $\hat{\Delta} = \textsc{Step}(y_k)$, i.e., the step length that produced $y_k$.
          – Set $\Delta_{k+1}^{(i)} = \max\{\hat{\Delta}, \Delta_{\min}\}$ for $i = 1, \ldots, p_{k+1}$.
          – Reset $\mathcal{A}_{k+1} = \emptyset$.
          – Prune the evaluation queue to $(q_{\max} - p_{k+1})$ or fewer entries.
          – Go to Step 1.

  STEP 5. The iteration is unsuccessful.
          – Set $x_{k+1} = x_k$.
          – Set $\mathcal{D}_{k+1} = \mathcal{D}_k$ (and $p_{k+1} = p_k$).
          – Let $\mathcal{I}_k = \{ \textsc{Dir}(y) : y \in \mathbf{Y}_k \text{ and } \textsc{Parent}(y) = x_k \}$, i.e., the
            directions that generated the points that have $x_k$ as their parent.
          – Update $\mathcal{A}_{k+1} \leftarrow \mathcal{A}_{k+1} \setminus \mathcal{I}_k$ where $\mathcal{A}_{k+1}$ is defined in Step 1.
          – Set $\Delta_{k+1}^{(i)} = \begin{cases} \frac{1}{2}\Delta_k^{(i)}, & \text{if } i \in \mathcal{I}_k \\ \Delta_k^{(i)}, & \text{if } i \notin \mathcal{I}_k \end{cases}$ for $i = 1, \ldots, p_{k+1}$.
          – If $\Delta_{k+1}^{(i)} < \Delta_{\text{tol}}$ for $i = 1, \ldots, p_{k+1}$, terminate. Else, goto Step 1.
```

FIG. 3.1. *Manager-worker APPS algorithm.*

```
For each $y \in \mathbf{Y}_k$
– Let $f(\hat{y}) = \textsc{ParentFx}(y)$, i.e., the function value of the parent of $y$.
– Let $\hat{\Delta} = \textsc{Step}(y)$, i.e., the step length that produced $y$.
– Define the set $\bar{\mathbf{Y}}_k = \Big\{ y \in \mathbf{Y}_k \ : \ f(y) < f(\hat{y}) - \rho(\hat{\Delta}) \Big\}$.
```

FIG. 3.2. *Sufficient decrease condition for Step* 3 *in APPS (Figure* 3.1*).*

parent function values. The specific criteria are presented in Figure 3.2 and discussed in more detail in section 3.1 and section 3.2.

In the case of a successful iteration (Step 4), the primary difference between APPS (Figure 3.1) and PPS (Figure 2.1) is the step length update. In both cases, the step length is updated to be the same as the step length that produced y_k. In PPS, this is simply Δ_k. However, in APPS, the step used to produce y_k is stored as $\textsc{Step}(y_k)$, part of the extra bookkeeping described above. All p_{k+1} step lengths are reset to the larger of either $\textsc{Step}(y_k)$ or the quantity $\Delta_{\min}$. If $\Delta_{\min} \leq \Delta_{\text{tol}}$, this has no affect in practice, but it is important in the convergence theory (where it cannot be less than Δ_{tol} since $\Delta_{\text{tol}} = 0$). A successful iteration clears the active directions, so $\mathcal{A}_{k+1}$ is reset to the empty set. At this point, the evaluation queue needs to be pruned to prevent it from growing too large; such a measure has analytical (see Condition 9) as well as practical benefits. Any or all points may be pruned.

In the case of an unsuccessful iteration (Step 5 in Figure 3.1), the step lengths are reduced individually depending on the trial points in $\mathbf{Y}_k$. Specifically, each evaluated trial point is considered, and if the trial point's parent is not x_k, then it is discarded. (Recall that keeping track of the parent is part of the bookkeeping described above.) Otherwise, the corresponding step is reduced by a factor of two. The correct step is identified by the direction index that was used to generate the trial point (also part of the bookkeeping). Termination is essentially the same, except that there are now p_{k+1} steps, all of which must be less than the specified tolerance before the algorithm terminates.

3.1. APPS with simple decrease.

In the simple decrease version of APPS, Conditions 1–3 are imposed as in the synchronous version discussed in section 2.1. Condition 6 is replaced with Condition 7 (see section 3.3); the new condition handles the multiple, possibly different step lengths.

In the simple decrease case, we can assume, without loss of generality, that $\bar{\mathbf{Y}}_k = \mathbf{Y}_k$ in Step 3. The reasoning is that it cannot be the case that a trial point y satisfies $f(y) < f(x_k)$ but not $f(y) < f(\hat{y})$ (where $\hat{y}$ is the parent of y). Since $\hat{y}$ is a previous best point, it must be true that $f(x_k) \leq f(\hat{y})$.

3.2. APPS with sufficient decrease.

In the sufficient decrease version of APPS, Conditions 1, 2, 4, and 7 (the replacement for Condition 6) are enforced.

Implementing sufficient decrease in an asynchronous environment adds a layer of difficulty because the sufficiency condition is with respect to the parent of the trial point. There is no assurance that x_k is the parent of the trial point, as in the synchronous case. Consequently, in the asynchronous case, determining whether or not an evaluated trial point is a new best point becomes a two-step process. First, the point is checked to see if it satisfies a sufficient decrease condition with respect to its parent's function value (see Figure 3.2). Second, it is assessed to see if simple decrease with respect to the current x_k is satisfied.

For example, consider an evaluated trial point y at iteration k. Let $f(\hat{y}) = \textsc{ParentFx}(y)$ be the parent function value, and let $\hat{\Delta} = \textsc{Step}(y)$ be the step length that produced y. To be a candidate for new best point, y must satisfy

$$f(y) < \min\{f(\hat{y}) - \rho(\hat{\Delta}), f(x_k)\}.$$

3.3. APPS with bound constraints. Bound constraints are handled essentially the same as before. Now, however, Condition 6 must be modified to reflect the p_k independent step lengths. Condition 7 results.

Condition 7. If $x_k + \Delta_k^{(i)} d_k^{(i)} \in \Omega$, then $\tilde{\Delta}_k^{(i)} = \Delta_k^{(i)}$.

Similarly, the step calculations in (2.1) and (2.2) need to be modified. The following choice is suitable for either simple or sufficient decrease [15]:

$$\tilde{\Delta}_k^{(i)} = \begin{cases} \Delta_k^{(i)} & \text{if } x_k + \Delta_k^{(i)} d_k^{(i)} \in \Omega, \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

In the sufficient decrease case, taking the longest possible feasible step is an alternative [15]. Define $\tilde{\Delta}_k^{(i)}$ as the solution to

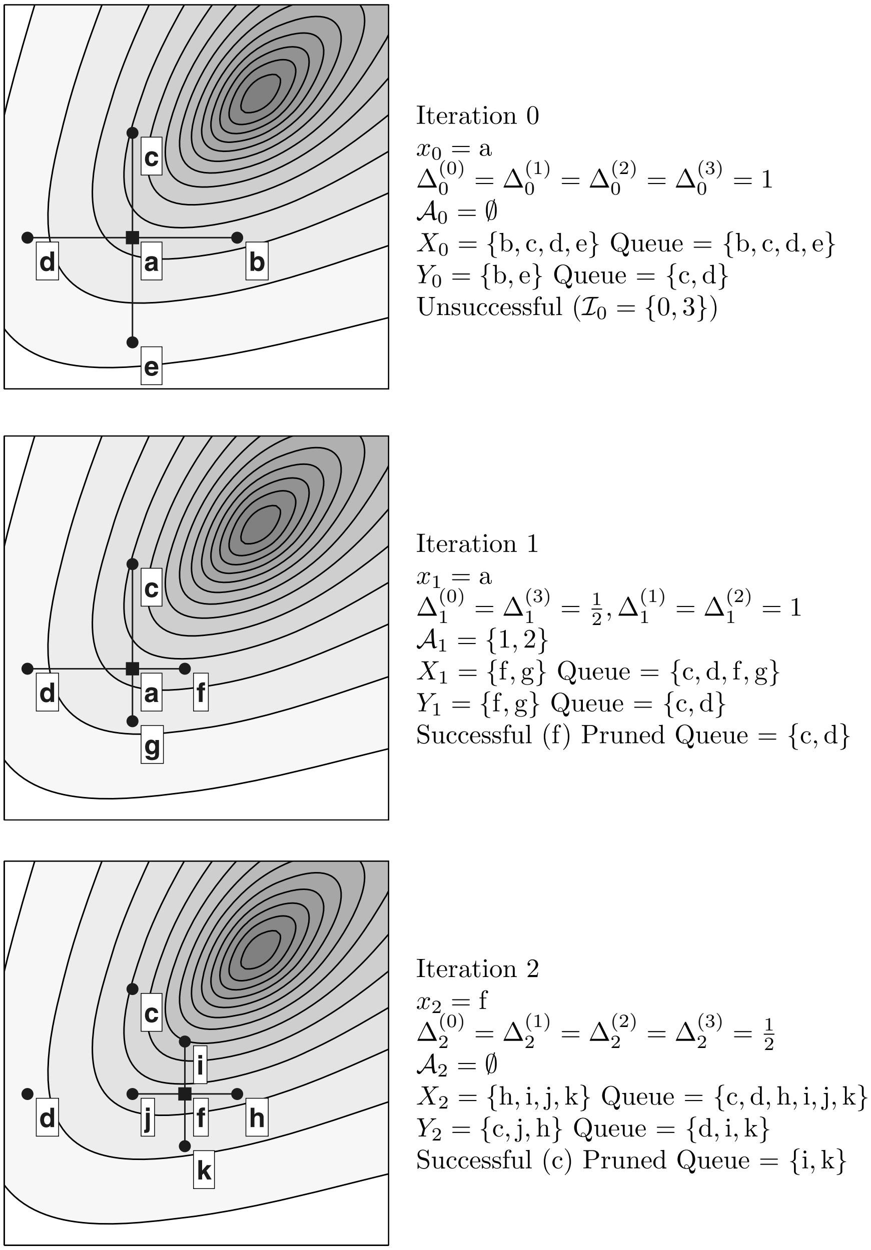
$$\begin{array}{ll} \max & \tilde{\Delta} \\ \text{subject to} & 0 \le \tilde{\Delta} \le \Delta_k^{(i)}, \\ & x_k + \tilde{\Delta}\, d_k^{(i)} \in \Omega. \end{array} \tag{3.2}$$

4. An illustrated example of APPS. A two-dimensional example is presented in Figures 4.1 and 4.2. The contour plot of the objective function uses darker shading to indicate lower function values. Each figure represents the state of the algorithm at an iteration. The square denotes the best point (i.e., x_k) at that iteration, and the circles denote points in the evaluation queue after Step 1 is completed. The lines denote the search directions. For simplicity, we use the same set of search directions throughout: $\mathcal{D}_k = \{e_1, e_2, -e_1, -e_2\}$. The points are labeled with letters, and the algorithm is initialized with the starting point $x_0 = \mathbf{a}$ and an initial step length of $\Delta_0 = 1$.

Before we continue, it is important to note the following. At each iteration, the set of evaluated trial points returned in Step 2 could be any nonempty subset of points in the evaluation queue—the choice of this subset is not controlled by the APPS algorithm. Thus, the set $\mathbf{Y}_k$ at each iteration can be interpreted as the result of random chance. (In truth, we have crafted the selection in this example to demonstrate certain features of the algorithm.) The algorithm makes no assumption that the points in the evaluation queue finish being evaluated in any particular order.

A couple of algorithmic choices also influence our example. In Step 2, we assume that no sufficient decrease criteria is employed (i.e., $\rho \equiv 0$) so that $\bar{\mathbf{Y}}_k \equiv \mathbf{Y}_k$ for all k. In Step 4, we assume that $q_{\max} = 6$.

Iteration 0 illustrates an unsuccessful iteration. We assume that only two evaluations ($\mathbf{b}$ and $\mathbf{e}$) are returned in Step 2. Neither $\mathbf{b}$ nor $\mathbf{e}$ improves the function value, so the iteration is unsuccessful. The parent of both $\mathbf{b}$ and $\mathbf{e}$ is $x_0 = \mathbf{a}$ and their corresponding direction indices are 0 and 3, thus $\mathcal{I}_0 = \{0, 3\}$ in Step 5. The



Iteration 0
$x_0 = \mathrm{a}$
$\Delta_0^{(0)} = \Delta_0^{(1)} = \Delta_0^{(2)} = \Delta_0^{(3)} = 1$
$\mathcal{A}_0 = \emptyset$
$X_0 = \{\mathrm{b,c,d,e}\}$ Queue $= \{\mathrm{b,c,d,e}\}$
$Y_0 = \{\mathrm{b,e}\}$ Queue $= \{\mathrm{c,d}\}$
Unsuccessful ($\mathcal{I}_0 = \{0,3\}$)

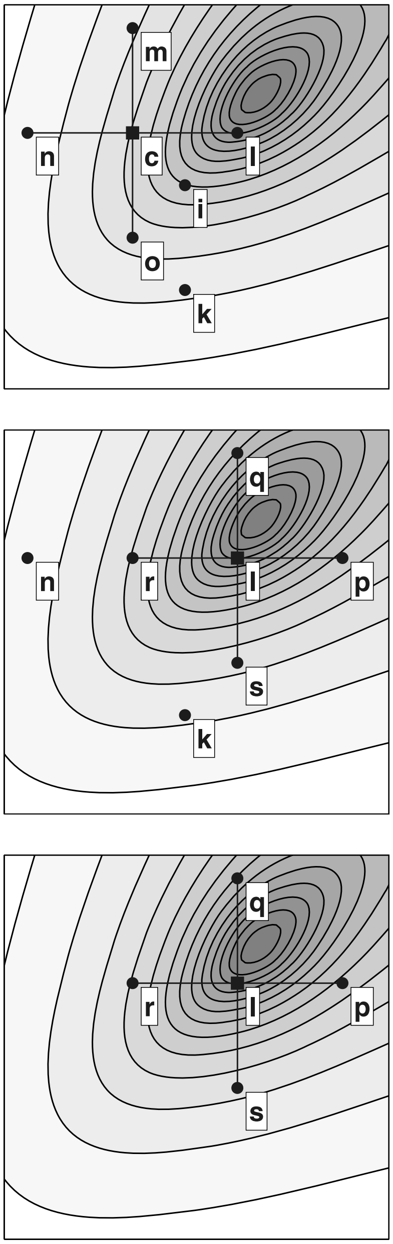
Iteration 1
$x_1 = \mathrm{a}$
$\Delta_1^{(0)} = \Delta_1^{(3)} = \frac{1}{2}, \Delta_1^{(1)} = \Delta_1^{(2)} = 1$
$\mathcal{A}_1 = \{1,2\}$
$X_1 = \{\mathrm{f,g}\}$ Queue $= \{\mathrm{c,d,f,g}\}$
$Y_1 = \{\mathrm{f,g}\}$ Queue $= \{\mathrm{c,d}\}$
Successful (f) Pruned Queue $= \{\mathrm{c,d}\}$

Iteration 2
$x_2 = \mathrm{f}$
$\Delta_2^{(0)} = \Delta_2^{(1)} = \Delta_2^{(2)} = \Delta_2^{(3)} = \frac{1}{2}$
$\mathcal{A}_2 = \emptyset$
$X_2 = \{\mathrm{h,i,j,k}\}$ Queue $= \{\mathrm{c,d,h,i,j,k}\}$
$Y_2 = \{\mathrm{c,j,h}\}$ Queue $= \{\mathrm{d,i,k}\}$
Successful (c) Pruned Queue $= \{\mathrm{i,k}\}$

Fig. 4.1. *Example APPS iterations: part* 1.

step lengths corresponding to those directions are reduced by a factor of 2. Note that points **c** and **d** remain in the evaluation queue.

Iteration 1 illustrates a successful iteration. In Step 1, this iteration generates only two new trial points (**f** and **g**) because Directions 1 and 2 are already active (i.e., $\mathcal{A}_1 = \{1,2\}$). In Step 2, we assume points **f** and **g** are returned. Since **f** reduces the function value, this iteration is successful. All step lengths for the next iteration are



Iteration 3
$x_3 = \mathrm{c}$
$\Delta_3^{(0)} = \Delta_3^{(1)} = \Delta_3^{(2)} = \Delta_3^{(3)} = 1$
$\mathcal{A}_3 = \emptyset$
$X_3 = \{\mathrm{l, m, n, o}\}$ Queue $= \{\mathrm{i, k, l, m, n, o}\}$
$Y_3 = \{\mathrm{i, l, m, o}\}$ Queue $= \{\mathrm{k, n}\}$
Successful (l) Pruned Queue $= \{\mathrm{k, n}\}$

Iteration 4
$x_4 = \mathrm{l}$
$\Delta_4^{(0)} = \Delta_4^{(1)} = \Delta_4^{(2)} = \Delta_4^{(3)} = 1$
$\mathcal{A}_4 = \emptyset$
$X_4 = \{\mathrm{p, q, r, s}\}$ Queue $= \{\mathrm{k, n, p, q, r, s}\}$
$Y_4 = \{\mathrm{k, n}\}$ Queue $= \{\mathrm{p, q, r, s}\}$
Unsuccessful ($\mathcal{I}_4 = \emptyset$)

Iteration 5
$x_5 = \mathrm{l}$
$\Delta_5^{(0)} = \Delta_5^{(1)} = \Delta_5^{(2)} = \Delta_5^{(3)} = 1$
$\mathcal{A}_5 = \{0, 1, 2, 3\}$
$X_5 = \emptyset$ Queue $= \{\mathrm{p, q, r, s}\}$
...

FIG. 4.2. *Example APPS iterations: part* 2.

reset to the length of the step length that produced **f** (i.e., $\hat{\Delta} = \frac{1}{2}$). No pruning of the queue is necessary.

Iteration 2 illustrates "disconnected" points in the evaluation queue and a successful iteration that results from one of these disconnected points. Two points remain in the evaluation queue, and four new trial points are generated and added in Step 1. Because $x_2 \neq x_1$, the older points in the queue are no longer connected to the current best point and so are referred to as disconnected. In Step 2, we assume the evaluation for the point **c** is finally returned (along with **j** and **h**) and results in another successful step. All step lengths for the next iteration are set to the step length that produced **c** (i.e., $\hat{\Delta} = 1$), and this step is from Iteration 0. This time, the evaluation queue is pruned by removing the oldest point, **d**.

Iteration 3 illustrates two points with improved function values returning simultaneously (**i** and **l**). As with synchronous parallel pattern search, we will assume that we take the best one, although this is not strictly necessary in terms of the theory.

Iteration 4 illustrates an unsuccessful iteration that results in no changes and thus no new trial points in the next iteration. Here, points **k** and **n** finish their evaluations and the result is an unsuccessful iteration. However, since both points are disconnected (i.e., neither has **l** as its parent), no step lengths are reduced in Step 5.

At the beginning of Iteration 5, no new trial points are generated, and four points remain in the evaluation queue. The process continues from there, marching toward a local minimizer.

5. APPS convergence theory. We develop convergence theory for APPS, concluding with results analogous to Theorems 2.1 and 2.2. The analysis borrows heavily from [14, 15, 17]. We begin in section 5.1 by determining bounds on $\| \nabla f(x_k) \|$ and $\chi(x_k)$ in terms of the step lengths. Next, in section 5.2, we present some results showing that a subsequence of the step lengths go to zero. Finally, in section 5.3, we give the convergence results.

It is implicitly assumed in the discussion of the asymptotic behavior that $\Delta_{\text{tol}} = 0$ in Figure 3.1.

We make explicit the bound on the number p_k of search directions in $\mathcal{D}_k$ in Condition 8. This is an implicit assumption in PPS.

Condition 8. There exists $p_{\max}$, independent of k, such that for all k, $p_k \leq p_{\max}$.

We also need to ensure that a trial point cannot languish in the evaluation queue indefinitely. This is also an implicit assumption in PPS.

Condition 9. If a trial point is submitted to the evaluation queue at iteration k, either its evaluation will have completed or it will have been pruned from the evaluation queue by iteration $k + \eta$.

Condition 9 is not saying that every function evaluation requires η iterations; instead, this is an upper bound on the number of iterations. In fact, the value of η may be quite large. A sticky point here is that an iteration does not necessarily correspond to a unit of time, so it is difficult to specify a maximum number of iterations for a function evaluation. However, if we assume that a unit of time is associated with an iteration, this assumption can be enforced as follows. Without loss of generality, let the minimum iteration time correspond to 1 time unit. Now, suppose that there are w workers available for computing function evaluations and that the maximum number of time units required to compute a single function evaluation on a single worker is ϕ. Next, assume that trial points submitted to the evaluation queue are sent to the workers in order (although there is no assumption that the function evaluations finish in order). Finally, assume the maximum queue size is $q_{\max} \geq p_{\max}$ and is always pruned to a size no greater than $(q_{\max} - p_{k+1})$ for any successful iteration. Then, η

can explicitly be computed as

$$\eta = \phi \left\lceil \frac{q_{\max}}{w} \right\rceil .$$

From an implementation point of view, the critical requirement is that the evaluation queue cannot be allowed to grow too large, and so the pruning in Step 4 is necessary for enforcing Condition 9.

5.1. Bounding the measure of stationarity.

Theorem 5.1, below, applies to the unconstrained case and bounds the norm of the gradient as a function of the step length. This result and its proof are nearly identical to Theorem 3.3 in [14]. The difference is identifying those iterations for which such a bound can be shown. The necessary condition is that there must have been at least one contraction in every direction since the last successful iteration.

THEOREM 5.1. *Consider the optimization problem* (1.1)*, satisfying Assumptions* 3–Lipschitz. *Let the APPS algorithm in Figure* 3.1 *satisfy Conditions* 1*,* 2*, either* 3 *or* 4*, and* 7*. For every k such that*

$$\hat{\Delta}_k \equiv \max_{1 \le i \le p_k} \left\{ 2\Delta_k^{(i)} \right\} \le \Delta_{\min}, \tag{5.1}$$

we have

$$\| \nabla f(x_k) \| \le \frac{1}{c_{\min}} \left[M \hat{\Delta}_k \beta_{\max} + \frac{\rho(\hat{\Delta}_k)}{\hat{\Delta}_k \beta_{\min}} \right]. \tag{5.2}$$

Proof. By hypothesis (5.1), $\Delta_k^{(i)} < \Delta_{\min}$ for all $i = 1, \ldots, p_k$. This implies that there has been at least one contraction along each direction since that last successful iteration, so

$$0 \le f(x_k + 2\Delta_k^{(i)} d_k^{(i)}) - f(x_k) + \rho(2\Delta_k^{(i)}) \text{ for } i = 1, \ldots, p_k. \tag{5.3}$$

The value of $2\Delta_k^{(i)}$ comes in because the current value of $\Delta_k^{(i)}$ is half of that for which the contraction was done. Also note that it is assumed $\tilde{\Delta}_k^{(i)} = \Delta_k^{(i)}$ by Condition 7.

Since, by hypothesis, Condition 1 is satisfied, there exists an $\bar{\imath} \in \{1, \ldots, p_k\}$ such that

$$c_{\min} \| \nabla f(x_k) \| \, \| d_k^{(\bar{\imath})} \| \le -\nabla f(x_k)^T d_k^{(\bar{\imath})}. \tag{5.4}$$

Employing Assumption 3, the mean value theorem can be applied to (5.3) for $i = \bar{\imath}$ to conclude that there exists $\bar{\alpha} \in [0, 1]$ such that

$$f(x_k + 2\Delta_k^{(\bar{\imath})} d_k^{(\bar{\imath})}) - f(x_k) = 2\Delta_k^{(\bar{\imath})} \nabla f(x_k + \bar{\alpha} 2\Delta_k^{(\bar{\imath})} d_k^{(\bar{\imath})})^T d_k^{(\bar{\imath})}. \tag{5.5}$$

Combining (5.3) and (5.5), dividing through by $2\Delta_k^{(\bar{\imath})}$, and subtracting $\nabla f(x_k)^T d_k^{(\bar{\imath})}$ from both sides yields

$$\begin{aligned} -\nabla f(x_k)^T d_k^{(\bar{\imath})} &\le \left(\nabla f(x_k + \bar{\alpha}\, 2\Delta_k^{(\bar{\imath})} d_k^{(\bar{\imath})}) - \nabla f(x_k) \right)^T d_k^{(\bar{\imath})} + \frac{\rho(2\Delta_k^{(\bar{\imath})})}{2\Delta_k^{(\bar{\imath})}} \\ &\le \| \nabla f(x_k + \bar{\alpha}\, 2\Delta_k^{(\bar{\imath})} d_k^{(\bar{\imath})}) - \nabla f(x_k) \| \, \| d_k^{(\bar{\imath})} \| + \frac{\rho(2\Delta_k^{(\bar{\imath})})}{2\Delta_k^{(\bar{\imath})}}. \end{aligned}$$

Using (5.4) to replace the left-hand side and dividing through by $\| d_k^{(\bar{\imath})} \|$, we now have

$$c_{\min} \| \nabla f(x_k) \| \leq \| \nabla f(x_k + \bar{\alpha} 2\Delta_k^{(\bar{\imath})} d_k^{(\bar{\imath})}) - \nabla f(x_k) \| + \frac{1}{\| d_k^{(\bar{\imath})} \|} \frac{\rho(2\Delta_k^{(\bar{\imath})})}{2\Delta_k^{(\bar{\imath})}}. \tag{5.6}$$

Since ∇f is Lipschitz (Assumption 4), the norm of any search direction is bounded (Condition 2), and $\bar{\alpha} \in [0,1]$, it follows that

$$\| \nabla f(x_k + \bar{\alpha} 2\Delta_k^{(\bar{\imath})} d_k^{(\bar{\imath})}) - \nabla f(x_k) \| \leq M \left(\bar{\alpha}\, 2\Delta_k^{(\bar{\imath})} \| d_k^{(\bar{\imath})} \| \right) \leq M \hat{\Delta}_k \beta_{\max}. \tag{5.7}$$

Now, either ρ is identically zero (Condition 3) or $\rho(t)/t$ is monotonically decreasing as $t \downarrow 0$ (Condition 4). In either case,

$$\frac{1}{\| d_k^{(\bar{\imath})} \|} \frac{\rho(2\Delta_k^{(\bar{\imath})})}{2\Delta_k^{(\bar{\imath})}} \leq \frac{1}{\beta_{\min}} \frac{\rho(\hat{\Delta}_k)}{\hat{\Delta}_k}. \tag{5.8}$$

Note that the lower bound in Condition 2 is also employed in the above inequality.

Finally, combining (5.6), (5.7), and (5.8) and dividing by $c_{\min}$ yields (5.2). Hence, the claim. □

A similar result can be proved in the constrained case that is nearly identical to Theorem 4.4 in [15]. The same adaptations are used as in the unconstrained case, so the proof is left to the reader.

Theorem 5.2. *Consider the optimization problem* (1.2)*, satisfying Assumptions* 1*,* 3*, and* 4*. Let the APPS algorithm in Figure* 3.1 *satisfy Conditions either* 3 *or* 4*,* 5*, and* 7*. Let $\epsilon_\star > 0$ be given. Then there exists a constant c such that, for every k that satisfies*

$$\hat{\Delta}_k \equiv \max_{1 \leq i \leq p_k} \left\{ 2\Delta_k^{(i)} \right\} \leq \max \left\{ \Delta_{\min}, \frac{\epsilon_\star}{\beta_{\max}} \right\}, \tag{5.9}$$

we have

$$\chi(x_k) \leq c \left[M \hat{\Delta}_k \beta_{\max} + \frac{\rho(\hat{\Delta}_k)}{\hat{\Delta}_k \beta_{\min}} \right]. \tag{5.10}$$

5.2. Globalization. Before we proceed to the globalization results, it is necessary to introduce some additional notation and assumptions.

We define $\Gamma_k^{(i)}$ for all k and $i = 1, \ldots, p_k$ as

$$\Gamma_k^{(i)} = \log_2 \left(\frac{\Delta_0}{\Delta_k^{(i)}} \right). \tag{5.11}$$

We can conclude that $\Gamma_k^{(i)} \in \mathbb{Z}$ because any $\Delta_k^{(i)}$ is an integral power of 2 times the initial step size, i.e.,

$$\Delta_{k+1}^{(i)} = 2^{-\Gamma_k^{(i)}} \Delta_0.$$

The following Lemma 5.3 applies to APPS with a sufficient decrease condition. Because x_k is not necessarily the parent of x_{k+1}, the proof is somewhat different than its synchronous analogue, Theorem 3.4 in [14].

Additional notation is required for the proof. For any successful iteration k, a set of ancestors may be constructed for the point x_{k+1}. Let Π_k denote the iteration indices of the ancestors of x_{k+1} as well as $(k+1)$ itself, and let ℓ_k denote the number of ancestors. (The size of Π_k will be $\ell_k + 1$). To illustrate, consider again the example of section 4. Iterations 1, 2, and 3 are successful and yield the following ancestor sets:

$$\begin{array}{ll} \Pi_1 = \{0, 2\}, & \ell_1 = 1, \\ \Pi_2 = \{0, 3\}, & \ell_2 = 1, \\ \Pi_3 = \{0, 1, 4\} & \ell_3 = 2. \end{array}$$

It is important to note that 0 is necessarily in every set Π_k since x_0 is an ancestor to every point.

LEMMA 5.3. *Consider the optimization problem* (1.1) *or* (1.2)*, satisfying Assumption* 2*. Let the APPS algorithm in Figure* 3.1 *satisfy Conditions* 4*,* 8*, and* 9*. Then there exists an index j and a set $\mathcal{K} \subset \{1, 2, \ldots\}$ such that*

$$\lim_{k \in \mathcal{K}} \Gamma_k^{(j)} = +\infty.$$

Proof. Suppose the lemma is false. Then there exists $\Gamma_\star$ such that $\Gamma_k^{(i)} < \Gamma_\star$ for all k and $i = 1, \ldots, p_k$. Consequently, the step lengths are bounded below:

$$\Delta_k^{(i)} \geq \Delta_\star = 2^{-\Gamma_\star} \Delta_0 \text{ for all } k \text{ and } i = 1, \ldots, p_k. \tag{5.12}$$

Then, by Condition 4, the forcing term is bounded below as well:

$$\rho(\Delta_k^{(i)}) \geq \rho_\star = \rho(\Delta_\star) \text{ for all } k \text{ and } i = 1, \ldots, p_k. \tag{5.13}$$

Suppose k is a successful iteration, and let $\Pi_k = \{i_1, i_2, \ldots, i_{\ell_k+1}\}$. Since each child-parent pair satisfies the sufficient decrease condition, we can apply a telescoping sum argument and (5.13) to obtain

$$f(x_{k+1}) - f(x_0) = \sum_{j=1}^{\ell_k} \left\{ f\left(x_{i_{j+1}}\right) - f\left(x_{i_j}\right) \right\} \geq \ell_k \; \rho_\star. \tag{5.14}$$

Another consequence of the lower bound on the step lengths in (5.12) is that each parent can only have a finite number of children. Specifically, a parent can have no more than $c = p_{\max} (\Gamma_\star + 1)$ children where the bound $p_{\max}$ comes from Condition 8. Thus, if iteration k is successful, x_{k+1} must have at least $\lceil k/c \rceil$ ancestors. Combining this information with (5.14) yields

$$f(x_{k+1}) \geq k \left(\frac{\rho_\star}{c} \right) + f(x_0).$$

Let $\mathcal{S}$ denote the subsequence of successful iterates. By Condition 9, the maximum number of iterations to evaluate a single trial point is bounded. This coupled with the method by which step lengths are updated implies that there must be infinitely many successful steps, i.e., $\mathcal{S}$ is infinite. Thus,

$$\lim_{k \in \mathcal{S}} f(x_{k+1}) \geq \lim_{k \in \mathcal{S}} k \left(\frac{\rho_\star}{c} \right) + f(x_0) = +\infty.$$

This contradicts Assumption 2. Hence, the claim. $\square$

Before we can establish a result analogous to Lemma 5.3 for the simple decrease case, we first state a result regarding the structure of the iterates. It is a standard result, so no proof is provided here; see instead, e.g., [14].

PROPOSITION 5.4 (see [17]). *Consider the optimization problem* (1.1) *or* (1.2). *Consider the APPS algorithm in Figure* 3.1 *satisfying Condition* 3. *Let* $\Gamma > 0$ *be a constant. Then, for any* k *with*

$$\Gamma \leq \Gamma_j^{(i)} \textit{ for all } j \leq k, i = 1, \ldots, p_j,$$

the following holds:

$$x_{k+1} = x_0 + 2^{-\Gamma}\ \Delta_0 \sum_{i=1}^{p} \zeta_k(i, \Gamma)\ d^{(i)}, \tag{5.15}$$

where $\zeta_k(i, \Gamma) \in \mathbb{Z}$ *for each* $i = 1, \ldots, p$ *and* $k = 0, 1, 2, \ldots$.

Given this result, the fact that the $\zeta_k(i, \Gamma)$ are integral, and the set $\mathbf{G}$ is as described in Condition 3, all iterates lie on on the lattice

$$\mathcal{M}(x_0, \Delta_0, \mathbf{G}, \Gamma) = \left\{ x_0 + 2^{-\Gamma} \Delta_0 \sum_{i=1}^{p} \zeta^{(i)} d^{(i)} \ :\ i \in \mathbb{Z} \right\}.$$

We can now present our result.

LEMMA 5.5. *Consider the optimization problem* (1.1) *or* (1.2), *satisfying Assumption* 1. *Let the APPS algorithm in Figure* 3.1 *satisfy Conditions* 3 *and* 9. *Then, there exists an index* j *and a set* $\mathcal{K} \subset \{1, 2, \ldots\}$ *such that*

$$\lim_{k \in \mathcal{K}} \Gamma_k^{(j)} = +\infty.$$

Proof. Suppose not. Then there exists $\Gamma_\star$ such that $\Gamma_k^{(i)} < \Gamma_\star$ for all k and $i = 1, \ldots, p_k$. By Proposition 5.4, every iterate must lie on the lattice $\mathcal{M}(x_0, \Delta_0, \mathbf{G}, \Lambda_\star)$. On the other hand, by Assumption 1, every iterate must lie in the bounded set $\mathcal{L}_f(x_0)$. The intersection of $\mathcal{M}(x_0, \Delta_0, \mathbf{G}, \Gamma_\star)$ and $\mathcal{L}_f(x_0)$ is finite, so every successful iterate is drawn from a finite set. Next, observe that a successful point can be successful only once because Step 3 in Figure 3.1 requires strict improvement. Therefore, there can be only finitely many successful iterates; let $\hat{k}$ denote the last successful iterate.

After iteration $\hat{k}$, the set of search directions does not change. Further, by Condition 9, there is a contraction in the step length along each direction at least once per η iterations. Thus,

$$\lim_{k \to \infty} \max_{1 \leq i \leq p_k} \left\{ \Delta_k^{(i)} \right\} = 0.$$

So, necessarily, $\min\{\Gamma_k^{(i)}\} \to +\infty$. This contradicts our original assumption. Hence, the claim. □

Both Lemma 5.3 and Lemma 5.5 lead to the following general result regarding the step lengths. Additional notation is required for this proof. Define

$$\tilde{\Gamma}_k^{(i)} = \Gamma_k^{(i)} - \Gamma_{\min}.$$

This quantity is equal to the number of contractions required to go from $\Delta_{\min}$ to $\Delta_k^{(i)}$.

Theorem 5.6. *Consider the optimization problem* (1.1) *or* (1.2), *satisfying Assumptions* 1 *and* 2. *Let the APPS algorithm in Figure* 3.1 *satisfy Conditions either* 3 *or* 4, 8, *and* 9, *Then, there exists a set* $\mathcal{K} \subset \{1, 2, \ldots\}$ *such that*

$$\lim_{k \in \mathcal{K}} \left\{ \max_{1 \leq i \leq p_k} \Delta_k^{(i)} \right\} = 0.$$

Proof. By either Lemma 5.3 (using Assumption 2 and Conditions 4, 8, and 9) or Lemma 5.5 (using Assumption 1 and Conditions 3 and 9), we have that there exists an index j and set $\mathcal{K}$ such that

$$\lim_{k \in \mathcal{K}} \Gamma_k^{(j)} = +\infty.$$

Without loss of generality, assume that

$$\Gamma_k^{(j)} > \eta \left(\Gamma_{\min} + 1 \right) \text{ for all } k \in \mathcal{K}, \tag{5.16}$$

where η is as defined in Condition 9.

Then if $k \in \mathcal{K}$, by (5.16), $\tilde{\Gamma}_k^{(i)} > 0$ and there has not been a success for at least $\tilde{\Gamma}_k^{(j)}$ iterations. On the other hand, by (5.16), $\lfloor \tilde{\Gamma}_k^{(j)} / \eta \rfloor > 0$ and there has been at least $\lfloor \tilde{\Gamma}_k^{(j)} / \eta \rfloor$ contractions in all other directions. Thus,

$$\tilde{\Gamma}_k^{(i)} \geq \lfloor \tilde{\Gamma}_k^{(j)} / \eta \rfloor \text{ for } k \in \mathcal{K}, 1, \leq i \leq p_k, i \neq j.$$

Thus,

$$\lim_{k \in \mathcal{K}} \left\{ \min_{1 \leq i \leq p_k} \Gamma_k^{(i)} \right\} = +\infty.$$

Hence, the claim. □

5.3. Convergence results. Using the machinery built in sections 5.1 and 5.2, results following Theorems 2.1 and 2.2 are immediate.

Theorem 5.7. *Consider the optimization problem* (1.1), *satisfying Assumptions* 1–4. *Let the APPS algorithm in Figure* 3.1 *satisfy Conditions* 1, 2, *either* 3 *or* 4, 7, 8, 9. *Then*

$$\liminf_{k \to +\infty} \| \nabla f(x_k) \| = 0.$$

Theorem 5.8. *Consider the optimization problem* (1.2), *satisfying Assumptions* 1–4. *Let the APPS algorithm in Figure* 3.1 *satisfy Conditions either* 3 *or* 4, 5, 7, 8, 9. *Then*

$$\liminf_{k \to +\infty} \chi(x_k) = 0.$$

6. Numerical results. As compared to (synchronous) PPS, the advantage of APPS is more efficient use of computation resources. Using PVM, the original APPS was shown to be faster than PPS on several different examples [12]. In later experiments on a small collection of test problems, "single-agent" versions of APPS (very similar to the new APPS proposed here) using MPI and PVM were shown to be overall faster than the original version of APPS using PVM [11].

Table 6.1
Comparison of PPS and APPS: average results over parallel multiple runs.

Problem	Method	Final F	Time (s)	% Idle	V1	V2	Evals	Cache
UNC5	APPS	1.24×10^5	63.9	1.11	10	74	274	36
	PPS	1.24×10^5	101.2	29.22	9	78	273	39
UNC6	APPS	1.24×10^5	73.1	0.85	86	66	306	36
	PPS	1.24×10^5	106.1	17.64	135	69	345	39
CON5	APPS	1.39×10^5	72.5	1.05	11	1	342	39
	PPS	1.39×10^5	106.8	28.24	7	3	325	37
CON6	APPS	1.39×10^5	97.1	0.82	129	2	435	43
	PPS	1.39×10^5	106.0	17.61	111	3	355	32
HC	APPS	2.38×10^4	207.1	1.65	2	80	152	24
	PPS	2.38×10^4	242.3	20.54	2	77	128	20

In this section, we present comparisons of the new version of APPS with PPS, both using MPI. We use APPSPACK 4.0 [8, 13] for our comparisons because it implements both PPS and APPS. The APPS implementation is identical to what is shown in Figure 3.1, and the PPS implementation is the same as APPS except that, in Step 2, we wait for all evaluations to be completed so that $\mathbf{Y}_k = \mathbf{X}_k$. We used the default settings for all parameters except that "Scaling" was set equal to the upper bounds and "Step Tolerance" was set to 0.02. Full details of the implementation can be found in [8].

The methods are compared on a set of five proposed benchmark test problems in well-field design (UNC5, UNC6, CON5, CON6) and hydraulic capture (HC), as described in detail in [5]. The problems are based on the MODFLOW [24] simulator from the U.S. Geological Survey. Problems UNC5 and CON5 each have 10 variables, UNC6 and CON6 each have 18 variables, and HC has 12 variables. The problems have bound constraints and three nonlinear constraints. The nonlinear constraints were treated by using an extreme barrier approach that sets $f(x) = +\infty$ whenever a constraint is violated. The nonlinear constraints are separated into two categories: Category 1 is constraints that can be checked before the simulator is called and are inexpensive to check; Category 2 is constraints that cannot be checked until after the simulator has been called and are therefore expensive to check.

Table 6.1 compares APPS and PPS on the test problems. We ran each problem 10 times on 11 nodes of Sandia's Catalyst Linux cluster, using MPICH [9, 10]. The Final F column lists the optimal function value, Time (s) lists the average parallel run time in seconds, % Idle lists the average per worker idle time, V1 lists the average number of constraint violations for Category 1 constraints, V2 lists the average number of constraints violations for Category 2 constraints, Evals lists the number of successful (i.e., feasible) evaluations, and Cache lists the total number of evaluations (including infeasible) that were looked up in the cache. Note that the sum of columns V2 and Evals gives the total number of calls to the simulator. Because APPS is asynchronous, the results can vary from run to run. In particular, it can converge to different local minima. This happened twice for problem UNC6 and twice for problem CON5. The function values were fairly close, but the other numbers (e.g., run times) were better, so those results were removed from the averages. Otherwise, all the methods converged to the same final value. We also had to discard one APPS run from the HC results because it had a bogus time reported.

Compared to PPS, APPS reduced the run time up to 37%. This can be attributed to better load balancing, resulting is substantially less idle time per worker process. The speculative nature of APPS can lead to more work: for problem CON6, APPS had 22% more calls to the simulator and was only 8% faster. On the other hand, it sometimes results in less work: for problem UNC6, APPS had 10% fewer calls to the simulator and ran 31% faster.

7. Conclusions. We presented a new version of APPS based on a manager-worker paradigm. This algorithm encapsulates either simple or sufficient decrease as well as the ability to handle bound constraints. A nice feature of this version of APPS is that it closely mirrors PPS (at least as described here).

In fact, neither PPS nor APPS has been presented in its most general form. For example, these algorithms handle updating the step lengths in a particular way. At unsuccessful iterations, the contraction factor in Step 5 is hardwired to $\frac{1}{2}$; in fact, this could be any fixed value in the interval $(0, 1)$, with the additional requirement that the value be rational in the simple decrease case. Similarly, an expansion factor could be used on successful iterations in Step 4. In both cases, these terms could be adaptive (i.e., different at each iteration). We also assume that the search directions are fixed between successful iterations. This is not required for PPS; however, we have presented it that way because it makes the description of APPS more straightforward. Finally, it is possible to modify APPS so that the search directions are allowed to change at even unsuccessful iterations provided that adequate controls are in place for globalization.

Likewise, some of the assumptions and conditions employed in the convergence analysis can be relaxed. We need not assume that the gradient is Lipschitz (Assumption 4); instead, continuous differentiability is sufficient. Part (d) in Condition 3 can be changed to say that either ρ is identically zero or it satisfies Condition 4; in other words, the argument based on lattice structure is independent of the decrease condition. Part (e) in Condition 3 can be generalized to say that the pseudo step can be anything of the form $2^{-\Gamma}\Delta_0$ for $\Gamma \in \mathbb{Z}$. Part (b) in Condition 4 is more restrictive than necessary for PPS (which only needs that $\rho(t)$ monotonically decreases), but this more restrictive assumption is needed by APPS. Condition 5 can be weakened, but the resulting condition is much more complex (see Condition 1 in [15]).

The convergence theory borrows heavily from the analysis of GSS in [14, 15] as well as the analysis of the original APPS [17]. The convergence results presented in section 5 are *weak* convergence results because it is possible that only a subsequence of the iterates will converge to a stationary point. Although strong convergence results are possible in the synchronous case [14], it is unclear whether such assurances can be made for the asynchronous algorithm because strong convergence requires that the algorithm always take the best direction at each iteration. Local convergence results exist for PPS [14] but are left as a topic for future study for APPS.

Just like the original version of APPS, the new version of APPS demonstrates faster execution times than its synchronous counterpart. More information on the test problems and a comparison of APPS with other derivative-free optimization methods can be found in [5]. We used APPSPACK 4.0 [13] for testing, full details of which are provided in [8].

We close by pointing toward the future. As we have already stated, one objective of the redesign of APPS is to enable easier incorporation of methods for handling linear constraints. In that case, the search directions must conform to the nearby boundary [20, 15], so the ability to change the search directions in Step 4 makes this relatively simple. This will be pursued in future research.

Appendix A. Evolution of peer-to-peer to manager-worker. The switch from the peer-to-peer version [12] to manager-worker was gradual and largely the result of user requests. As mentioned in the introduction, peer-to-peer APPS is based on the concept of what are called agents. Each agent handles a single direction (and up to one corresponding trial point) and launches its own workers to execute the function evaluation. Thus, there is one agent per search direction and the number of search directions is necessarily fixed. The working assumption is that there is one direction per processor and one processor per machine.

The first step in the evolution to the manager-worker design is motivated by multiprocessor (i.e., SMP) machines. On a cluster of machines that each have, say, four processors, it is more efficient to have one agent (as opposed to four) for every four search directions. The peer-to-peer design remained intact, but a single agent could handle multiple search directions. The directions sharing a common agent also shared one common best point. In fact, this is equivalent to the original peer-to-peer model with instantaneous communication between appropriate subsets of the agents. Once agents were designed and implemented to handle multiple directions, having one agent handle all directions was trivial.

The difference between this first manager-worker concept and the algorithm described here is the handling of the search directions. Having a fixed set of search directions is fairly crucial to the peer-to-peer design. In particular, it is implemented so that there is at most one function evaluation per search direction at any given time. The disconnected points described in section 4 cannot exist. Although it would certainly be possible to design a peer-to-peer APPS that allows the search directions to change as the optimization progresses, it is much simpler to do this in a manager-worker context.

Acknowledgments. For many of the ideas and concepts that have appeared in this paper, I am pleased to acknowledge my collaborators on APPS and GSS: Virginia Torczon, Robert Michael Lewis, Patricia Hough, and Genetha Gray. I am indebted to C. T. Kelley, Kathleen Fowler, and the rest of the “shootout” team for putting together the real-life benchmark problems that are used for the numerical experiements, and to Genetha Gray for helping to make them work with our software. I also thank my colleagues and the reviewers for reading earlier versions of this manuscript and offering suggestions that have greatly improved the presentation.

REFERENCES

[1] C. Audet and J. E. Dennis, Jr., *Analysis of generalized pattern searches*, SIAM J. Optim., 13 (2003), pp. 889–903.

[2] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*, MPS-SIAM Ser. Optim. 1, SIAM, Philadelphia, 2000.

[3] J. E. Dennis and V. Torczon, *Managing approximation models in optimization*, in Multidisciplinary Design Optimization: State of the Art, N. M. Alexandrov and M. Y. Hussaini, eds., SIAM, Philadelphia, 1997, pp. 330–347.

[4] G. E. Fagg and J. Dongarra, *FT–MPI: Fault tolerant mpi, supporting dynamic applications in a dynamic world*, in Proceedings of the 7th European PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface, Lecture Notes in Comput. Sci., Springer-Verlag, New York, 2000, pp. 346–353.

[5] K. R. Fowler, J. P. Reese, C. E. Kees, J. E. Dennis, C. T. Kelley, C. T. Miller, C. Audet, A. J. Booker, G. Couture, R. W. Darwin, M. W. Farthing, D. E. Finkel, J. M. Gablonsky, G. Gray, and T. G. Kolda, *A comparison of optimization methods for problems involving flow and transport phenomena in saturated subsurface systems*, in preparation.

[6] U. M. García-Palomares and J. F. Rodríguez, *New sequential and parallel derivative-free algorithms for unconstrained minimization*, SIAM J. Optim., 13 (2002), pp. 79–96.

[7] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. S. Sunderam, *PVM: Parallel Virtual Machine: A Users' Guide and Tutorial for Network Parallel Computing*, MIT Press, Cambridge, MA, 1994.

[8] G. A. Gray and T. G. Kolda, *APPSPACK* 4.0*: Asynchronous Parallel Pattern Search for Derivative-Free Optimization*, Tech. Report SAND2004-6391, Sandia National Laboratories, Livermore, CA, 2004, ACM Trans. Math. Software.

[9] W. Gropp, E. Lusk, N. Doss, and A. Skjellum, *A high-performance, portable implementation of the MPI message passing interface standard*, Parallel Comp., 22 (1996), pp. 789–828.

[10] W. D. Gropp and E. Lusk, *User's Guide for* `mpich`*, a Portable Implementation of MPI*, Tech. Report ANL-96/6, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1996.

[11] P. D. Hough, T. G. Kolda, and H. A. Patrick, *Usage Manual for APPSPACK Version* 2.0, Tech. Report SAND2000-8843, Sandia National Laboratories, Livermore, CA, 2001.

[12] P. D. Hough, T. G. Kolda, and V. J. Torczon, *Asynchronous parallel pattern search for nonlinear optimization*, SIAM J. Sci. Comput., 23 (2001), pp. 134–156.

[13] T. G. Kolda et al., *Appspack Version* 4.0. http://software.sandia.gov/appspack/ (2004).

[14] T. G. Kolda, R. M. Lewis, and V. Torczon, *Optimization by direct search: New perspectives on some classical and modern methods*, SIAM Rev., 45 (2003), pp. 385–482.

[15] T. G. Kolda, R. M. Lewis, and V. Torczon, *Stationarity results for generating set search for linearly constrained optimization*, Tech. Report SAND2003–8550, Sandia National Laboratories, Livermore, CA, 2003. Submitted to SIAM Journal on Optimization.

[16] T. G. Kolda and V. J. Torczon, *Understanding asynchronous parallel pattern search*, in High Performance Algorithms and Software for Nonlinear Optimization, G. Di Pillo and A. Murli, eds., Appl. Optim. 82, Kluwer Academic Publishers, Boston, 2003, pp. 316–335.

[17] T. G. Kolda and V. J. Torczon, *On the convergence of asynchronous parallel pattern search*, SIAM J. Optim., 14 (2004), pp. 929–964.

[18] R. M. Lewis and V. Torczon, *Rank Ordering and Positive Bases in Pattern Search Algorithms*, Tech. Report 96–71, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA, 1996.

[19] R. M. Lewis and V. Torczon, *Pattern search algorithms for bound constrained minimization*, SIAM J. Optim., 9 (1999), pp. 1082–1099.

[20] R. M. Lewis and V. Torczon, *Pattern search methods for linearly constrained minimization*, SIAM J. Optim., 10 (2000), pp. 917–941.

[21] R. M. Lewis, V. Torczon, and M. W. Trosset, *Why pattern search works*, Optima, 59 (1998), pp. 1–7.

[22] S. Lucidi and M. Sciandrone, *A derivative-free algorithm for bound constrained optimization*, Comput. Optim. Appl., 21 (2002), pp. 119–142.

[23] S. Lucidi and M. Sciandrone, *On the global convergence of derivative-free methods for unconstrained optimization*, SIAM J. Optim., 13 (2002), pp. 97–116.

[24] M. G. McDonald and A. W. Harbaugh, *Modular three-dimensional finite-difference groundwater flow model*, available from Books and Open File Report Section, USGS Box 25425, Denver, CO 80225. Techniques of Water-Resources Investigations, Book 6: Modeling Techniques, chapter A1, 1988.

[25] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra, *MPI: The Complete Reference, Volume* 1*, The MPI Core, 2nd ed.*, MIT Press, Cambridge, MA, 1998.

[26] V. Torczon, *On the convergence of pattern search algorithms*, SIAM J. Optim., 7 (1997), pp. 1–25.

SIAM J. Optim.
Vol. 16, No. 2, pp. 587–609

A TWO-SIDED RELAXATION SCHEME FOR MATHEMATICAL PROGRAMS WITH EQUILIBRIUM CONSTRAINTS[*]

VICTOR DEMIGUEL[†], MICHAEL P. FRIEDLANDER[‡], FRANCISCO J. NOGALES[§], AND STEFAN SCHOLTES[¶]

Abstract. We propose a relaxation scheme for mathematical programs with equilibrium constraints (MPECs). In contrast to previous approaches, our relaxation is two-sided: both the complementarity and the nonnegativity constraints are relaxed. The proposed relaxation update rule guarantees (under certain conditions) that the sequence of relaxed subproblems will maintain a strictly feasible interior—even in the limit. We show how the relaxation scheme can be used in combination with a standard interior-point method to achieve superlinear convergence. Numerical results on the MacMPEC test problem set demonstrate the fast local convergence properties of the approach.

Key words. nonlinear programming, mathematical programs with equilibrium constraints, complementarity constraints, constrained minimization, interior-point methods, primal-dual methods, barrier methods

AMS subject classifications. 49M37, 65K05, 90C30

DOI. 10.1137/04060754x

1. Introduction.

Consider the generic mathematical program with equilibrium constraints (MPEC), expressed as

$$\boxed{\begin{array}{lll} \text{(MPEC)} \qquad\qquad & \underset{x}{\text{minimize}} & f(x) \\ & \text{subject to} & \qquad c(x) = 0, \\ & & \min(x_1, x_2) = 0, \\ & & \qquad\quad x_0 \ge 0, \end{array}}$$

where $x = (x_0, x_1, x_2) \in \mathrm{R}^{p\times n\times n}$, $f : \mathrm{R}^{p+2n} \to \mathrm{R}$ is the objective function, and $c : \mathrm{R}^{p+2n} \to \mathrm{R}^m$ is a vector of constraint functions. The complementarity constraint $\min(x_1, x_2) = 0$ requires that either $[x_1]_j$ or $[x_2]_j$ vanishes for each component $j = 1, \ldots, n$ and that the vectors x_1 and x_2 are nonnegative. See the survey paper by Ferris and Pang [5] for examples of complementarity models and the monographs by Luo, Pang, and Ralph [13] and Outrata, Kocvara, and Zowe [17] for details on MPEC theory and applications.

MPECs can be reformulated as standard nonlinear programs (NLPs) by replacing the nonsmooth complementarity constraint by a set of equivalent smooth constraints:

$$\min(x_1, x_2) = 0 \qquad \Longleftrightarrow \qquad X_1 x_2 = 0, \quad x_1, x_2 \ge 0,$$

[*] Received by the editors April 29, 2004; accepted for publication (in revised form) July 21, 2005; published electronically December 30, 2005.
http://www.siam.org/journals/siopt/16-2/60754.html

[†] London Business School, Regent's Park, London NW1 4SA, UK (avmiguel@london.edu).

[‡] Department of Computer Science, University of British Columbia, Vancouver, BC V6T 1Z4, Canada (mpf@cs.ubc.ca). The work of this author was partially supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy contract W-31-109-Eng-38.

[§] Department of Statistics, Universidad Carlos III de Madrid, Avda. de la Universidad 30, 28911 Leganes, Madrid, Spain (FcoJavier.Nogales@uc3m.es).

[¶] Judge Institute of Management, University of Cambridge, Cambridge CB2 1AG, UK (s.scholtes@jims.cam.ac.uk).

where $X_1 = \operatorname{diag}(x_1)$. However, these constraints do not admit a strictly feasible point, which implies that both the linear independence and the weaker Mangasarian–Fromovitz constraint qualifications are violated at every feasible point. These conditions are key ingredients for standard convergence analyses of NLP methods.

We propose a strategy that forms a sequence of NLP approximations to the MPEC, each with a feasible set that has a strict interior and that will typically satisfy a constraint qualification. In contrast to previous approaches, the relaxation is two-sided: both the complementarity ($X_1 x_2 = 0$) and the nonnegativity ($x_1, x_2 \geq 0$) constraints are relaxed. The proposed relaxation update rules guarantee (under certain conditions) that the sequence of relaxed subproblems will maintain a strictly feasible interior—even in the limit. Consequently, a standard interior method may be applied to the relaxed subproblem, as we show in section 4. The relaxation scheme could, in principle, be used in combination with other Newton-type methods, such as sequential quadratic programming or linearly constrained Lagrangian [8] methods.

1.1. Other work on MPECs.

The direct application of off-the-shelf nonlinear optimization codes to MPECs was long neglected following early reports of their poor performance. See, for example, Luo, Pang, and Ralph [13] and, more recently, Anitescu [1], who describes the poor performance of the popular MINOS [15] code on MacMPEC [11] test problems. Interest in the application of standard NLP methods to MPECs has been revitalized for two reasons, however. First, it is now clear that the approach makes sense because strong stationarity implies the existence of standard NLP multipliers for MPECs in their NLP form, albeit an unbounded set (see Fletcher et al. [6]). Second, Fletcher and Leyffer [7] report promising numerical results for sequential quadratic programming (SQP) codes. These favorable numerical results are complemented by the local convergence analysis in [6].

Considerable effort has gone into the specialization of standard nonlinear programming methods in answer to the attendant difficulties of reformulating MPECs as NLPs. The approaches can be roughly divided into two categories: penalization and relaxation strategies. Such a categorization may be viewed as synthetic, however, because both approaches share the same philosophy: to relax the troublesome complementarity constraints. The difference is evident in the methodology.

The first analyses of penalization approaches can be found in Scholtes and Stöhr [21] and in Anitescu [1]. The strategy is to eliminate the explicit complementarity constraints $X_1 x_2 = 0$ by adding an exact penalty function to the objective to account for complementarity violation. The structural ill-posedness is thereby removed from the constraints. Anitescu gives conditions under which SQP methods with an elastic mode, such as SNOPT, will converge locally at a fast rate when applied to MPECs. Hu and Ralph [9] analyze the global convergence of the penalization method with exact solves. Anitescu [2] gives global convergence results with inexact solves. The penalization approach has been applied within the interior-point context by Benson, Shanno, and Vanderbei [4] and Leyffer, Lopez-Calva, and Nocedal [10]. Both papers report very good numerical results. Leyffer, Lopez-Calva, and Nocedal also give a comprehensive global and local convergence analysis of the penalization approach within an interior-point framework.

The relaxation approach (sometimes called regularization) keeps the complementarity constraints explicit but relaxes them to $X_1 x_2 \leq \delta_k$, where δ_k is a positive vector that is driven to zero. This scheme replaces the MPEC by a sequence of relaxed subproblems whose strictly feasible region is nonempty. The approach was extensively analyzed by Scholtes [22]. We call this a one-sided relaxation scheme to contrast it against our approach. The one-sided relaxation strategy has been adopted by Liu and Sun [12] and Raghunathan and Biegler [18]. Liu and Sun propose an interior method that solves each of the relaxed subproblems to within a prescribed tolerance. On the other hand, the method of Raghunathan and Biegler takes only one iteration of an interior method on each of the relaxed subproblems. A difficulty associated with both methods is that the strictly feasible regions of the relaxed problems become empty in the limit, and this may lead to numerical difficulties. Raghunathan and Biegler address this difficulty by using a modified search direction that ensures that their algorithm converges locally at a quadratic rate.

The relaxation scheme that we propose (described in section 3) does not force the strictly feasible regions of the relaxed MPECs to become empty in the limit. As a result, one can apply a standard interior method to the relaxed problems without having to modify the search direction, as in [18]. But like [18], our algorithm (described in section 4) performs only one interior iteration per relaxed problem. We show in section 4.2 that it converges locally at a superlinear rate, and in section 5 we discuss some implementation issues. We illustrate in section 6 the performance of the algorithm on a subset of the MacMPEC test problems. The numerical results seem to reflect our local convergence analysis and give evidence to the algorithm's effectiveness in practice.

1.2. Definitions.

Unless otherwise specified, the function $\|x\|$ represents the Euclidean norm of a vector x. With vector arguments, the functions $\min(\cdot,\cdot)$ and $\max(\cdot,\cdot)$ apply componentwise to each element of the arguments. Denote by $[\,\cdot\,]_i$ the ith component of a vector. The uppercase variables X, S, V, and Z denote diagonal matrices formed from the elements of the vectors x, s, v, and z, respectively. Let $g(x)$ denote the gradient of the objective function $f(x)$. Let $A(x)$ denote the Jacobian of $c(x)$, a matrix whose ith row is the gradient of $[c(x)]_i$. Let $H_i(x)$ denote the Hessian of $[c(x)]_i$.

We make frequent use of standard definitions for linear independence constraint qualification (LICQ) and strict complementary slackness (SCS), and the second order sufficiency condition (SOSC). These definitions can be found in [16, Ch. 12].

2. Optimality conditions for MPECs.

The standard KKT theory of nonlinear optimization is not directly applicable to MPECs because standard constraint qualifications do not hold. There is a simple way around this problem, however, as observed by Scheel and Scholtes [20]. At every feasible point of the MPEC one can define the relaxed NLP, which is typically well behaved in nonlinear programming terms. It is shown in [20] that the KKT conditions of the relaxed NLP are necessary optimality conditions for (MPEC), provided that the relaxed NLP satisfies LICQ.

2.1. First-order conditions and constraint qualification.

Let $\bar{x}$ be feasible with respect to (MPEC). The relaxed NLP at $\bar{x}$ is defined as

$$\boxed{\begin{array}{llllll}
(\mathrm{RNLP}_{\bar{x}}) & \underset{x}{\text{minimize}} & f(x) & & & \\
 & \text{subject to} & c(x) = 0, & x_0 \ge 0 & & \\
 & & [x_1]_j = 0, & [x_2]_j \ge 0 & \text{for all } j \text{ such that } [\bar{x}_1]_j = 0 < [\bar{x}_2]_j, \\
 & & [x_1]_j \ge 0, & [x_2]_j = 0 & \text{for all } j \text{ such that } [\bar{x}_1]_j > 0 = [\bar{x}_2]_j, \\
 & & [x_1]_j \ge 0, & [x_2]_j \ge 0 & \text{for all } j \text{ such that } [\bar{x}_1]_j = 0 = [\bar{x}_2]_j.
\end{array}}$$

The feasible region defined by the bound constraints of $(\mathrm{RNLP}_{\bar{x}})$ is larger than that defined by the equilibrium constraints. Hence the term *relaxed* NLP. Most important, the problematic equilibrium constraints of (MPEC) have been substituted by a better-posed system of equality and inequality constraints.

Define

$$\mathcal{L}(x,y) = f(x) - y^T c(x)$$

as the Lagrangian function of RNLP$_{\bar{x}}$.

Despite a possibly larger feasible set, it can be shown that if LICQ holds for (RNLP$_{x^*}$), its KKT conditions are also necessary optimality conditions for (MPEC) [20]. This observation leads to the following stationarity concept for MPECs.

Definition 2.1. *A point (x^*, y^*, z^*) is* strongly stationary *for (MPEC) if it satisfies the KKT conditions for (RNLP$_{x^*}$):*

$$\nabla_x \mathcal{L}(x^*, y^*) = z^*, \tag{2.1a}$$

$$c(x^*) = 0, \tag{2.1b}$$

$$\min(x_0^*, z_0^*) = 0, \tag{2.1c}$$

$$\min(x_1^*, x_2^*) = 0, \tag{2.1d}$$

$$[x_1^*]_j [z_1^*]_j = 0, \tag{2.1e}$$

$$[x_2^*]_j [z_2^*]_j = 0, \tag{2.1f}$$

$$[z_1^*]_j, [z_2^*]_j \geq 0, \quad \textit{if} \quad [x_1^*]_j = [x_2^*]_j = 0. \tag{2.1g}$$

With the relaxed NLP we can define a constraint qualification for MPECs analogous to LICQ and deduce a necessary optimality condition for MPECs.

Definition 2.2. *The point x^* satisfies* MPEC linear independence constraint qualification *(MPEC-LICQ) for (MPEC) if it is feasible and if LICQ holds at x^* for (RNLP$_{x^*}$).*

Proposition 2.3 (see, for example, Scheel and Scholtes [20]). *If x^* is a local minimizer for (MPEC) at which MPEC-LICQ holds, then there exist unique multipliers y^* and z^* such that (x^*, y^*, z^*) is strongly stationary.*

2.2. Strict complementarity and second-order sufficiency. Through the relaxed NLP we can define strict complementarity and second-order conditions for MPECs. These play a crucial role in the development and analysis of the relaxation scheme proposed in this paper.

We define two different strict complementary slackness conditions for MPECs. The first of the two is stronger and is the one assumed in [22, Theorem 4.1]. It requires all multipliers z_0, z_1, and z_2 to be strictly complementary with respect to their associated primal variables. In our analysis, we only assume the second, less restrictive condition, which only requires strict complementarity of z_0.

Definition 2.4. *The triple (x^*, y^*, z^*) satisfies the* MPEC strict complementary slackness *(MPEC-SCS) condition for (MPEC) if it is strongly stationary, if $\max(x_0^*, z_0^*) > 0$, and if $[x_i^*]_j + [z_i^*]_j \neq 0$ for each $i = 1, 2$ and $j = 1, \ldots, n$.*

Definition 2.5. *The triple (x^*, y^*, z^*) satisfies* MPEC weak strict complementary slackness *(MPEC-WSCS) for (MPEC) if it is strongly stationary and if $\max(x_0^*, z_0^*) > 0$.*

We define two second-order sufficient conditions for MPECs: MPEC-SOSC and MPEC-SSOSC. The first condition is equivalent to the RNLP-SOSC defined by Ralph and Wright [19, Definition 2.7]. The second condition is stronger than the RNLP-SSOSC defined in [19, Definition 2.8].

The tangent cone of the feasible set of (RNLP$_{x^*}$) is given by

$$\begin{aligned}\mathcal{T} = &\{\alpha p \mid \alpha > 0,\ p \in \mathrm{R}^n\}\\ &\cap \{p \mid A(x^*)p = 0\}\\ &\cap \{p \mid [p_0]_j \geq 0 \text{ for all } j \text{ such that } [x_0^*]_j = 0\}.\end{aligned}$$

The second-order sufficient condition for optimality depends on positive curvature of the Lagrangian in a subspace, i.e.,

$$p^T\nabla^2_{xx}\mathcal{L}(x^*, y^*)p > 0, \quad p \neq 0, \tag{2.2}$$

for all p in some subset of the feasible directions $\mathcal{T}$.

Definition 2.6. *The triple* (x^*, y^*, z^*) *satisfies the* MPEC second-order sufficiency condition *(MPEC-SOSC) for (MPEC) if it is strongly stationary and if* (2.2) *holds for all nonzero* $p \in \overline{\mathcal{F}}$*, where*

$$\begin{aligned}\overline{\mathcal{F}} \overset{\text{def}}{=} \{p \in \mathcal{T} \mid [p_0]_j = 0 &\quad \text{for all } j \text{ such that } [x_0^*]_j = 0 \text{ (and } [z_0^*]_j > 0),\\ [p_0]_j \geq 0 &\quad \text{for all } j \text{ such that } [x_0^*]_j = 0 \text{ (and } [z_0^*]_j = 0),\\ [p_i]_j = 0 &\quad \text{for all } j \text{ such that } [x_i^*]_j = 0 \text{ (and } [z_i^*]_j \neq 0),\ i = 1, 2,\\ [p_i]_j \geq 0 &\quad \text{for all } j \text{ such that } [x_i^*]_j = 0 \text{ (and } [z_i^*]_j = 0),\ i = 1, 2,\\ [p_i]_j = 0 &\quad \text{for all } j \text{ such that } [x_i^*]_j = 0 < [x_\ell^*]_j,\ i, \ell = 1, 2,\ i \neq \ell\}.\end{aligned}$$

If the last two conditions in the definition of $\overline{\mathcal{F}}$ are dropped, we obtain a stronger second-order condition, which is equivalent to the one assumed in [22, Theorem 4.1].

Definition 2.7. *The triple* (x^*, y^*, z^*) *satisfies* MPEC strong second-order sufficiency condition *(MPEC-SSOSC) for (MPEC) if it is strongly stationary and if* (2.2) *holds for all nonzero* $p \in \mathcal{F}$*, where*

$$\begin{aligned}\mathcal{F} \overset{\text{def}}{=} \{p \in \mathcal{T} \mid [p_0]_j = 0 &\quad \text{for all } j \text{ such that } [x_0^*]_j = 0 \text{ (and } [z_0^*]_j > 0),\\ [p_0]_j \geq 0 &\quad \text{for all } j \text{ such that } [x_0^*]_j = 0 \text{ (and } [z_0^*]_j = 0),\\ [p_i]_j = 0 &\quad \text{for all } j \text{ such that } [x_i^*]_j = 0 \text{ (and } [z_i^*]_j \neq 0),\ i = 1, 2\}.\end{aligned}$$

Note that MPEC-SSOSC ensures that the Hessian of the Lagrangian has positive curvature in the range space of all nonnegativity constraints $(x_1, x_2 \geq 0)$ whose multipliers are zero. Note also that MPEC-SOSC and -SSOSC are equivalent when MPEC-SCS holds.

In our analysis, we assume MPEC-WSCS and -SSOSC. However, we note that our results are also valid under MPEC-SCS and -SOSC. To see this, simply note that MPEC-SCS implies MPEC-WSCS and that MPEC-SOSC and -SSOSC are equivalent when MPEC-SCS holds. Thus our analysis holds either under MPEC-SCS and -SOSC, or under a weaker SCS (at the expense of assuming a stronger SOSC).

Raghunathan and Biegler [18] make a strict complementarity assumption that is more restrictive than MPEC-WSCS but less restrictive than MPEC-SCS. In particular, they require $\max(x_0^*, z_0^*) > 0$ and $[z_i^*]_j \neq 0$ for all j such that $[\bar{x}_1]_j = [\bar{x}_2]_j = 0$. This condition is termed upper-level SCS (MPEC-USCS) in [19]. The strength of the second-order condition they assume is also between that of MPEC-SOSC and MPEC-SSOSC. In particular, their second-order condition is obtained from the MPEC-SOSC by dropping the last condition in the definition of $\overline{\mathcal{F}}$.

3. A strictly feasible relaxation scheme.

In this section we propose a relaxation scheme for which the strictly feasible region of the relaxed problems may remain nonempty even in the limit.

A standard relaxation of the complementarity constraint proceeds as follows. The complementarity constraint $\min(x_1, x_2) = 0$ is first reformulated as the system of inequalities $X_1 x_2 \leq 0$ and $x_1, x_2 \geq 0$. A vector $\delta_c \in \mathrm{R}^n$ of strictly positive parameters relaxes the complementarity constraint to arrive at

$$X_1 x_2 \leq \delta_c, \quad x_1, x_2 \geq 0. \tag{3.1}$$

The original complementarity constraint is recovered when $\delta_c = 0$. Note that at all points feasible for (MPEC) the gradients of the active constraints in (3.1) are linearly independent when $\delta_c > 0$. Moreover, the strictly feasible region of the relaxed constraints (3.1) is nonempty when $\delta_c > 0$. Unfortunately, the strictly feasible region of the relaxed MPEC becomes empty as the components of δ_c tend to zero.

3.1. A two-sided relaxation.

In contrast to (3.1), our proposed scheme additionally relaxes each component of the bounds $x_1, x_2 \geq 0$ by the amounts $[\delta_1]_j > 0$ and $[\delta_2]_j > 0$ so that the relaxed complementarity constraints become

$$X_1 x_2 \leq \delta_c, \quad x_1 \geq -\delta_1, \quad x_2 \geq -\delta_2, \tag{3.2}$$

where $\delta_c, \delta_1, \delta_2 \in \mathrm{R}^n$ are vectors of strictly positive relaxation parameters. Note that for any relaxation parameter vectors $(\delta_1, \delta_2, \delta_c)$ that satisfy $\max(\delta_c, \delta_1) > 0$ and $\max(\delta_c, \delta_2) > 0$, the strictly feasible region of (3.2) is nonempty, and the active constraint gradients are linearly independent.

The main advantage of the strictly feasible relaxation scheme (3.2) is that there is no need to drive both relaxation parameters to zero to recover a stationary point of the MPEC. As we show in Theorem 3.1, for any strongly stationary point of (MPEC) that satisfies MPEC-LICQ, -WSCS, and -SSOSC, there exist relaxation parameter vectors $(\delta_1^*, \delta_2^*, \delta_c^*)$ satisfying $\max(\delta_c^*, \delta_1^*) > 0$ and $\max(\delta_c^*, \delta_2^*) > 0$ such that the relaxed MPEC satisfies LICQ, SCS, and SOSC.

3.2. An example.

The intuition for the relaxation scheme proposed in section 3.1 is best appreciated with an example. Consider the MPEC [22]

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \frac{1}{2}\left[(x_1 - a)^2 + (x_2 - b)^2\right] \\ \text{subject to} & \min(x_1, x_2) = 0 \end{array} \tag{3.3}$$

and the associated relaxed MPEC derived by applying the relaxation (3.2) to (3.3):

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \frac{1}{2}\left[(x_1 - a)^2 + (x_2 - b)^2\right] \\ \text{subject to} & \quad x_1 \geq -\delta_1, \\ & \quad x_2 \geq -\delta_2, \\ & X_1 x_2 \leq \delta_c. \end{array} \tag{3.4}$$

For any choice of parameters $a, b > 0$, (3.3) has two local minimizers: $(a, 0)$ and $(0, b)$. Each is strongly stationary and satisfies MPEC-LICQ, -SCS, and -SOSC and thus they also satisfy MPEC -LICQ, -WSCS, and -SSOSC. Evidently, these local minimizers are also minimizers of (3.4) for $\delta_c = 0$ and for any $\delta_1, \delta_2 > 0$. If the data are changed so that $a > 0$ and $b < 0$, then the point $(a, 0)$ is a unique minimizer of (3.3), and also a unique minimizer of (3.4) for any $\delta_c > 0$ and for $\delta_1 = \delta_2 = 0$. Moreover, if $a, b < 0$, then $(0, 0)$ is the unique minimizer of (3.3) and also a unique minimizer of (3.4) for any $\delta_c > 0$ and for $\delta_1 = \delta_2 = 0$. Thus there is no need to drive both δ_c and δ_1, δ_2 to zero to recover a stationary point of (MPEC).

A key property of MPECs that we exploit is the fact that the MPEC multipliers provide information about which relaxation parameters need to be driven to zero. To illustrate this, let us suppose $a, b > 0$ and consider the local minimizer $(a, 0)$ of the MPEC. In this simple example the minimizer of the relaxed problem will lie on the curve $X_1 x_2 = \delta_c$ for all sufficiently small δ_c. The MPEC solution will be recovered if we drive δ_c to zero. The values of the other parameters δ_1, δ_2 have no impact as long as they remain positive; the corresponding constraints will remain inactive. Note that this situation occurs precisely if the MPEC multiplier of the active constraint, here $x_2 \geq 0$, is negative, that is, the gradient of the objective function points outside of the positive orthant. If this situation is observed algorithmically, we will reduce δ_c and keep δ_1, δ_2 positive. A similar argument can be made if the gradient points in the interior of the positive orthant, in which case δ_1 or δ_2 need to be driven to zero to recover the MPEC minimizer. The parameter δ_c, however, must remain positive to maintain the strict interior of the feasible set.

The foregoing cases correspond to nondegenerate solutions; that is, there are no biactive constraints. Biactivity occurs in the example if $a, b < 0$. In this case the minimizer is the origin, and both MPEC multipliers are positive. Hence, we need to drive δ_1, δ_2 to zero and keep δ_c positive to avoid a collapsing strictly feasible region.

To see how one can recover an MPEC minimizer that satisfies MPEC-WSCS and -SSOSC, consider the example with $a = 0$ and $b = 1$. In this case $(0, 1)$ is a minimizer satisfying MPEC-WSCS and -SSOSC. To recover this minimizer from the relaxed MPEC (3.4) we do not need to drive any of the three relaxation parameters to zero. In particular, it is easy to see that $(0, 1)$ is a minimizer to the relaxed problem satisfying LICQ, SCS, and SOSC for any $\delta_1, \delta_2, \delta_c > 0$.

Our goal in the remainder of this paper is to turn this intuition into an algorithm and to analyze its convergence behavior for general MPECs.

3.3. The relaxed MPEC. In addition to introducing the relaxation parameter vectors $(\delta_1, \delta_2, \delta_c)$, we introduce slack variables $s \equiv (s_0, s_1, s_2, s_c)$ so that only equality and nonnegativity constraints on s are present. The resulting relaxed MPEC is

$$\boxed{\begin{array}{llrl} \text{(MPEC-}\delta\text{)} & \underset{x,s}{\text{minimize}} & f(x) & \\ & \text{subject to} & c(x) = 0 & : y, \\ & & s_0 - x_0 = 0 & : v_0, \\ & & s_1 - x_1 = \delta_1 & : v_1, \\ & & s_2 - x_2 = \delta_2 & : v_2, \\ & & s_c + X_1 x_2 = \delta_c & : v_c, \\ & & s \geq 0, & \end{array}}$$

where the dual variables y and $v \equiv (v_0, v_1, v_2, v_c)$ are shown next to their corresponding constraints. We note that the slack variable s_0 is not strictly necessary—the nonnegativity of x_0 could be enforced directly. However, such a device may be useful in practice because an initial value of x can be used without modification, and we need to choose starting values only for s, y, and v. Moreover, this notation greatly simplifies the following discussion.

To formulate the stationarity conditions for the relaxed MPEC, we group the set of equality constraints involving the slack variables s into a single expression by defining

$$h(x,s) = -\begin{bmatrix} s_0 - x_0 \\ s_1 - x_1 \\ s_2 - x_2 \\ s_c + X_1 x_2 \end{bmatrix} \qquad \text{and} \qquad \delta = \begin{bmatrix} 0 \\ \delta_1 \\ \delta_2 \\ \delta_c \end{bmatrix}. \tag{3.5}$$

The Jacobian of h with respect to the variables x is given by

$$B(x) \equiv \nabla_x h(x,s)^T = \begin{bmatrix} I & & \\ & I & \\ & & I \\ & -X_2 & -X_1 \end{bmatrix}. \tag{3.6}$$

The tuple (x^*, s^*, y^*, v^*) is a KKT point for (MPEC-δ) if it satisfies

$$\nabla_x \mathcal{L}(x,y) - B(x)^T v \equiv r_d = 0, \tag{3.7a}$$

$$\min(s,v) \equiv r_c = 0, \tag{3.7b}$$

$$c(x) \equiv r_f = 0, \tag{3.7c}$$

$$h(x,s) + \delta \equiv r_\delta = 0. \tag{3.7d}$$

Define the vector $w = (x, s, y, v)$ and the vector $r(w;\delta) = (r_d, r_c, r_f, r_\delta)$ as a function of w and δ. With this notation, w^* is a KKT point for (MPEC-δ) if $\|r(w^*;\delta)\| = 0$. The Jacobian of (3.7) is given by

$$K(w) \equiv \begin{bmatrix} H(x) & & -A(x)^T & -B(x)^T \\ & V & & S \\ A(x) & & & \\ B(x) & -I & & \end{bmatrix},$$

where

$$H(x) \equiv \nabla^2_{xx} \mathcal{L}(x,y) + \begin{bmatrix} 0 & & \\ & & V_c \\ & V_c & \end{bmatrix}.$$

3.4. Properties of the relaxed MPEC. Stationary points of (MPEC-δ) are closely related to those of (MPEC) for certain values of the relaxation parameters. The following theorem makes this relationship precise.

THEOREM 3.1. *Let (x^*, y^*, z^*) be a strongly stationary point of (MPEC), and let the vector δ^* satisfy*

$$[\delta_i^*]_j = 0 \quad \textit{if} \quad [z_i^*]_j > 0, \tag{3.8a}$$

$$[\delta_i^*]_j > 0 \quad \textit{if} \quad [z_i^*]_j \le 0, \tag{3.8b}$$

$$[\delta_c^*]_j = 0 \quad \textit{if} \quad [z_1^*]_j < 0 \quad \textit{or} \quad [z_2^*]_j < 0, \tag{3.8c}$$

$$[\delta_c^*]_j > 0 \quad \textit{if} \quad [z_1^*]_j \ge 0 \quad \textit{and} \quad [z_2^*]_j \ge 0 \tag{3.8d}$$

for $i = 1, 2$ and $j = 1, \dots, n$. Then

$$\max(\delta_c^*, \delta_1^*) > 0 \quad \textit{and} \quad \max(\delta_c^*, \delta_2^*) > 0, \tag{3.9}$$

and the point (x^*, s^*, y^*, v^*), with

$$(s_0^*, s_1^*, s_2^*) = (x_0^*, x_1^* + \delta_1^*, x_2^* + \delta_2^*), \tag{3.10a}$$

$$(v_0^*, v_1^*, v_2^*) = (z_0^*, [z_1^*]^+, [z_2^*]^+), \tag{3.10b}$$

$$s_c^* = \delta_c^*, \tag{3.10c}$$

and

$$[v_c^*]_j = \begin{cases} [-z_1^*]_j^+/[x_2^*]_j & \text{if} \quad [x_2^*]_j > 0 \quad (\text{and } [x_1^*]_j = 0), \\ [-z_2^*]_j^+/[x_1^*]_j & \text{if} \quad [x_1^*]_j > 0 \quad (\text{and } [x_2^*]_j = 0), \\ 0 & \text{if} \quad [x_1^*]_j = [x_2^*]_j = 0 \end{cases} \tag{3.10d}$$

for $j = 1, \dots, n$, is a stationary point for (MPEC-δ^*). Moreover, if (x^*, y^*, z^*) satisfies MPEC- LICQ, -WSCS, or -SSOSC for (MPEC), then (x^*, s^*, y^*, v^*) satisfies the LICQ, SCS, or SOSC, respectively, for (MPEC-δ^*).

Proof. The proof is divided into three parts. The first part demonstrates that (x^*, s^*, y^*, v^*) is a stationary point for (MPEC-δ^*) and that SCS is satisfied. The second and third parts prove that LICQ and SOSC hold for (MPEC-δ^*), respectively, if MPEC-LICQ and MPEC-SOSC hold.

Part 1. *Stationarity and SCS.* We first need to show that (3.9) holds. For $j = 1, \dots, n$ consider the following cases. If $[z_1^*]_j, [z_2^*]_j \leq 0$, then by (3.8b) we have that $[\delta_1^*]_j, [\delta_2^*]_j > 0$, and thus (3.9) holds. Note that the case $[z_1^*]_j > 0$ and $[z_2^*]_j < 0$ (or $[z_1^*]_j < 0$ and $[z_2^*]_j > 0$) cannot take place because otherwise (2.1e)–(2.1f) imply that $[x_1^*]_j, [x_2^*]_j = 0$, and then (2.1g) requires $[z_1^*]_j, [z_2^*]_j \geq 0$, which is a contradiction. Finally, if $[z_1^*]_j, [z_2^*]_j \geq 0$, then by (3.8d) we have that $[\delta_c^*]_j > 0$. Thus (3.9) holds, as required.

Next we verify stationarity of (x^*, s^*, y^*, v^*) for (MPEC-δ^*). The point (x^*, y^*, z^*) is strongly stationary for (MPEC), and so by Definition 2.1, it satisfies conditions (2.1). Then from (3.5), (3.6), and (3.10), (x^*, y^*, s^*, v^*) satisfies (3.7a) and (3.7c)–(3.7d).

We now show that s^* and v^* satisfy (3.7b). First, note from (3.10) that $s^*, v^* \geq 0$ because $x^* \geq 0$ and $\delta_c^*, \delta_1^*, \delta_2^* \geq 0$.

To see that s^* and v^* are componentwise strictly complementary if WSCS holds for the (MPEC), recall that WSCS requires that x_0^* and z_0^* are strictly complementary; hence (3.10a) and (3.10b) imply that s_0^* and v_0^* are also strictly complementary. For x_1^* and x_2^*, consider the indices $i = 1, 2$. If $[z_i^*]_j = 0$, then $[v_i^*]_j = 0$ and $[\delta_i^*]_j > 0$. From (3.10a) it follows that $[s_i^*]_j > 0$, as required. If $[z_i^*]_j > 0$, then (3.10b) implies that $[v_i^*]_j > 0$. Moreover, by (2.1e)–(2.1f), and (3.8a), $[x_i^*]_j = [\delta_i^*]_j = 0$. Hence $[s_i^*]_j = 0$, and $[s_i^*]_j$ and $[v_i^*]_j$ are strictly complementary, as required. If $[z_i^*]_j < 0$, then $[v_i^*]_j = 0$, and by (3.10a) and (3.8b), $[s_i^*]_j > 0$. Hence $[s_i^*]_j$ and $[v_i^*]_j$ are again strictly complementary. It remains to verify that $[s_c^*]_j$ and $[v_c^*]_j$ are strictly complementary. If $[s_c^*]_j = 0$, then (3.10c) and (3.8c) imply that $[z_1^*]_j < 0$ or $[z_2^*]_j < 0$ and $[v_c^*]_j > 0$ by (3.10d), as required. If $[s_c^*]_j > 0$ then (3.10c) implies that $[\delta_c^*]_j > 0$, and by (3.8d) we have that $[z_1^*]_j \geq 0$ and $[z_2^*]_j \geq 0$. Then by (3.10d) we know that $[v_c^*]_j = 0$.

Part 2. *LICQ.* Next we prove that (x^*, s^*, y^*, v^*) satisfies LICQ for (MPEC-δ^*) if (x^*, y^*, z^*) satisfies MPEC-LICQ for (MPEC). Note that LICQ holds for (MPEC-δ^*)

if and only if LICQ holds at x^* for the following system of equalities and inequalities:

$$c(x) = 0, \tag{3.11a}$$

$$x_0 \geq 0, \tag{3.11b}$$

$$x_1 \geq -\delta_1^*, \tag{3.11c}$$

$$x_2 \geq -\delta_2^*, \tag{3.11d}$$

$$X_1 x_2 \leq \delta_c^*. \tag{3.11e}$$

But MPEC-LICQ implies that the following system of equalities and inequalities satisfies LICQ at x^*:

$$c(x) = 0, \qquad x \geq 0. \tag{3.12}$$

We now show that the gradients of the active constraints in (3.11) are either a subset or a nonzero linear combination of the gradients of the active constraints in (3.12), and that therefore they must be linearly independent at x^*. To do so, for $j = 1, \ldots, n$, we consider the two cases $[\delta_c^*]_j > 0$ and $[\delta_c^*]_j = 0$.

If $[\delta_c^*]_j > 0$, the feasibility of x^* with respect to (MPEC) implies that the inequality $[X_1^* x_2^*]_j \leq [\delta_c^*]_j$ is not active. Moreover, because $\delta_1^*, \delta_2^* \geq 0$ and x^* is feasible with respect to (MPEC), we have that if the constraint $[x_1^*]_j \geq -\delta_1^*$ or $[x_2^*]_j \geq -\delta_2^*$ is active, then the corresponding constraint $[x_1^*]_j \geq 0$ or $[x_2^*]_j \geq 0$ is active. Thus, for the case $[\delta_c^*]_j > 0$, the set of constraints active in (3.11) is a subset of the set of constraints active in (3.12).

Now consider the case $[\delta_c^*]_j = 0$. By (3.9) we have that $[\delta_1^*]_j, [\delta_2^*]_j > 0$, and because x^* is feasible for (MPEC), the jth component (3.11e) is active, but the jth components of (3.11c)–(3.11d) are inactive. In addition, note that the gradient of this constraint has all components equal to zero except $\partial[X_1^* x_2^*]_j / \partial[x_1]_j = [x_2^*]_j$ and $\partial[X_1^* x_2^*]_j / \partial[x_2]_j = [x_1^*]_j$. Moreover, by (3.8c) we know that either $[z_1^*]_j$ or $[z_2^*]_j$ is strictly negative, and thus by (2.1g) we have that $[\max(x_1^*, x_2^*)]_j > 0$. Also, because x^* is feasible for (MPEC), $[\min(x_1^*, x_2^*)]_j = 0$. Thus one, and only one, of $[x_1^*]_j$ and $[x_2^*]_j$ is zero, and thus the gradient of the active constraint $[x_1^*]_j [x_2^*]_j \leq [\delta_c^*]_j$ is a nonzero linear combination of the gradient of whichever of the two constraints $[x_1^*]_j \geq 0$ and $[x_2^*]_j \geq 0$ is active.

Thus, the gradients of the constraints active in system (3.11) are either a subset or a nonzero linear combination of the constraints active in (3.12), and thus LICQ holds for (MPEC-δ^*).

Part 3. *SOSC.* To complete the proof, we need to show that SSOSC at (x^*, y^*, z^*) for (MPEC) implies SOSC at (x^*, s^*, y^*, v^*) for (MPEC-δ^*). Because the slack variables appear linearly in (MPEC-δ^*), we need only to show that (x^*, y^*, v^*) satisfies SOSC for the equivalent problem without slack variables,

$$\begin{array}{rr} \underset{x,s}{\text{minimize}} & f(x) \\ \text{subject to} & c(x) = 0, \\ & x_0 \geq 0, \\ & x_1 \geq -\delta_1^*, \\ & x_2 \geq -\delta_2^*, \\ & X_1 x_2 \leq \delta_c^*, \end{array} \tag{3.13}$$

and with solution (x^*, y^*, v^*). First, we show that the set of critical directions at (x^*, y^*, v^*) for (3.13) is equal to $\mathcal{F}$ (see Definition 2.6). Consider the critical directions for the first two constraints of (3.13). Because the constraints $c(x) = 0$ and $x_0 \geq 0$ and their multipliers are the same for (MPEC) and (3.13), their contribution to the definition of the set of critical directions is the same. In particular, we need to consider critical directions such that $A(x^*)p = 0$, $[p_0]_j \geq 0$ for all j such that $[x_0^*]_j = 0$, and $[p_0]_j = 0$ for all j such that $[z_0^*]_j > 0$. Next, consider the critical direction for the last three constraints of (3.13), $x_1 \geq -\delta_1^*$, $x_2 \geq -\delta_2^*$ and $X_1 x_2 \leq \delta_c^*$. Because we have shown that SCS holds for (3.13) at (x^*, y^*, v^*), we need only to impose the condition $[p_i]_j = 0$ for all j such that $[v_i^*]_j > 0$ for $i = 1, 2$, and $[p_i]_j = 0$ for all i and j such that $[v_c^*]_j > 0$ and $[x_i^*]_j = 0$. But note that because of (3.10) and (3.8), this is equivalent to imposing $[p_i]_j = 0$ for all j such that $[x_i^*]_j = 0$ and $[z_i^*]_j \neq 0$ for $i = 1, 2$, which is the definition of $\mathcal{F}$.

But note that the Hessian of the Lagrangian for (3.13) is different from the Hessian of the Lagrangian for (MPEC). The reason is that in (3.13) the complementarity constraint $X_1 x_2 \leq \delta_c^*$ is included in the Lagrangian, whereas we excluded this constraint from the definition of the Lagrangian for (MPEC). But it is easy to see that this has no impact on the value of $p^T \nabla_{xx}^2 \mathcal{L}(x^*, y^*) p > 0$ for all $p \in \mathcal{F}$. To see this, note that the Hessian of $[X_1 x_2]_j$ has only two nonzero elements:

$$(3.14) \qquad \nabla^2_{[x_1]_j [x_2]_j} [X_1 x_2]_j = \begin{bmatrix} & 1 \\ 1 & \end{bmatrix}.$$

If $[v_c^*]_j = 0$, then the Hessian of the complementarity constraint $[X_1 x_2]_j \leq [\delta_c^*]_j$ is multiplied by zero, and thus the Hessian of the Lagrangian for (MPEC-δ^*) is the same as the Hessian of the Lagrangian for (MPEC). Now suppose $[v_c^*]_j \neq 0$. Because SCS holds for (3.13), we have that $[v_c^*]_j \neq 0$ implies that the set of critical directions satisfies either $[p_1]_j = 0$ or $[p_2]_j = 0$. This, together with (3.14), implies that $p^T \nabla_{xx}^2 ([X_1^* x_2^*]_j) p = 0$ for all $p \in \mathcal{F}$. In other words, the second derivative of the complementarity constraint over the axis $[x_1^*]_j = 0$ or $[x_2^*]_j = 0$ is zero. As a result, if MPEC-SOSC holds, then SOSC must hold for (MPEC-δ^*) because all other terms of the Hessians of the Lagrangians of both problems are the same and the sets of critical directions of both problems are the same. □

The corollary to Theorem 3.1 is much clearer, but it requires the additional condition that (x^*, s^*, y^*, z^*) is feasible for (MPEC)—in other words, the partitions x_1^* and x_2^* are nonnegative and complementary.

Corollary 3.2. *Suppose that δ^* satisfies* (3.9) *and that (x^*, s^*, y^*, v^*) is a solution of (MPEC-δ^*) such that* $\min(x_1^*, x_2^*) = 0$. *Then the point (x^*, y^*, z^*) is strongly stationary for (MPEC), where*

$$(3.15) \qquad z^* = B(x^*)^T v^*.$$

Proof. Equation (3.15) is derived by comparing (2.1) with (3.7). □

3.5. Relaxation parameter updates. In this section we show how to construct a sequence of relaxation parameters δ_k such that $\lim_{k \to \infty} \delta_k = \delta^*$, where δ^* satisfies (3.8)–(3.9). We are guided by Theorem 3.1 in developing such a parameter update. Under certain conditions (discussed in section 3.6), we can recover the solution of the original MPEC from the solution of (MPEC-δ^*).

Suppose that $w_k = (x_k, s_k, y_k, v_k)$ is an estimate of the solution of (MPEC-δ_k), and let $z_k = B(x_k)^T v_k$ be the corresponding MPEC multipliers given by (3.15). Given an improved estimate w_{k+1}, Algorithm 1 defines a set of rules for updating the relaxation parameter vector δ_k. The algorithm also updates a companion sequence

Algorithm 1. Relaxation parameter update.

```
    Input: $\delta_k, \delta_k^*, w_{k+1}, z_{k+1}$
    Set fixed parameters $\kappa, \tau \in (0, 1)$
1   [Compute bounds for the KKT residual]
    $\underline{r}_{k+1}^* \leftarrow \|r(w_{k+1}; \delta_k^*)\|^{1+\tau}$
    $\overline{r}_{k+1}^* \leftarrow \|r(w_{k+1}; \delta_k^*)\|^{1-\tau}$
    for $i = 1, 2$ and $j = 1, \dots, n$ do
2       [Update bound-constraint relaxations]
        if $[z_{ik+1}]_j > \overline{r}_{k+1}^*$ then
            $[\delta_{ik+1}]_j \leftarrow \min(\kappa[\delta_{ik}]_j, \underline{r}_{k+1}^*)$
            $[\delta_{ik+1}^*]_j \leftarrow 0$
        else
            $[\delta_{ik+1}]_j \leftarrow [\delta_{ik}]_j$
            $[\delta_{ik+1}^*]_j \leftarrow [\delta_{ik}]_j$
3       [Update complementarity-constraint relaxations]
        if $[z_{1k+1}]_j < -\overline{r}_{k+1}^*$ or $[z_{2k+1}]_j < -\overline{r}_{k+1}^*$ then
            $[\delta_{ck+1}]_j \leftarrow \min(\kappa[\delta_{ck}]_j, \underline{r}_{k+1}^*)$
            $[\delta_{ck+1}^*]_j \leftarrow 0$
        else
            $[\delta_{ck+1}]_j \leftarrow [\delta_{ck}]_j$
            $[\delta_{ck+1}^*]_j \leftarrow [\delta_{ck}]_j$
    return $\delta_{k+1}, \delta_{k+1}^*$
```

$\delta_k^* \equiv (0, \delta_{1k}^*, \delta_{2k}^*, \delta_{ck}^*)$ that defines a nearby relaxed problem (MPEC-δ_k^*). In the vicinity of the minimizer, this nearby relaxed problem gives an estimate of the active constraint set. Also, the residual of (MPEC-δ_k^*) is a better optimality measure than the residual of (MPEC-δ_k) because while all components of the relaxation parameter vector δ_k are strictly positive, some of the components of δ_k^* may be zero. The scalars $\underline{r}_k^*$ and $\overline{r}_k^*$ are lower and upper bounds on the KKT residual of (MPEC-δ_k^*); they provide a measure of nearness to zero of the MPEC multipliers and are used to predict the sign of the optimal MPEC multipliers.

3.6. Active-set identification. Suppose that δ_k^* is a set of relaxation parameters that satisfies (3.8) and that therefore defines a one-sided relaxation. Let $w_k^* = (x_k^*, s_k^*, y_k^*, v_k^*)$ be the minimizer of the associated relaxed problem (MPEC-δ_k^*) defined via (3.10), and let w_k be an estimate of w_k^*. If w_k is close enough to w^* and Algorithm 1 is given an improved estimate w_{k+1}, then it will return the same one-sided relaxation parameter $\delta_{k+1}^* = \delta_k^*$. Therefore, (MPEC-$\delta_k^*$) will remain fixed. Thus, the update rules continue to update (and reduce) the same relaxation parameters at every iteration—this property is used to guarantee that the feasible region remains nonempty even in the limit. In some sense, it implies that the correct active set is identified.

We make the following nondegeneracy assumptions about the MPEC minimizer (x^*, y^*, z^*). These assumptions hold throughout the remainder of the paper.

Assumption 3.3. *There exist strictly positive relaxation parameters δ such that the second derivatives of f and c are Lipschitz continuous over the set*

$$X_1 x_2 \le \delta_c, \qquad x_1 \ge -\delta_1, \qquad x_2 \ge -\delta_2.$$

Assumption 3.4. *The point (x^*, y^*, z^*) satisfies MPEC-LICQ for (MPEC).*

Assumption 3.5. *The point (x^*, y^*, z^*) satisfies MPEC-WSCS and -SSOSC for (MPEC).*

Theorem 3.6 proves that Algorithm 1 will leave the one-sided relaxation parameter unchanged if w_{k+1} improves the estimate w_k of w_k^*. Applied iteratively, the algorithm continues to update the same relaxation parameters δ_{ik} or δ_{ck}.

Note from (3.10) that the influence of δ_k^* on $w_k^* = (x_k^*, s_k^*, y_k^*, v_k^*)$ is relegated to only s_k^*. We may therefore write $w_k^* \equiv (x^*, s_k^*, y^*, v^*)$. We assume that δ_k^* satisfies (3.8). This implies that δ_k^* reveals the sign of the MPEC multipliers at the solution w^*. We also assume that the $(k+1)$th iterate w_{k+1} is closer to the minimizer than the kth iterate so that $\|w_{k+1} - w_k^*\| < \|w_k - w_k^*\|$. This assumption will hold whenever we apply a linearly convergent algorithm to compute w_{k+1} starting from w_k.

Theorem 3.6. *Let (x^*, y^*, z^*) be a strongly stationary point of (MPEC) and suppose that Assumptions* 3.3–3.5 *hold. Moreover, assume that δ_k^* satisfies* (3.8) *and that $[\delta_k^*]_j = [\delta_k]_j > 0$ for all j such that $[\delta_k^*]_j \neq 0$. Let $w_k^* = (x^*, s_k^*, y^*, v^*)$ be the solution of the corresponding relaxation (MPEC-δ_k^*) given by* (3.10)*, and assume that $\|w_{k+1} - w_k^*\| < \|w_k - w_k^*\|$. Then if w_k is close enough to w^*, the parameter δ_{k+1}^* generated by Algorithm* 1 *satisfies $\delta_{k+1}^* = \delta_k^*$.*

Proof. We first show that $\overline{r}_{k+1}^*$ is bounded above and below by a finite multiple of $\|w_{k+1} - w_k^*\|^{1-\tau}$. By definition of w_k^* and δ_k^*, $r(w_k^*; \delta_k^*) = 0$. Also, by the hypothesis of this theorem, both w_k and w_{k+1} are close to w^*. Moreover, Assumption 3.3 implies that the KKT residual $r(w; \delta)$ is differentiable. This by Taylor's theorem implies that

$$r(w_{k+1}; \delta_k^*) = K(w_k^*)(w_{k+1} - w_k^*) + O(\|w_{k+1} - w_k^*\|^2), \tag{3.16}$$

where $K(w_k^*)$ is the Jacobian of the KKT residual $r(w; \delta)$ with respect to w evaluated at w_k^*. Note that this Jacobian does not depend on δ_k^*. In addition, as a consequence of Theorem 3.1, $K(w_k^*)$ is nonsingular. This together with (3.16), imply that there exist positive constants $\beta_2 > \beta_1$ such that for w_{k+1} in the vicinity of w_k^*

$$\beta_1 \|w_{k+1} - w_k^*\| \leq \|r(w_{k+1}; \delta_k^*)\| \leq \beta_2 \|w_{k+1} - w_k^*\|.$$

Then, by the definition of $\overline{r}_{k+1}^*$ (Step 1 of Algorithm 1) we have that

$$\beta_3 \|w_{k+1} - w_k^*\|^{1-\tau} \leq \overline{r}_{k+1}^* \leq \beta_4 \|w_{k+1} - w_k^*\|^{1-\tau}, \tag{3.17}$$

where $\beta_3 = \beta_1^{1-\tau}$ and $\beta_4 = \beta_2^{1-\tau}$.

Let $\epsilon \equiv \frac{1}{2} \min(|[z^*]_j| \mid \text{for all } j \text{ such that } [z^*]_j \neq 0)$. Then, condition (3.17) and the assumptions that w_k is close enough to w^* and $\|w_{k+1} - w_k^*\| < \|w_k - w_k^*\|$ imply that

$$\overline{r}_{k+1}^* < \epsilon. \tag{3.18}$$

Moreover, because $z_{k+1} = B(x_{k+1})^T v_{k+1}$, $z^* = B(x^*)^T v^*$, and $B(x)$ is Lipschitz continuous by Assumption 3.3, we have that for w_k close enough to w^* and $\|w_{k+1} - w_k^*\| < \|w_k - w_k^*\|$ the following holds:

$$\|z_{k+1} - z^*\| < \epsilon. \tag{3.19}$$

Consider the indices $i = 1, 2$ and $j = 1, \ldots, n$. Suppose that $[z^*]_j > 0$. Then (3.18) and (3.19) imply that

$$[z_{ik+1}]_j = [z^*]_j + ([z_{ik+1}]_j - [z^*]_j) > [z^*]_j - \epsilon \geq \epsilon > \overline{r}_{k+1}^*. \tag{3.20}$$

Suppose instead that $[z^*]_j < 0$. Then (3.18) and (3.19) imply that

$$-\overline{r}^*_{k+1} > -\epsilon > [z^*]_j + \epsilon = [z_{ik+1}]_j - ([z_{ik+1}]_j - [z^*]_j) + \epsilon > [z_{ik+1}]_j.$$

Finally, suppose that $[z^*]_j = 0$. Then because $\tau > 0$, we have that for w_k close enough to w^* and $\|w_{k+1} - w^*_k\| < \|w_k - w^*_k\|$

$$|[z_{ik+1}]_j| = |[z_{ik+1}]_j - [z^*]_j| \leq \|w_{k+1} - w^*_k\| \leq \beta_3 \|w_{k+1} - w^*_k\|^{1-\tau} \leq \overline{r}^*_{k+1}. \tag{3.21}$$

Because $\delta_k > 0$, the updates in Algorithm 1 imply that $\delta_{k+1} > 0$, and together with (3.20)–(3.21), we have that δ^*_{k+1} satisfies (3.8). This in turn implies that the set of indices j for which $[\delta^*_{k+1}]_j \neq 0$ coincides with the set of indices j for which $[\delta^*_k]_j \neq 0$. For this same set of indices, moreover, the parameter updates imply that $[\delta_{k+1}]_j = [\delta_k]_j$. Then because $[\delta^*_k]_j = [\delta_k]_j$ for such j, the update rules imply that $\delta^*_{k+1} = \delta^*_k$, as required. □

Note that $\delta^*_{k+1} = \delta^*_k$ implies that $w^*_{k+1} = w^*_k$; that is, w^*_k is also a local minimizer for the relaxed problem for the $(k+1)$th iterate.

4. An interior-point algorithm.

The discussion thus far has not made use of a specific optimization algorithm. Theorem 3.6 makes use of an improved estimate of (MPEC-δ^*_k) but does not specify the manner in which it is computed. In this section we show how to construct a primal-dual interior-point algorithm that at each iteration will satisfy the conditions of Theorem 3.6. The parameter update rule in Algorithm 1 is invoked at each iteration of the interior method. The barrier parameter is updated simultaneously. This iteration scheme is repeated until certain convergence criteria are satisfied.

4.1. Algorithm summary.

For the remainder of this section, we omit the dependence of each variable on the iteration counter k when the meaning of a variable is clear from its context. The search direction is computed by means of Newton's method on the KKT conditions of the barrier subproblem corresponding to (MPEC-δ). These are given by (3.7), where (3.7b) is replaced by

$$Sv - \mu e \equiv r_\mu = 0 \tag{4.1}$$

and $\mu > 0$ is the barrier parameter. An iteration of Newton's method based on (3.7) (where (4.1) replaces (3.7b)) computes a step direction by solving the system

$$K(w)\mathit{\Delta} w = -r(w; \mu, \delta), \tag{4.2}$$

where $\mathit{\Delta} w \equiv (\mathit{\Delta} x, \mathit{\Delta} s, \mathit{\Delta} y, \mathit{\Delta} v)$ and $r(w; \mu, \delta) \equiv (r_d, r_\mu, r_f, r_\delta)$ is the KKT residual of the barrier problem. (Note the identity $r(w; 0, \delta) \equiv r(w; \delta)$.) The Jacobian K is independent of the barrier *and* relaxation parameters—these appear only in the right-hand side of (4.2). This is a useful property because it considerably simplifies the convergence analysis in section 4.2.

To ensure that s and v remain strictly positive (as required by interior-point methods), each computed Newton step $\mathit{\Delta} w$ may need to be truncated. Let γ be a steplength parameter such that $0 < \gamma < 1$. At each iteration we choose a steplength α so that

$$\alpha = \min(\alpha_s, \alpha_v), \tag{4.3}$$

where

$$\alpha_d = \min\left(1, \gamma \min_{[\Delta d]_j < 0} -[d]_j / [\Delta d]_j\right), \qquad d = \{s, v\}.$$

Algorithm 2. Interior-Point Relaxation for MPECs.

```
Input: $x_0, y_0, z_0$
Output: $x^*, y^*, z^*$
[Initialize variables and parameters]
    Choose starting vectors $s_0, v_0 > 0$. Set $w_0 = (x_0, s_0, y_0, v_0)$. Set the relaxation
    and barrier parameters $\delta_0, \mu_0 > 0$. Set parameters $0 < \kappa, \tau, \bar{\gamma} < 1$. Set the
    starting steplength parameter $\bar{\gamma} \le \gamma_0 < 1$. Set the convergence tolerance $\epsilon > 0$.
$k \leftarrow 0$
repeat
    [Compute the Newton step]
        Solve (4.2) for $\Delta w_k$
1   [Truncate the Newton Step]
        Determine the maximum steplength $\alpha_k$, given by (4.3);
        $w_{k+1} \leftarrow w_k + \alpha_k \Delta w_k$
    [Compute MPEC multipliers]
        $z_{k+1} \leftarrow B(x_{k+1})^T v_{k+1}$
    [Update relaxation parameters]
        Compute $\delta_{k+1}, \delta^*_{k+1}$ with Algorithm 1
2   [Update barrier and step parameters]
        $\mu_{k+1} = \min(\kappa\mu_k, \underline{r}^*_{k+1})$        [Ensures that $\lim_{k\to\infty} \mu_k = 0$]
        $\gamma_{k+1} = \max(\bar{\gamma}, 1 - \mu_{k+1})$        [Ensures that $\lim_{k\to\infty} \gamma_k = 1$]
    $k \leftarrow k + 1$
until (4.4) holds;
$x^* \leftarrow x_k$; $y^* \leftarrow y_k$; $z^* \leftarrow z_k$
return $x^*, y^*, z^*$
```

Because our analysis focuses on the local convergence properties of the proposed algorithm, the $(k+1)$th iterate is computed as $w_{k+1} = w_k + \alpha \Delta w_k$. (A globalization scheme that can choose shorter steps is discussed in section 5.)

Algorithm 2 outlines the interior-point relaxation method. The method takes as a starting point the triple (x_0, y_0, z_0) as an estimate of a solution of the relaxed NLP corresponding to (MPEC). The algorithm terminates when the optimality conditions for (MPEC-δ^*_k) are satisfied, that is, when

$$\|r(w_k; \delta^*_k)\| < \epsilon \tag{4.4}$$

for some small and positive ϵ. Recall that $w^*_k = (x^*_k, s^*_k, y^*_k, v^*_k)$ is the solution to the one-sided relaxation (MPEC-δ^*_k); therefore, $\|r(w^*_k; \delta^*_k)\| = 0$. Note that we never compute w^*_k—it is used only as an analytical device.

4.2. Superlinear convergence. In this section we analyze the local convergence properties of the interior-point relaxation algorithm. The distinguishing feature of the proposed algorithm is the relaxation parameters and their associated update rules. If we were to hold the relaxation parameters constant, the relaxation method would reduce to a standard interior-point algorithm applied to a fixed relaxed MPEC; it would converge locally and superlinearly provided that the starting iterate is close to a nondegenerate minimizer of (MPEC-δ_k) (and that standard assumptions held). The main challenge is to show that the interior-point relaxation algorithm continues to converge locally and superlinearly even when the relaxation parameters change at each iteration. We use the shorthand notation $r_k \equiv r(w_k; \mu_k, \delta_k)$ and $r^*_k \equiv r(w_k; \delta^*_k)$.

THEOREM 4.1. *Let (x^*, y^*, z^*) be a strongly stationary point of (MPEC) and suppose that Assumptions* 3.3–3.5 *hold. Assume that δ_k^* satisfies* (3.8), *and let $w_k^* = (x^*, s_k^*, y^*, v^*)$ be the solution of the corresponding relaxation (MPEC-δ_k^*) given by* (3.10). *Then there exists $\epsilon > 0$ and $\beta > 0$ such that if Algorithm* 2 *is started with iterates at $k = 0$ that satisfy*

$$\|w_k - w_k^*\| < \epsilon, \tag{4.5}$$

$$\|\delta_k - \delta_k^*\| < \beta \|w_k - w_k^*\|^{1+\tau}, \tag{4.6}$$

$$\mu_k < \beta \|w_k - w_k^*\|^{1+\tau}, \tag{4.7}$$

$$1 - \gamma_k < \beta \|w_k - w_k^*\|^{1+\tau}, \tag{4.8}$$

and

$$[\delta_k^*]_j = [\delta_k]_j > 0 \quad \text{for all j such that} \quad [\delta_k^*]_j \neq 0, \tag{4.9}$$

then the sequence $\{w_k^*\}$ is constant over all k and $\{w_k\}$ converges Q-superlinearly to $w^* \equiv w_k^*$.

Proof. The proof has three parts. First, we show that there exists a constant $\sigma > 0$ such that $\|w_{k+1} - w_k^*\| \leq \sigma \|w_k - w_k^*\|^{1+\tau}$. Second, we show that $\delta_{k+1}^* = \delta_k^*$, and thus that w_k^* is also a minimizer to the relaxed MPEC corresponding to the $(k+1)$th iterate. Finally, we show that the conditions of the theorem hold also for the $(k+1)$th iterate. The main result therefore follows by induction.

Part 1. $\|w_{k+1} - w_k^*\| \leq \sigma \|w_k - w_k^*\|^{1+\tau}$. From Assumptions 3.3–3.5 and Theorem 3.1 we know that $K(w_k)$ is nonsingular for all $\epsilon > 0$ small enough, so that $\|K(w_k)^{-1}\|$ is bounded in the vicinity of w^*. Consider only such ϵ. Define the vector $\eta_k^* = (0, \mu_k e, 0, \delta_k^* - \delta_k)$ with components partitioned as per (3.7). (Note that $r_k = r_k^* - \eta_k^*$.) Then

$$\begin{aligned} w_{k+1} - w_k^* &= w_k - w_k^* - \alpha_k K(w_k)^{-1} r_k \\ &= (1 - \alpha_k)(w_k - w_k^*) + \alpha_k K(w_k)^{-1} (K(w_k)(w_k - w_k^*) - r_k^* + \eta_k^*) \\ &= (1 - \alpha_k)(w_k - w_k^*) \\ &\quad + \alpha_k K(w_k)^{-1} \eta_k^* + \alpha_k K(w_k)^{-1} (K(w_k)(w_k - w_k^*) - r_k^*). \end{aligned} \tag{4.10}$$

Each term on the right-hand side of (4.10) can be bounded as follows. Because (x^*, y^*, z^*) is a strongly stationary point of (MPEC) satisfying assumptions 3.3–3.5, Theorem 3.1 applies. Therefore, w^* satisfies LICQ, SCS, and SOSC for (MPEC-δ^*). Then by Lemma 5 of [24] we know that there exists a positive constant ϵ_1 such that $|1 - \alpha_k| \leq 1 - \gamma_k + \epsilon_1 \|\Delta w_k\|$. Therefore

$$\|(1 - \alpha_k)(w_k - w_k^*)\| \leq \big((1 - \gamma_k) + \epsilon_1 \|\Delta w_k\|)\big) \|w_k - w_k^*\|. \tag{4.11}$$

We now further bound the right-hand side of (4.11). Because $\|K(w_k)^{-1}\|$ is bounded for ϵ small enough, there exists a positive constant ϵ_2 such that

$$\|\Delta w_k\| = \|K(w_k)^{-1}(-r_k^* + \eta_k^*)\| \leq \epsilon_2 (\|r_k^*\| + \|\eta_k^*\|). \tag{4.12}$$

Assumption 3.3 implies that the KKT residual $r(w; \mu, \delta)$, and thus, $r(w; \delta)$, is differentiable. Hence there exists a positive constant ϵ_3 such that

$$\|r_k^*\| = \|r(w_k; \delta_k^*) - r(w_k^*; \delta_k^*)\| \leq \epsilon_3 \|w_k - w_k^*\|. \tag{4.13}$$

Moreover, (4.6) and (4.7) imply that there exists a positive constant ϵ_4 such that

$$\|\eta_k^*\| \le \epsilon_4 \|w_k - w_k^*\|^{1+\tau}. \tag{4.14}$$

Then substituting (4.12), (4.13), (4.14), and condition (4.8), into (4.11) we have

$$\|(1-\alpha_k)(w_k - w_k^*)\| \le (\beta + \epsilon_1\epsilon_2\epsilon_4)\|w_k - w_k^*\|^{2+\tau} + \epsilon_1\epsilon_2\epsilon_3\|w_k - w_k^*\|^2. \tag{4.15}$$

From the boundedness of $\|K(w_k)^{-1}\|$ around w_k^* and (4.14), the second term in (4.10) satisfies

$$\|\alpha_k \; K(w_k)^{-1}\eta_k^*\| \le \alpha_k \|K(w_k)^{-1}\| \; \|\eta_k^*\| \le \epsilon_5 \|w_k - w_k^*\|^{1+\tau}$$

for some positive constant ϵ_5. Finally, the third term in (4.10) satisfies (using Taylor's theorem and again the fact that $\|K(w_k)^{-1}\|$ is bounded around w_k^*)

$$\|\alpha_k \; K(w_k)^{-1}(K(w_k)(w_k - w_k^*) - r_k^*)\| \le \epsilon_6 \|w_k - w_k^*\|^2 \tag{4.16}$$

for some positive constant ϵ_6. Hence, (4.10) and (4.15)–(4.16) yield

$$\|w_{k+1} - w_k^*\| \le \sigma \|w_k - w_k^*\|^{1+\tau} \tag{4.17}$$

for some positive constant σ, as required.

Part 2. $\delta_{k+1}^* = \delta_k^*$. Note that by (4.17) we know that for ϵ small enough the assumptions of Theorem 3.6 hold and therefore $\delta_{k+1}^* = \delta_k^*$. As a result, w_k^* is also a minimizer of (MPEC-δ_{k+1}^*).

Part 3. *The theorem hypotheses also hold for the* $(k+1)$*th iterate.* As $\delta_{k+1}^* = \delta_k^*$, then δ_{k+1}^* satisfies (3.8). Moreover, (4.17) implies for ϵ small enough that (4.5) holds for w_{k+1}. Because $r(w;\mu,\delta)$ is differentiable, Theorem 3.1 implies that $K(w)$, the Jacobian of $r(w;\mu,\delta)$ with respect to w, is bounded in the vicinity of w_k^*. Together with the definition of $\overline{r}_{k+1}^*$, the fact that $\delta_{k+1}^* = \delta_k^*$, Steps 2 and 3 of Algorithm 1, and Step 2 of Algorithm 2, this implies that (4.6)–(4.9) hold for δ_{k+1}, μ_{k+1}, and γ_{k+1}.

The proof finishes noting that $w_{k+1}^* = w_k^*$ because $\delta_{k+1}^* = \delta_k^*$, so that by induction, $w^* = w_k^*$ for all iterations $k+1, k+2, \ldots$. The superlinear convergence of w_k to w^* then follows by induction from (4.17). □

Note that in addition to the assumptions made in Theorem 3.6, we assume that the barrier and steplength parameters satisfy $\mu_k < \beta\|w_k - w_k^*\|^{1+\tau}$ and $1 - \gamma_k < \beta\|w_k - w_k^*\|^{1+\tau}$ for some $\tau \in (0,1)$ and $\beta > 0$. These are standard assumptions used to prove superlinear convergence of interior methods. They imply the barrier and steplength parameters are updated fast enough. In addition, we assume that $\|\delta_k - \delta_k^*\| < \beta\|w_k - w_k^*\|^{1+\tau}$. This assumption implies that the distance between δ_k and δ_k^* is small compared to the distance between the current iterate w_k and the minimizer w^*. Note that in Part 3 of the proof of Theorem 4.1, we show that this assumption will hold when the relaxation parameter update rule in Algorithm 1 is applied for two or more iterations. Finally, the technical Assumption 4.9 simplifies the proof and that is also satisfied whenever Algorithm 1 is applied for two or more consecutive iterations.

5. Implementation details. In this section we discuss two practical aspects of our implementation. First, to globalize the interior-point method, we perform a backtracking linesearch on an augmented Lagrangian merit function (although other globalization schemes could be used). The theoretical properties of this merit function have been analyzed by Moguerza and Prieto [14]. We also modify the Jacobian $K(w)$ as in [23] to ensure a sufficient descent direction for the augmented Lagrangian merit function.

Second, we make use of a safeguard to the relaxation parameter update that prevents the algorithm from converging to stationary points of the relaxed MPEC that are not feasible with respect to MPEC. To see how this may happen, again consider the example MPEC (3.3). The relaxed MPEC (with slack variables) is given by

$$\begin{array}{ll} \underset{x_1,x_2,s_1,s_2,s_c \in \mathrm{R}}{\text{minimize}} & \frac{1}{2}(x_1-a_1)^2+\frac{1}{2}(x_2-a_2)^2 \\ \text{subject to} & s_1 - x_1 = \delta_1, \\ & s_2 - x_2 = \delta_2, \\ & s_c + X_1 x_2 = \delta_c, \\ & s \geq 0. \end{array} \tag{5.1}$$

For $a_1 = a_2 = 0.01$, $\delta_c = 1$, and $\delta_1 = \delta_2 = 0$, the point

$$(x_1, x_2, s_1, s_2, s_c) = (0.01, 0.01, 0.01, 0.01, 0.9999)$$

with multipliers $(v_1, v_2, v_c) = (0, 0, 0)$ is clearly a stationary point of (5.1), but it is not feasible for (3.3). However, note that a point $(x_0, x_1, x_2, s_0, s_1, s_2, s_c)$ feasible for (MPEC-δ) is feasible for (MPEC) if and only if

$$(s_0, s_1, s_2, s_c) = (x_0, x_1 + \delta_1, x_2 + \delta_2, \delta_c) \tag{5.2}$$

(cf. (3.10a)). To ensure that (5.2) always holds at the limit point, we propose a modification of the bound-constraint relaxations in Steps 2 and 3 of Algorithm 1. The proposed modification is the following:

$$\begin{array}{llll} [\delta_{ik+1}]_j = \min(\kappa[\delta_{ik}]_j, \underline{r}^*_{k+1}) & \text{if} & [z_{ik+1}]_j > \overline{r}^*_{k+1}, & \\ [\delta_{ik+1}]_j = \min([\delta_{ik}]_j, [s_{ik}]_j) & \text{if} & [z_{ik+1}]_j \leq \overline{r}^*_{k+1}, & \\ [\delta_{ck+1}]_j = \min(\kappa[\delta_{ck}]_j, \underline{r}^*_{k+1}) & \text{if} & [z_{1k+1}]_j < -\overline{r}^*_{k+1} \quad \text{or} & [z_{2k+1}]_j < -\overline{r}^*_{k+1}, \\ [\delta_{ck+1}]_j = \min([\delta_{ck}]_j, [s_{ck}]_j) & \text{if} & [z_{1k+1}]_j \geq -\overline{r}^*_{k+1} \quad \text{and} & [z_{2k+1}]_j \geq -\overline{r}^*_{k+1} \end{array}$$

for $i = 1, 2$ and $j = 1, \ldots, n$.

Thus, the above parameter update prevents the algorithm from converging to spurious stationary points for the relaxed MPEC that are not stationary for the MPEC.

Finally, it is possible to show that the local convergence results of previous sections still hold when using both the globalization strategy for the interior point method and the safeguard of the relaxation parameter update. But to simplify the exposition, we have decided to leave these two aspects out of the local convergence analysis of previous sections.

6. Numerical results. We illustrate in this section the numerical performance of the interior-point relaxation algorithm on the MacMPEC test problem set [11]. The results confirm our local convergence analysis and show that our implementation performs well in practice.

The interior-point relaxation algorithm has been implemented as a Matlab program. Problems from the MacMPEC test suite (coded in AMPL [11]) are accessed via a Matlab MEX interface. Because the algorithm has been implemented using dense linear algebra, we apply the method to a subset of 87 small- to medium-size problems from the MacMPEC test suite.

We stop the algorithm under three different circumstances: (i) if the iteration limit of 150 is exceeded; (ii) if the current iterate is a stationary point of (MPEC-δ^*), i.e., if $\|r(w_k; 0, \delta_k^*)\| < 10^{-6}(1 + \|\nabla f(x_k)\|)$ (cf. (4.4)); or (iii) if the steplength is too small. We use the following parameter values for the barrier and relaxation updates: $\tau = 0.3$ and $\kappa = 0.9$.

Table 6.1 gives information regarding the performance of our algorithm on each test problem. The first column indicates the name of the problem, the second and third columns indicate the number of iterations and function evaluations, the fourth column shows the final objective function value, the fifth and sixth columns show the norm of the multiplier vector v_c and the norm of the KKT residual of the nearby relaxed MPEC (MPEC-δ^*) at the solution, and the last two columns indicate the exit status of the algorithm. The exit flags are described in Table 6.2. The quantities $(\delta_1^*, \delta_2^*, \delta_c^*)$ are the final values of the relaxation parameters.

The results seem to confirm that the global convergence safeguards proposed in section 5 are effective in practice. In particular, the algorithm converges to a strongly stationary point of (MPEC) for most of the test problems in the collection, that is, flag1 = 1 for most of the problems. Moreover, note that all stationary points of (MPEC-δ^*) found by the algorithm are also strongly stationary for the original MPEC; that is, flag1 is never equal to 2. Finally, some of the problems on which our algorithm fails are ill-posed according to [18, 4, 3]. For instance, *ex9.2.2*, *qpec2*, *ralph1*, *scholtes4*, and *tap-15* do not have a strongly stationary point, the *pack* problems have an empty strictly feasible region, *ralphmod* is unbounded, and *design-cent-3* is infeasible.

Table 6.1
Performance of the interior-point relaxation algorithm on the selected MacMPEC test problems.

Problem	iter	nfe	f	$\|v_c^*\|$	$\|r\|$	flag1	flag2
bar-truss-3	36	73	1.017e+04	4.521e+00	4.543e−04	1	1
bard1	13	27	1.700e+01	7.621e−01	4.170e−04	1	1
bard2	66	133	6.163e+03	1.036e+01	5.221e−05	1	1
bard3	16	33	-1.268e+01	3.625e−01	3.225e−06	1	1
bard1m	88	397	1.700e+01	1.504e−03	1.373e−04	1	0
bard2m	66	133	-6.598e+03	1.128e−04	5.444e−05	1	1
bard3m	16	33	-1.268e+01	1.350e+00	4.770e−06	1	1
bilevel1	16	33	5.000e+00	8.700e−02	1.382e−06	1	1
bilevel2	67	135	-6.600e+03	3.848e−01	3.174e−04	1	1
bilevel3	83	277	-8.636e+00	4.587e−03	8.352e−04	1	0
bilin	24	49	-1.215e−04	1.996e+00	1.513e−03	1	0
dempe	17	35	3.125e+01	5.002e+00	3.619e−06	1	1
design-cent-2	150	774	-3.182e−15	2.024e−05	3.749e+02	0	1
design-cent-3	150	2649	3.546e−02	1.930e+00	7.977e+00	0	1
design-cent-4	99	425	1.508e−18	3.616e−04	1.027e−08	1	1
ex9.1.1	19	39	-1.300e+01	1.087e+00	1.343e−03	1	0
ex9.1.2	14	29	-6.250e+00	1.902e+00	1.110e−03	1	0
ex9.1.3	39	80	-2.920e+01	5.357e+00	4.327e−03	1	1
ex9.1.4	33	80	-3.700e+01	1.999e+00	1.389e−07	1	1
ex9.1.5	11	23	-1.000e+00	3.674e+00	6.727e−06	1	1
ex9.1.6	22	47	-1.500e+01	1.000e+00	1.848e−05	1	0

Table 6.1
Cont'd.

Problem	iter	nfe	f	$\|v_c^*\|$	$\|r\|$	flag1	flag2
ex9.1.7	87	310	-2.600e+01	2.001e+00	1.497e−03	1	0
ex9.1.8	102	441	-3.250e+00	3.180e+00	1.694e−01	1	0
ex9.1.9	26	63	3.111e+00	2.678e+00	3.081e−03	1	1
ex9.1.10	102	441	-3.250e+00	3.180e+00	1.694e−01	1	0
ex9.2.1	19	39	1.700e+01	2.881e+00	7.365e−04	1	0
ex9.2.2	150	655	1.000e+02	7.374e+03	1.402e−02	0	1
ex9.2.3	16	33	5.000e+00	4.700e−09	2.093e−08	1	1
ex9.2.4	10	21	5.000e−01	1.000e+00	1.778e−08	1	1
ex9.2.5	13	27	9.000e+00	6.185e+00	1.646e−06	1	1
ex9.2.6	66	239	-1.000e+00	7.071e−01	1.981e−02	1	0
ex9.2.7	19	39	1.700e+01	2.881e+00	7.365e−04	1	0
ex9.2.8	12	25	1.500e+00	5.000e−01	1.080e−06	1	1
ex9.2.9	13	27	2.000e+00	1.987e+00	4.019e−08	1	1
flp2	22	49	1.076e−17	1.517e−05	3.595e−04	1	1
flp4-1	35	75	5.411e−07	1.315e−06	2.607e−06	1	1
flp4-2	41	89	7.376e−07	4.076e−06	8.233e−06	1	1
flp4-3	52	126	1.018e−06	1.913e−06	3.905e−06	1	1
flp4-4	56	117	2.456e−06	7.803e−07	8.825e−06	1	1
gauvin	11	23	2.000e+01	2.500e−01	8.152e−07	1	1
hakonsen	150	351	1.113e+01	4.898e−05	2.825e−01	0	1
hs044-i	83	279	3.765e+01	2.271e+00	1.344e−01	1	0
incid-set1-8	54	117	5.016e−06	1.722e−04	3.536e−06	1	1
incid-set1c-8	101	210	4.554e−06	9.816e−04	3.754e−06	1	1
incid-set2-8	149	302	8.929e+00	2.069e+03	2.363e+04	7	1
jr1	8	17	5.000e−01	5.779e−09	1.259e−08	1	1
jr2	8	17	5.000e−01	2.000e+00	2.282e−08	1	1
kth1	9	19	3.950e−07	7.046e−07	5.989e−07	1	1
kth2	8	17	1.432e−09	1.355e−07	2.180e−07	1	1
kth3	7	15	5.000e−01	1.000e+00	9.131e−07	1	1
liswet1-050	36	89	1.399e−02	2.552e−09	5.998e−09	1	1
nash1	26	53	1.339e−07	2.499e−04	3.930e−04	1	1
outrata31	88	184	3.208e+00	3.234e+01	3.653e−07	1	0
outrata32	86	177	3.449e+00	6.586e+01	4.908e−07	1	0
outrata33	83	174	4.604e+00	6.089e+02	2.808e−06	1	0
outrata34	107	218	6.593e+00	8.386e+00	1.549e−06	1	0
pack-comp1-8	97	818	6.240e−01	5.388e+01	5.923e+04	7	1
pack-comp1c-8	126	300	5.741e−01	1.308e+01	2.099e+04	7	0
pack-comp1p-8	135	347	-3.649e+04	3.230e+03	1.383e+05	7	1
pack-comp2-8	38	82	7.724e−01	2.677e+01	1.039e+04	7	1
pack-comp2c-8	150	309	6.537e−01	6.595e+00	2.979e+04	0	1
pack-rig1-8	150	1109	6.623e−01	6.294e+00	1.562e+03	0	1
pack-rig1c-8	61	174	6.013e−01	5.803e+00	4.770e+03	7	1
pack-rig1p-8	150	948	-4.048e+01	1.621e+01	4.220e+03	0	0
pack-rig2-8	150	307	7.804e−01	8.259e−09	9.463e−04	0	1
pack-rig2c-8	75	289	6.046e−01	5.751e+00	4.974e+03	7	0
pack-rig2p-8	147	403	-1.573e+02	1.093e+00	2.086e+02	7	1
portfl-i-1	28	59	2.096e−06	4.971e−04	5.158e−04	1	1
portfl-i-2	30	61	1.099e−06	8.256e−03	2.070e−03	1	1
portfl-i-3	31	64	1.743e−06	3.498e−02	1.864e−04	1	1
portfl-i-4	31	64	2.755e−06	1.418e−02	4.518e−04	1	1
portfl-i-6	28	58	2.394e−06	3.893e−02	4.654e−04	1	1
qpec-100-1	80	163	9.900e−02	1.762e+01	7.324e−06	1	1
qpec1	10	21	8.000e+01	3.044e−07	5.138e−07	1	1
qpec2	150	303	4.500e+01	9.669e+04	2.425e−02	0	1
ralph1	150	303	-1.563e−05	3.191e+04	1.885e−03	0	1
ralph2	15	31	-2.228e−07	2.001e+00	3.071e−07	1	1
ralphmod	75	151	-5.726e+02	8.219e+02	1.167e+02	7	0

Table 6.1
Cont'd.

Problem	iter	nfe	f	$\|v_c^*\|$	$\|r\|$	flag1	flag2
scholtes1	10	21	2.000e+00	1.008e−08	2.302e−08	1	1
scholtes2	21	43	1.500e+01	6.894e−06	2.881e−06	1	1
scholtes3	8	18	5.000e−01	1.000e+00	5.044e−07	1	1
scholtes4	150	301	-4.994e−05	3.994e+04	3.895e−03	0	1
scholtes5	8	17	1.000e+00	1.870e+00	1.277e−06	1	1
sl1	30	61	1.003e−04	3.337e−07	1.715e−05	1	1
stackelberg1	12	25	-3.267e+03	8.998e−01	5.536e−06	1	1
tap-09	106	320	1.546e+02	1.964e−01	8.807e−04	1	0
tap-15	136	300	3.131e+02	2.664e−01	3.389e−02	7	1

Table 6.2
Exit flags in Table 6.1. *The second exit flag indicates when the final relaxation parameters* $(\delta_c^*, \delta_1^*, \delta_2^*)$ *satisfy the complementarity condition given by* (3.9).

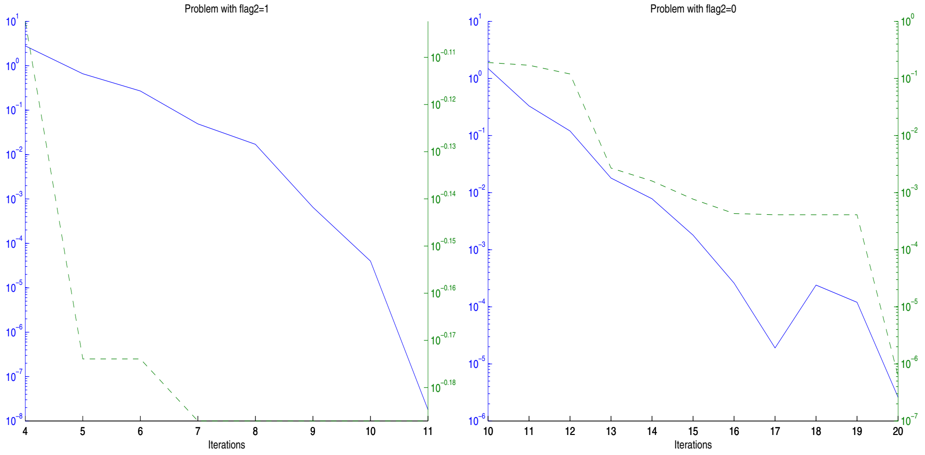
flag1	Status
0	Terminated by iteration limit (150)
1	Found stationary point of (MPEC-δ^*) and strongly stationary point of (MPEC)
2	Found stationary point of (MPEC-δ^*) but not strongly stationary point of (MPEC)
7	Terminated because steplength too small ($\alpha_k < 10^{-12}$) or descent direction not found

flag2	Status
0	$\max(\delta_c^*, \delta_i^*) = 0$
1	$\max(\delta_c^*, \delta_i^*) > 0$

In addition, we have observed that the algorithm is particularly efficient on those problems for which the iterates converge to a strongly stationary point that satisfies the MPEC-WSCS and -SSOSC. For these problems, in particular, the final relaxation parameter satisfy $\max\left(\delta_c^*, \min(\delta_1^*, \delta_2^*)\right) > 0$ and the iterates converge at a superlinear rate. On the other hand, for those problems for which the algorithm converges to a strongly stationary point that does not satisfy the MPEC-WSCS and -SSOSC, there is a zero or very small component of $\max\left(\delta_c^*, \min(\delta_1^*, \delta_2^*)\right)$, and the iterates converge only at a linear rate. In other words, when $\max\left(\delta_c^*, \min(\delta_1^*, \delta_2^*)\right) > 0$ (i.e., flag2 = 1), the condition number of the KKT matrix remains bounded and the algorithm converges superlinearly. On the other hand, when $\max\left(\delta_c^*, \min(\delta_1^*, \delta_2^*)\right) = 0$ (i.e., flag2 = 0), the condition number of the KKT matrix grows large, and the algorithm converges only linearly.

This behavior can be observed in Figure 6.1, which depicts the evolution of $\|r_k^*\|$ and the minimum value of the vector $\max\left(\delta_c^*, \min(\delta_1^*, \delta_2^*)\right)$ for two problems of the MacMPEC collection. Both vertical axes are in a logarithmic (base 10) scale. The first subfigure shows the last eight iterates generated by the algorithm for problem ex9.2.4 (which confirms $\max\left(\delta_c^*, \min(\delta_1^*, \delta_2^*)\right) > 0$). The second subfigure shows the last 11 iterates generated by the algorithm for problem ex9.2.7 (which confirms a numerically zero component of $\max\left(\delta_c^*, \min(\delta_1^*, \delta_2^*)\right)$.

Moreover, the numerical results confirm the relevance of our relaxing the MPEC-SCS assumption in our analysis. In particular, there are eight problems (approximately 10% of the total) for which the MPEC-SCS does not hold at the minimizer (although MPEC-WSCS and -SSOSC hold) and yet $\max\left(\delta_c^*, \min(\delta_1^*, \delta_2^*)\right) > 0$ in the limit. Likewise, we have confirmed that for all problems for which the minimum value of the vector $\max\left(\delta_c^*, \min(\delta_1^*, \delta_2^*)\right)$ is zero, the algorithm converges to points where the MPEC-WSCS or -SSOSC do not hold.



(a) Problem ex9.2.4

(b) Problem ex9.2.7

FIG. 6.1. *Final iterations of two problems from the MacMPEC test suite. Each graph shows the KKT residual* $\|r_k^*\|$ *(solid line and left axis) and the minimum value of the vector* $\max\left(\delta_c^*, \min(\delta_1^*, \delta_2^*)\right)$ *(dashed line and right axis) against the iteration count.*

Acknowledgments. We are grateful to Sven Leyffer and to three anonymous referees for their detailed feedback and suggestions. We are also grateful to Sven Leyffer for his effort in assembling and maintaining the excellent MacMPEC test-problem set. Many people have given us helpful advice along the way, but we would like to especially thank Mihai Anitescu, Jorge Nocedal, and Danny Ralph for their valuable input and encouragement.

REFERENCES

[1] M. Anitescu, *On Solving Mathematical Programs with Complementarity Constraints as Nonlinear Programs*, Tech. Report ANL/MCS-P864-1200, Argonne National Laboratory, Argonne, IL, 2000.

[2] M. Anitescu, *Global Convergence of an Elastic Mode Approach for a Class of Mathematical Programs with Complementarity Constraints*, Tech. Report ANL/MCS-P1143-0404, Argonne National Laboratory, Argonne, IL, 2004.

[3] H. Y. Benson, A. Sen, D. F. Shanno, and R. J. Vanderbei, *Interior-Point Algorithms, Penalty Methods and Equilibrium Problems*, Tech. Report ORFE-03-02, Operations Research and Financial Engineering, Princeton University, Princeton, NJ, 2003.

[4] H. Y. Benson, D. F. Shanno, and R. J. Vanderbei, *Interior-Point Methods for Nonconvex Nonlinear Programming: Complementarity Constraints*, Tech. Report ORFE-02-02, Operations Research and Financial Engineering, Princeton University, Princeton, NJ, 2002.

[5] M. C. Ferris and J. S. Pang, *Engineering and economic applications of complementarity problems*, SIAM Rev., 39 (1997), pp. 669–713.

[6] R. Fletcher, S. Leyffer, D. Ralph, and S. Scholtes, *Local Convergence of SQP Methods for Mathematical Programs with Equilibrium Constraints*, Tech. Report NA-209, University of Dundee, Dundee, Scotland, UK, 2002.

[7] R. Fletcher and S. Leyffer, *Numerical Experience with Solving MPECs as NLPs*, Tech. Report NA-210, University of Dundee, Dundee, Scotland, UK, August, 2002.

[8] M. P. Friedlander and M. A. Saunders, *A globally convergent linearly constrained Lagrangian method for nonlinear optimization*, SIAM J. Optim., 15 (2004), pp. 863–897.

[9] X. Hu and D. Ralph, *Convergence of a penalty method for mathematical programming with complementarity constraints*, J. Optim. Theory Appl., 123 (2004), pp. 365–390.

[10] S. Leyffer, G. Lopez-Calva, and J. Nocedal, *Interior Methods for Mathematical Programs with Complementarity Constraints*, Tech. Report ANL/MCS-P-1211-1204, Argonne National Laboratory, Argonne, IL, 2004.

[11] S. Leyffer, *MacMPEC: AMPL collection of MPECs*. http://www.mcs.anl.gov/~leyffer/MacMPEC (April 2004).

[12] X. Liu and J. Sun, *Generalized Stationary Points and an Interior Point Method for Mathematical Programs with Equilibrium Constraints*, Tech. Report, National University of Singapore, Singapore, 2001.

[13] Z. Luo, J. Pang, and D. Ralph, *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press, London, 1996.

[14] J. Moguerza and F. Prieto, *An augmented lagrangian interior-point method using directions of negative curvature*, Math. Program. A, 95 (2003), pp. 573–616.

[15] B. A. Murtagh and M. A. Saunders, *MINOS* 5.5 *User's Guide*, Tech. Report 83-20R, Systems Optimization Laboratory, Department of Management Science and Engineering, Stanford University, Stanford, CA, December, 1983.

[16] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.

[17] J. Outrata, M. Kocvara, and J. Zowe, *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints: Theory, Applications, and Numerical Results*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[18] A. U. Raghunathan and L. T. Biegler, *Interior Point Methods for Mathematical Programs with Complementarity Constraints*, Tech. Report, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA, 2003.

[19] D. Ralph and S. J. Wright, *Some Properties of Regularization Schemes for MPECs*, Tech. Report 03-04, Computer Sciences, University of Wisconsin, Madison, Wisconsin, December, 2003.

[20] H. Scheel and S. Scholtes, *Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity*, Math. Operat. Res., 25 (2000), pp. 1–22.

[21] S. Scholtes and M. Stöhr, *Exact penalization of mathematical programs with equilibrium constraints*, SIAM J. Control and Optim., 37 (1999), pp. 617–652.

[22] S. Scholtes, *Convergence properties of a regularization scheme for mathematical programs with complementarity constraints*, SIAM J. Optim., 11 (2001), pp. 918–936.

[23] D. Shanno and R. Vanderbei, *An interior-point algorithm for nonconvex nonlinear programming*, Comput. Optim. Appl., 13 (1999), pp. 231–252.

[24] H. Yamashita and H. Yabe, *Superlinear and quadratic convergence of some primal-dual interior point methods for constrained optimization*, Math. Program., 75 (1996), pp. 377–397.

SIAM J. OPTIM.
Vol. 16, No. 2, pp. 610–628

SUM OF SQUARES APPROXIMATION OF POLYNOMIALS, NONNEGATIVE ON A REAL ALGEBRAIC SET*

JEAN B. LASSERRE[†]

Abstract. Wih every real polynomial f, we associate a family $\{f_{\epsilon r}\}_{\epsilon,r}$ of real polynomials, in explicit form in terms of f and the parameters $\epsilon > 0, r \in \mathbb{N}$, and such that $\|f - f_{\epsilon r}\|_1 \to 0$ as $\epsilon \to 0$.

Let $V \subset \mathbb{R}^n$ be a real algebraic set described by finitely many polynomials equations $g_j(x) = 0, j \in J$, and let f be a real polynomial, nonnegative on V. We show that for every $\epsilon > 0$, there exist nonnegative scalars $\{\lambda_j(\epsilon)\}_{j\in J}$ such that, for all r sufficiently large,

$$f_{\epsilon r} + \sum_{j\in J} \lambda_j(\epsilon)\, g_j^2 \quad \text{is a sum of squares.}$$

This representation is an obvious certificate of nonnegativity of $f_{\epsilon r}$ on V, and very specific in terms of the g_j that define the set V. In particular, it is valid with no assumption on V. In addition, this representation is also useful from a computation point of view, as we can define semidefinite programming relaxations to approximate the global minimum of f on a real algebraic set V, or a semialgebraic set $\mathbb{K}$, and again, with no assumption on V or $\mathbb{K}$.

Key words. real algebraic geometry, positive polynomials, sum of squares, semidefinite programming

AMS subject classifications. 11E25, 12D15, 13P05, 12Y05, 90C22, 90C25

DOI. 10.1137/040614141

1. Introduction.

Let $V \subset \mathbb{R}^n$ be the real algebraic set

$$V := \{x \in \mathbb{R}^n \mid \quad g_j(x) = 0, \quad j = 1, \dots, m\} \tag{1.1}$$

for some family of real polynomials $\{g_j\} \subset \mathbb{R}[x](= \mathbb{R}[x_1, \dots, x_n])$.

The main motivation for this paper is to provide a characterization of polynomials $f \in \mathbb{R}[x]$, nonnegative on V, in terms of a certificate of positivity. In addition, and in view of the many potential applications, one would like to obtain a representation that is also useful from a computational point of view.

In some particular cases, when V is compact, and viewing the equations $g_j(x) = 0$ as two opposite inequations $g_j(x) \geq 0$ and $g_j(x) \leq 0$, one may obtain Schmüdgen's sum of squares (s.o.s.) representation [20] for $f + \epsilon$ ($\epsilon > 0$), instead of f. The latter representation may be even refined to become Putinar [16] and Jacobi and Prestel [6] s.o.s. representation, that is, $f + \epsilon$ can be written

$$f + \epsilon \,=\, f_0 + \sum_{j=1}^{m} f_j\, g_j, \tag{1.2}$$

for some polynomials $\{f_j\} \subset \mathbb{R}[x]$, with f_0 a s.o.s. Hence, if f is nonnegative on V, every approximation $f + \epsilon$ of f (with $\epsilon > 0$) has the representation (1.2). The interested reader is referred to Marshall [12], Prestel and Delzell [15], and Scheiderer [18, 19] for a nice account of such results.

*Received by the editors August 30, 2004; accepted for publication (in revised form) August 6, 2005; published electronically December 30, 2005.
http://www.siam.org/journals/siopt/16-2/61414.html

[†]LAAS-CNRS and Institute of Mathematics, LAAS 7 Avenue du Colonel Roche, 31077 Toulouse Cédex 4, France (lasserre@laas.fr).

Contribution. We prove the following result. Let $\|f\|_1 = \sum_\alpha |f_\alpha|$ whenever $x \mapsto f(x) = \sum_\alpha f_\alpha x^\alpha$. Let $f \in \mathbb{R}[x]$ be nonnegative on V, as defined in (1.1), and let $F := \{f_{\epsilon r}\}_{\epsilon,r}$ be the family of polynomials

$$f_{\epsilon r} \,=\, f + \epsilon \sum_{k=0}^{r} \sum_{i=1}^{n} \frac{x_i^{2k}}{k!}, \qquad \epsilon \geq 0, \quad r \in \mathbb{N}. \tag{1.3}$$

(So, for every $r \in \mathbb{N}$, $\|f - f_{\epsilon r}\|_1 \to 0$ as $\epsilon \downarrow 0$.)

Then, for every $\epsilon > 0$, there exist nonnegative scalars $\{\lambda_j(\epsilon)\}_{j=1}^m$, such that for all r sufficiently large (say, $r \geq r(\epsilon)$),

$$f_{\epsilon r} \,=\, q_\epsilon - \sum_{j=1}^{m} \lambda_j(\epsilon)\, g_j^2, \tag{1.4}$$

for some s.o.s. polynomial $q_\epsilon \in \mathbb{R}[x]$, that is, $f_{\epsilon r} + \sum_{j=1}^m \lambda_j(\epsilon) g_j^2$ is s.o.s.

Thus, with no assumption on the set V, one obtains a representation of $f_{\epsilon r}$ (which is positive on V as $f_{\epsilon r} > f$ for all $\epsilon > 0$) in the simple and explicit form (1.4), an obvious certificate of positivity of $f_{\epsilon r}$ on V. In particular, when $V \equiv \mathbb{R}^n$, one retrieves the result of [11], which states that every nonnegative real polynomial f can be aproximated as closely as desired, by a family of s.o.s. polynomials $\{f_{\epsilon r(\epsilon)}\}_\epsilon$, with $f_{\epsilon r}$ as in (1.3).

Notice that $f + n\epsilon = f_{\epsilon 0}$. So, on the one hand, the approximation $f_{\epsilon r}$ in (1.4) is more complicated than $f + \epsilon$ in (1.2), valid for the compact case but on the other hand, the coefficients of the g_j in (1.4) are now scalars instead of s.o.s., and (1.4) is valid for an arbitrary algebraic set V.

The case of a semialgebraic set $\mathbb{K} = \{x \in \mathbb{R}^n | g_j(x) \geq 0,\ j = 1, \ldots, m\}$ reduces to the case of an algebraic set $V \in \mathbb{R}^{n+m}$ by introducing m slack variables $\{z_j\}$ and replacing $g_j(x) \geq 0$ with $g_j(x) - z_j^2 = 0$ for all $j = 1, \ldots, m$. Let $f \in \mathbb{R}[x]$ be nonnegative on $\mathbb{K}$. Then, for every $\epsilon > 0$, there exist nonnegative scalars $\{\lambda_j(\epsilon)\}_{j=1}^m$ such that for all sufficiently large r,

$$f + \epsilon \sum_{k=0}^{r} \left[\sum_{i=1}^{n} \frac{x_i^{2k}}{k!} + \sum_{j=1}^{m} \frac{z_j^{2k}}{k!} \right] \,=\, q_\epsilon - \sum_{j=1}^{m} \lambda_j(\epsilon)\, (g_j - z_j^2)^2 \tag{1.5}$$

for some s.o.s. $q_\epsilon \in \mathbb{R}[x, z]$. Equivalently, everywhere on $\mathbb{K}$, the polynomial

$$x \mapsto f(x) + \epsilon \sum_{k=0}^{r} \sum_{i=1}^{n} \frac{x_i^{2k}}{k!} + \epsilon \sum_{k=0}^{r} \sum_{j=1}^{m} \frac{g_j(x)^k}{k!}$$

coincides with the polynomial $x \mapsto q_\epsilon(x_1, \ldots, x_n, \sqrt{g_1(x)}, \ldots, \sqrt{g_m(x)})$, obviously nonnegative. Indeed, if q_ϵ is a s.o.s. of polynomials in $\mathbb{R}[x, z]$, from (1.5), it is also a polynomial in $\mathbb{R}[x, z^2]$, and so, on $\mathbb{K}$, $x \mapsto q_\epsilon(x_1, \ldots, x_n, \sqrt{g_1(x)}, \ldots, \sqrt{g_m(x)})$ is a nonnegative polynomial in $\mathbb{R}[x]$, as $z_j^2 = g_j(x)$ for all $j = 1, \ldots, m$.

The representation (1.4) is also useful for computational purposes. Indeed, using (1.4), one can approximate the global minimum of f on V, by solving a sequence of semidefinite programming (SDP) problems. The same applies to an arbitrary semialgebraic set $\mathbb{K} \subset \mathbb{R}^n$, defined by m polynomials inequalities, as explained above. Again, and in contrast to previous SDP-relaxation techniques as in, e.g., [8, 9, 10, 14, 21], no compacity assumption on V or $\mathbb{K}$ is required.

In a sense, the family $F = \{f_{\epsilon r}\} \subset \mathbb{R}[x]$ (with $f_{0r} \equiv f$) is a set of regularizations of f, because one may approximate f by members of F, and those members always have nice representations when f is nonnegative on an algebraic set V (including the case $V \equiv \mathbb{R}^n$), whereas f itself might not have such a nice representation.

Methodology. To prove our main result, we proceed in three main steps.

1. We first define an infinite dimensional linear programming problem on an appropriate space of measures, whose optimal value is the global minimum of f on the set V.

2. We then prove a crucial result, namely, that there is no duality gap between this linear programming problem and its dual. The approach is similar to but different from that taken in [11] when $V \equiv \mathbb{R}^n$. Indeed, the approach in [11] does not work when $V \not\equiv \mathbb{R}^n$. Here, we use the important fact that the polynomial θ_r is a moment function. And so, if a set of probability measures Π satisfies $\sup_{\mu\in\Pi} \int \theta_r d\mu < \infty$, it is tight and therefore, by Prohorov's theorem, relatively compact. This latter intermediate result is crucial for our purpose.

3. In the final step, we use our recent result [11], which states that if a polynomial $h \in \mathbb{R}[x]$ is nonnegative on $\mathbb{R}^n$, then $h + \epsilon\theta_r$ ($\epsilon > 0$) is a sum of squares, provided that r is sufficiently large.

The paper in organized as follows. Notation and definitions are introduced in section 2, and some preliminary results are stated in section 3, whereas our main result is stated and discussed in section 4. For clarity of exposition, most proofs are postponed to section 5, and some auxiliary results are stated in an appendix; in particular, duality results for linear programming in infinite dimensional spaces are briefly reviewed.

2. Notation and definitions.

Let $\mathbb{R}_+ \subset \mathbb{R}$ denote the cone of nonnegative real numbers. For a real symmetric matrix A, the notation $A \succeq 0$ (resp., $A \succ 0$) stands for A positive semidefinite (resp., positive definite). The sup-norm $\sup_j |x_j|$ of a vector $x \in \mathbb{R}^n$ is denoted by $\|x\|_\infty$. Let $\mathbb{R}[x]$ be the ring of real polynomials, and let

$$v_r(x) := (1, x_1, x_2, \ldots x_n, x_1^2, x_1x_2, \ldots, x_1x_n, x_2^2, x_2x_3, \ldots, x_n^r) \tag{2.1}$$

be the canonical basis for the $\mathbb{R}$-vector space $\mathcal{A}_r$ of real polynomials of degree at most r, and let $s(r)$ be its dimension. Similarly, $v_\infty(x)$ denotes the canonical basis of $\mathbb{R}[x]$ as a $\mathbb{R}$-vector space, denoted $\mathcal{A}$. So a vector in $\mathcal{A}$ has always finitely many nonzero coefficients.

Therefore, a polynomial $p \in \mathcal{A}_r$ is written

$$x \mapsto p(x) = \sum_\alpha p_\alpha x^\alpha = \langle \mathbf{p}, v_r(x)\rangle, \qquad x \in \mathbb{R}^n$$

(where $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$) for some vector $\mathbf{p} = \{p_\alpha\} \in \mathbb{R}^{s(r)}$, the vector of coefficients of p in the basis (2.1).

Extending $\mathbf{p}$ with zeros, we can also consider $\mathbf{p}$ as a vector indexed in the basis $v_\infty(x)$ (i.e., $\mathbf{p} \in \mathcal{A}$). If we equip $\mathcal{A}$ with the usual scalar product $\langle ., . \rangle$ of vectors, then for every $p \in \mathcal{A}$,

$$p(x) = \sum_{\alpha\in\mathbb{N}^n} p_\alpha x^\alpha = \langle \mathbf{p}, v_\infty(x)\rangle, \qquad x \in \mathbb{R}^n.$$

Given a sequence $\mathbf{y} = \{y_\alpha\}$ indexed in the basis $v_\infty(x)$, let $L_\mathbf{y} : \mathcal{A} \to \mathbb{R}$ be the linear functional

$$p \mapsto L_\mathbf{y}(p) := \sum_{\alpha\in\mathbb{N}^n} p_\alpha y_\alpha = \langle \mathbf{p}, \mathbf{y}\rangle. \tag{2.2}$$

Given a sequence $\mathbf{y} = \{y_\alpha\}$ indexed in the basis $v_\infty(x)$, the moment matrix $M_r(\mathbf{y}) \in \mathbb{R}^{s(r)\times s(r)}$ with rows and columns indexed in the basis $v_r(x)$ in (2.1) satisfies

$$[M_r(\mathbf{y})(1,j) = y_\alpha \text{ and } M_r(y)(i,1) = y_\beta] \Rightarrow M_r(y)(i,j) = y_{\alpha+\beta}. \tag{2.3}$$

For instance, with $n = 2$,

$$M_2(\mathbf{y}) = \left[\begin{array}{cccccc} y_{00} & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\ y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\ y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} \end{array}\right].$$

A sequence $\mathbf{y} = \{y_\alpha\}$ has a representing (positive) measure $\mu_\mathbf{y}$ if

$$y_\alpha = \int_{\mathbb{R}^n} x^\alpha \, d\mu_\mathbf{y} \qquad \forall\, \alpha \in \mathbb{N}^n. \tag{2.4}$$

In this case one also says that $\mathbf{y}$ is a moment sequence. In addition, if $\mu_\mathbf{y}$ is unique, then $\mathbf{y}$ is said to be a determinate moment sequence, and the representing measure $\mu_\mathbf{y}$ is said to be determinate (and indeterminate otherwise).

A very useful criterion for existence of a determinate representing measure is the generalized (or multidimensional) Carleman's condition (2.5) below, due to Nussbaum [13]:

$$\sum_{k=1}^{\infty} L_\mathbf{y}(x_i^{2k})^{-1/2k} = +\infty, \qquad i = 1, \ldots, n. \tag{2.5}$$

Indeed, let $\mathbf{y} = \{y_\alpha\}$ be a sequence indexed in the basis $v_\infty(x)$. If (2.5) holds, then $\mathbf{y}$ is a determinate moment sequence; see, e.g., Berg [3] and Berg [4, Theorem 5].

The matrix $M_r(\mathbf{y})$ defines a bilinear form $\langle ., .\rangle_\mathbf{y}$ on $\mathcal{A}_r$, by

$$\langle q, p\rangle_\mathbf{y} := \langle \mathbf{q}, M_r(\mathbf{y})\mathbf{p}\rangle = L_\mathbf{y}(qp), \quad q, p \in \mathcal{A}_r,$$

and if $\mathbf{y}$ has a representing measure $\mu_\mathbf{y}$, then

$$L_\mathbf{y}(q^2) = \langle \mathbf{q}, M_r(\mathbf{y})\mathbf{q}\rangle = \int_{\mathbb{R}^n} q(x)^2 \, \mu_\mathbf{y}(dx) \geq 0 \quad \forall\, q \in \mathcal{A}_r, \tag{2.6}$$

so that $M_r(\mathbf{y})$ is positive semidefinite, i.e., $M_r(\mathbf{y}) \succeq 0$.

3. Preliminaries. Let $V \subset \mathbb{R}^n$ be the real algebraic set defined in (1.1), and let B_M be the closed ball

$$B_M = \{x \in \mathbb{R}^n \mid \ \|x\|_\infty \leq M\}. \tag{3.1}$$

PROPOSITION 3.1. *Let $f \in \mathbb{R}[x]$ be such that $-\infty < f^* := \inf_{x \in V} f(x)$. Then, for every $\epsilon > 0$, there is some $M_\epsilon \in \mathbb{N}$ such that*

$$f_M^* := \inf \{ f(x) \mid x \in B_M \cap V \} < f^* + \epsilon \qquad \forall M \geq M_\epsilon.$$

Equivalently, $f_M^* \downarrow f^*$ as $M \to \infty$.

Proof. Suppose it is false. That is, there is some $\epsilon_0 > 0$ and an infinite sequence $\{M_k\} \subset \mathbb{N}$, with $M_k \to \infty$, such that $f_{M_k}^* \geq f^* + \epsilon_0$ for all k. But let $x_0 \in V$ be such that $f(x_0) < f^* + \epsilon_0$. With any $M_k \geq \|x_0\|_\infty$, one obtains the contradiction $f^* + \epsilon_0 \leq f_{M_k}^* \leq f(x_0) < f^* + \epsilon_0$. □

For every $r \in \mathbb{N}$, let $\theta_r \in \mathbb{R}[x]$ be the polynomial

$$x \mapsto \theta_r(x) := \sum_{k=0}^{r} \sum_{i=1}^{n} \frac{x_i^{2k}}{k!}, \qquad x \in \mathbb{R}^n, \tag{3.2}$$

and notice that $n \leq \theta_r(x) \leq \sum_{i=1}^n \mathrm{e}^{x_i^2} =: \theta_\infty(x)$ for all $x \in \mathbb{R}^n$. Moreover, θ_r is a moment function, as it satisfies

$$\lim_{M \to \infty} \inf_{x \in B_M^c} \theta_r(x) = +\infty, \tag{3.3}$$

where B_M^c denotes the complement of B_M in $\mathbb{R}^n$; see section 6.2.

Next, with V as in (1.1) we introduce the following optimization problems:

$$\mathbb{P} : \qquad f^* := \inf_{x \in V} f(x), \tag{3.4}$$

and for $0 < M \in \mathbb{N}$, $r \in \mathbb{N} \cup \{\infty\}$,

$$\mathcal{P}_M^r : \begin{cases} & \inf_\mu \int f \, d\mu \\ \text{s.t.} & \int g_j^2 \, d\mu \quad \leq 0, \quad j = 1, \dots, m, \\ & \int \theta_r \, d\mu \quad \leq n\mathrm{e}^{M^2}, \\ & \mu \in \mathcal{P}(\mathbb{R}^n), \end{cases} \tag{3.5}$$

where $\mathcal{P}(\mathbb{R}^n)$ is the space of probability measures on $\mathbb{R}^n$ (with $\mathcal{B}$ its associated Borel σ-algebra). The respective optimal values of $\mathbb{P}$ and $\mathcal{P}_M^r$ are denoted $\inf \mathbb{P} = f^*$ and $\inf \mathcal{P}_M^r$, or $\min \mathbb{P}$ and $\min \mathcal{P}_M^r$ if the minimum is attained (in which case, the problem is said to be solvable).

PROPOSITION 3.2. *Let $f \in \mathbb{R}[x]$, and let $\mathbb{P}$ and $\mathcal{P}_M^r$ be as in* (3.4) *and* (3.5)*, respectively. Assume that $f^* > -\infty$. Then, for every $r \in \mathbb{N} \cup \{\infty\}$, $\inf \mathcal{P}_M^r \downarrow f^*$ as $M \to \infty$. If f has a global minimizer $x^* \in V$, then $\min \mathcal{P}_M^r = f^*$ whenever $M \geq \|x^*\|_\infty$.*

Proof. When M is sufficiently large, $B_M \cap V \neq \emptyset$, and so $\mathcal{P}_M^r$ is consistent, and $\inf \mathcal{P}_M^r < \infty$. Let $\mu \in \mathcal{P}(\mathbb{R}^n)$ be admissible for $\mathcal{P}_M^r$. From $\int g_j^2 \, d\mu \leq 0$ for all $j = 1, \dots, m$, it follows that $g_j(x)^2 = 0$ for μ-almost all $x \in \mathbb{R}^n$, $j = 1, \dots, m$. That is, for every $j = 1, \dots, m$, there exists a set $A_j \in \mathcal{B}$ such that $\mu(A_j^c) = 0$ and $g_j(x) = 0$ for all $x \in A_j$. Take $A = \cap_j A_j \in \mathcal{B}$ so that $\mu(A^c) = 0$, and for all $x \in A$, $g_j(x) = 0$ for all $j = 1, \dots, m$. Therefore, $A \subset V$, and as $\mu(A^c) = 0$,

$$\int_{\mathbb{R}^n} f \, d\mu = \int_A f \, d\mu \geq f^* \quad \text{because } f \geq f^* \text{ on } A \subset V,$$

which proves $\inf \mathcal{P}_M^r \geq f^*$.

As V is closed and B_M is closed and bounded, the set $B_M \cap V$ is compact and so, with f_M^* as in Proposition 3.1, there is some $\hat{x} \in B_M \cap V$ such that $f(\hat{x}) = f_M^*$. In addition let $\mu \in \mathcal{P}(\mathbb{R}^n)$ be the Dirac probability measure at the point $\hat{x}$. As $\|\hat{x}\|_\infty \leq M$,

$$\int \theta_r \, d\mu \;=\; \theta_r(\hat{x}) \;\leq\; n\mathrm{e}^{M^2}.$$

Moreover, as $\hat{x} \in V$, $g_j(\hat{x}) = 0$, for all $j = 1, \ldots, m$, and so

$$\int g_j^2 \, d\mu \;=\; g_j(\hat{x})^2 \;=\; 0, \quad j = 1, \ldots, m,$$

so that μ is an admissible solution of $\mathcal{P}_M^r$ with value $\int f \, d\mu = f(\hat{x}) = f_M^*$, which proves that $\inf \mathcal{P}_M^r \leq f_M^*$. This latter fact, combined with Proposition 3.1 and with $f^* \leq \inf \mathcal{P}_M^r$, implies $\inf \mathcal{P}_M^r \downarrow f^*$ as $M \to \infty$, the desired result. The final statement is immediate by taking as feasible solution for $\mathcal{P}_M^r$, the Dirac probability measure at the point $x^* \in B_M \cap V$ (with $M \geq \|x^*\|_\infty$). As its value is now f^*, it is also optimal, and so, $\mathcal{P}_M^r$ is solvable with optimal value $\min \mathcal{P}_M^r = f^*$. □

Consider now the following optimization problem $\mathcal{Q}_M^r$, the dual problem of $\mathcal{P}_M^r$, i.e.,

$$(3.6) \qquad \mathcal{Q}_M^r: \quad \begin{array}{ll} & \max\limits_{\lambda,\delta,\gamma} \ \gamma - n\delta\mathrm{e}^{M^2} \\ \text{s.t.} & f + \delta\theta_r + \sum_{j=1}^m \lambda_j g_j^2 \qquad \geq \gamma, \\ & \gamma \in \mathbb{R}, \delta \in \mathbb{R}_+, \lambda \in \mathbb{R}_+^m, \end{array}$$

with optimal value denoted by $\sup \mathcal{Q}_M^r$. Indeed, $\mathcal{Q}_M^r$ is a dual of $\mathcal{P}_M^r$ because weak duality holds. To see this, consider any two feasible solutions $\mu \in \mathcal{P}(\mathbb{R}^n)$ and $(\lambda, \delta, \gamma) \in \mathbb{R}_+^m \times \mathbb{R}_+ \times \mathbb{R}$, of $\mathcal{P}_M^r$ and $\mathcal{Q}_M^r$, respectively. Then, integrating both sides of the inequality in $\mathcal{Q}_M^r$ with respect to μ yields

$$\int f d\mu + \delta \int \theta_r \, d\mu + \sum_{j=1}^m \lambda_j \int g_j^2 \, d\mu \geq \gamma,$$

and so, using that μ is feasible for $\mathcal{P}_M^r$,

$$\int f d\mu \;\geq\; \gamma - \delta n\mathrm{e}^{M^2}.$$

Hence, the value of any feasible solution of $\mathcal{Q}_M^r$ is always smaller than the value of any feasible solution of $\mathcal{P}_M^r$, i.e., weak duality holds.

In fact we can get the more important and crucial following result.

THEOREM 3.3. *Let M be large enough so that $B_M \cap V \neq \emptyset$. Let $f \in \mathbb{R}[x]$, and let $r_0 > \max[\deg f, \deg g_j]$. Then, for every $r \geq r_0$, $\mathcal{P}_M^r$ is solvable, and there is no duality gap between $\mathcal{P}_M^r$ and its dual $\mathcal{Q}_M^r$. That is,* $\sup \mathcal{Q}_M^r = \min \mathcal{P}_M^r$.

For a proof see section 5.1. We finally end up this section by restating a result proved in [11], which, together with Theorem 3.3, will be crucial to prove our main result.

THEOREM 3.4 (see [11]). *Let $f \in \mathbb{R}[x]$ be nonnegative. Then for every $\epsilon > 0$, there is some $r(\epsilon) \in \mathbb{N}$ such that*

$$(3.7) \qquad f_{\epsilon r(\epsilon)} \; (= f + \epsilon\theta_{r(\epsilon)}) \qquad \textit{is a sum of squares,}$$

and so is $f_{\epsilon r}$ for all $r \geq r(\epsilon)$.

4. Main result. Recall that for given $(\epsilon, r) \in \mathbb{R}\times\mathbb{N}$, $f_{\epsilon r} = f+\epsilon\theta_r$, with $\theta_r \in \mathbb{R}[x]$ being the polynomial defined in (3.2). We now state our main result.

THEOREM 4.1. *Let $V \subset \mathbb{R}^n$ be as in* (1.1)*, and let $f \in \mathbb{R}[x]$ be nonnegative on V. Then for every $\epsilon > 0$, there exists $r(\epsilon) \in \mathbb{N}$ and nonnegative scalars $\{\lambda_j\}_{j=1}^m$ such that for all $r \geq r(\epsilon)$,*

$$f_{\epsilon r} \,=\, q - \sum_{j=1}^m \lambda_j \, g_j^2 \tag{4.1}$$

for some s.o.s. polynomial $q \in \mathbb{R}[x]$. In addition, $\|f - f_{\epsilon r}\|_1 \to 0$, as $\epsilon \downarrow 0$.

For a proof see section 5.2.

Remark 4.2. (i) Observe that (4.1) is an obvious certificate of positivity of $f_{\epsilon r}$ on the algebraic set V, because everywhere on V, $f_{\epsilon r}$ coincides with the s.o.s. polynomial q. Therefore, when f is nonnegative on V, one obtains with no assumption on the algebraic set V, a certificate of positivity for any approximation $f_{\epsilon r}$ of f (with $r \geq r(\epsilon)$), whereas f itself might not have such a representation. In other words, the (ϵ, r)-perturbation $f_{\epsilon r}$ of f has a regularization effect on f as it permits to derive nice representations.

(ii) From the proof of Theorem 4.1, instead of the representation (4.1), one may also provide the alternative representation

$$f_{\epsilon r} \,=\, q - \lambda \sum_{j=1}^m g_j^2$$

for some s.o.s. polynomial q and some (single) nonnegative scalar λ (instead of m nonnegative scalars in (4.1)).

4.1. The case of a semialgebraic set. We now consider the representation of polynomials, nonnegative on a semialgebraic set $\mathbb{K} \subset \mathbb{R}^n$, defined as

$$\mathbb{K} \,:=\, \{x \in \mathbb{R}^n \mid \quad g_j(x) \,\geq\, 0, \quad j = 1, \ldots, m\} \tag{4.2}$$

for some family $\{g_j\}_{j=1}^m \subset \mathbb{R}[x]$.

One may apply the machinery developed previously for algebraic sets, because the semialgebraic set $\mathbb{K}$ may be viewed as the projection on $\mathbb{R}^n$ of an algebraic set in $\mathbb{R}^{n+m}$. Indeed, let $V \subset \mathbb{R}^{n+m}$ be the algebraic set defined as

$$V \,:=\, \{(x, z) \in \mathbb{R}^n \times \mathbb{R}^m \mid \quad g_j(x) - z_j^2 \,=\, 0, \quad j = 1, \ldots, m\}. \tag{4.3}$$

Then every $x \in \mathbb{K}$ is associated with the point $(x, \sqrt{g_1(x)}, \ldots, \sqrt{g_m(x)}) \in V$.

Let $\mathbb{R}[z] := \mathbb{R}[z_1, \ldots, z_m]$, and $\mathbb{R}[x, z] := \mathbb{R}[x_1, \ldots x_n, z_1, \ldots, z_m]$, and for every $r \in \mathbb{N}$, let $\varphi_r \in \mathbb{R}[z]$ be the polynomial

$$z \,\mapsto\, \varphi_r(z) \,=\, \sum_{k=0}^r \sum_{j=1}^m \frac{z_j^{2k}}{k!}. \tag{4.4}$$

We then get the following corollary.

COROLLARY 4.3. *Let $\mathbb{K}$ be as in* (4.2) *and θ_r, φ_r be as in* (3.2) *and* (4.4)*. Let $f \in \mathbb{R}[x]$ be nonnegative on $\mathbb{K}$. Then, for every $\epsilon > 0$, there exist nonnegative scalars $\{\lambda_j\}_{j=1}^m$ such that for all r sufficiently large,*

$$f + \epsilon\theta_r + \epsilon\varphi_r \,=\, q_\epsilon - \sum_{j=1}^m \lambda_j (g_j - z_j^2)^2 \tag{4.5}$$

for some s.o.s. polynomial $q_\epsilon \in \mathbb{R}[x, z]$.

Equivalently, everywhere on $\mathbb{K}$, the polynomial

$$x \mapsto f(x) + \epsilon \sum_{k=0}^{r} \sum_{i=1}^{n} \frac{x_i^{2k}}{k!} + \epsilon \sum_{k=0}^{r} \sum_{j=1}^{m} \frac{g_j(x)^k}{k!} \tag{4.6}$$

coincides with the nonnegative polynomial $x \mapsto q_\epsilon(x, \sqrt{g_1(x)}, \ldots, \sqrt{g_m(x)})$.

If q_ϵ is an s.o.s. of polynomials in $\mathbb{R}[x, z]$, from (4.5) q_ϵ is also a nonnegative polynomial in $\mathbb{R}[x, z^2]$, because the variables z_j only appear through even powers. Therefore, on $\mathbb{K}$, $x \mapsto q_\epsilon(x, \sqrt{g_1(x)}, \ldots, \sqrt{g_m(x)}) \in \mathbb{R}[x]$ as $z_j^2 = g_j(x)$ for all $j = 1, \ldots, m$.

So, as for the case of an algebraic set $V \subset \mathbb{R}^n$, (4.5) is an obvious certificate of positivity on the semialgebraic set $\mathbb{K}$ for the polynomial $f_{\epsilon r} \in \mathbb{R}[x, z]$

$$f_{\epsilon r} := f + \epsilon\theta_r + \epsilon\varphi_r,$$

and in addition, viewing f as an element of $\mathbb{R}[x, z]$, one has $\|f - f_{\epsilon r}\|_1 \to 0$ as $\epsilon \downarrow 0$. Notice that no assumption on $\mathbb{K}$ or on the g_j that define $\mathbb{K}$ is needed.

Now, assume that $\mathbb{K}$ is compact and the g_j that define $\mathbb{K}$ satisfy Putinar's condition, i.e., (i) there exits some $u \in \mathbb{R}[x]$ such that u can be written $u_0 + \sum_j u_j g_j$ for some s.o.s. polynomials $\{u_j\}_{j=0}^m$, and (ii) the level set $\{x | u(x) \geq 0\}$ is compact.

If f is nonnegative on $\mathbb{K}$, then $f + \epsilon\theta_r$ is strictly positive on $\mathbb{K}$ and therefore by Putinar's theorem [16]

$$f + \epsilon\theta_r = q_0 + \sum_{j=1}^{m} q_j g_j \tag{4.7}$$

for some s.o.s. family $\{q_j\}_{j=0}^m$. One may thus have either Putinar's representation (4.7) in $\mathbb{R}^n$ or (4.5) via a lifting in $\mathbb{R}^{n+m}$.

One may relate (4.5) and (4.7) by

$$q_\epsilon(x, z) = q_\epsilon^1(x) + q_\epsilon^2(x, z^2)$$

with

$$x \mapsto q_\epsilon^1(x) := q_0(x) + \sum_{j=1}^{m} \left(q_j(x) g_j(x) + \lambda_j g_j(x)^2\right)$$

and

$$(x, z) \mapsto q_\epsilon^2(x, z^2) := \epsilon\varphi_r(z) + \sum_{j=1}^{m} \lambda_j z_j^4 - 2 g_j(x) z_j^2.$$

4.2. Computational implications. The results of the previous section can be applied to compute (or at least approximate) the global minimum of f on V. Indeed, with $\epsilon > 0$ fixed, and $2r \geq \max[\deg f, \deg g_j^2]$, consider the convex optimization problem

$$\mathbb{Q}_{\epsilon r} \left\{ \begin{array}{lll} & \min\limits_{\mathbf{y}} L_{\mathbf{y}}(f_{\epsilon r}), & \\ \text{s.t.} & M_r(\mathbf{y}) & \succeq 0, \\ & L_{\mathbf{y}}(g_j^2) & \leq 0, \quad j = 1, \ldots, m, \\ & y_0 & = 1, \end{array} \right. \tag{4.8}$$

where θ_r is as in (3.2) and $L_{\mathbf{y}}$ and $M_r(\mathbf{y})$ are the linear functional and the moment matrix associated with a sequence $\mathbf{y}$ indexed in the basis (2.1); see (2.2) and (2.3).

$\mathbb{Q}_{\epsilon r}$ is called an SDP problem, and its associated dual SDP problem reads

$$(4.9)\qquad \mathbb{Q}^*_{\epsilon r}\left\{\begin{array}{lll} & \max\limits_{\lambda,\gamma,q} & \gamma \\ \text{s.t.} & f_{\epsilon r}-\gamma & = q - \sum_{j=1}^m \lambda_j g_j^2, \\ & \lambda\in\mathbb{R}^m, & \lambda\geq 0, \\ & q\in\mathbb{R}[x], & q \text{ s.o.s. of degree } \leq 2r. \end{array}\right.$$

Their optimal values are denoted $\inf\mathbb{Q}_{\epsilon r}$ and $\sup\mathbb{Q}^*_{\epsilon r}$, respectively (or $\min\mathbb{Q}_{\epsilon r}$ and $\max\mathbb{Q}^*_{\epsilon r}$ if the optimum is attained, in which case the problems are said to be solvable). Both problems $\mathbb{Q}_{\epsilon r}$ and its dual $\mathbb{Q}^*_{\epsilon r}$ are nice convex optimization problems that, in principle, can be solved efficiently by standard software packages. For more details on SDP theory, see [22].

That weak duality holds between $\mathbb{Q}_{\epsilon r}$ and $\mathbb{Q}^*_{\epsilon r}$ is straightforward. Let $\mathbf{y}=\{y_\alpha\}$ and $(\lambda,\gamma,q)\in\mathbb{R}^m_+\times\mathbb{R}\times\mathbb{R}[x]$ be feasible solutions of $\mathbb{Q}_{\epsilon r}$ and $\mathbb{Q}^*_{\epsilon r}$, respectively. Then, by linearity of $L_{\mathbf{y}}$,

$$\begin{aligned} L_{\mathbf{y}}(f_{\epsilon r})-\gamma &= L_{\mathbf{y}}(f_{\epsilon r}-\gamma) \\ &= L_{\mathbf{y}}\left(q-\sum_{j=1}^m \lambda_j g_j^2\right) \;=\; L_{\mathbf{y}}(q)-\sum_{j=1}^m \lambda_j L_{\mathbf{y}}(g_j^2) \\ &\geq L_{\mathbf{y}}(q) \quad [\text{because } L_{\mathbf{y}}(g_j^2)\leq 0\,\forall\, j=1,\dots,m] \\ &\geq 0 \quad [\text{because } q \text{ is s.o.s. and } M_r(\mathbf{y})\succeq 0\,;\text{ see (2.6)}]. \end{aligned}$$

Therefore, $L_{\mathbf{y}}(f_{\epsilon r})\geq\gamma$, the desired conclusion. Moreover, $\mathbb{Q}_{\epsilon r}$ is an obvious relaxation of the perturbed problem

$$\mathbb{P}_{\epsilon r}: \quad f^*_{\epsilon r} := \min_x\,\{f_{\epsilon r}\mid x\in V\}.$$

Indeed, let $x\in V$ and let $\mathbf{y}:=v_{2r}(x)$ (see (2.1)), i.e., $\mathbf{y}$ is the vector of moments (up to order $2r$) of the Dirac measure at $x\in V$. Then, $\mathbf{y}$ is feasible for $\mathbb{Q}_{\epsilon r}$ because $y_0=1$, $M_r(\mathbf{y})\succeq 0$, and $L_{\mathbf{y}}(g_j^2)=g_j(x)^2=0$, for all $j=1,\dots,m$. Similarly, $L_{\mathbf{y}}(f_{\epsilon r})=f_{\epsilon r}(x)$. Therefore, $\inf\mathbb{Q}_{\epsilon r}\leq f^*_{\epsilon r}$.

THEOREM 4.4. *Let $V\subset\mathbb{R}^n$ be as in* (1.1) *and θ_r as in* (3.2). *Assume that f has a global minimizer $x^*\in V$ with $f(x^*)=f^*$. Let $\epsilon>0$ be fixed. Then*

$$(4.10)\qquad f^* \;\leq\; \sup\mathbb{Q}^*_{\epsilon r} \;\leq\; \inf\mathbb{Q}_{\epsilon r} \;\leq\; f^*+\epsilon\theta_r(x^*) \;\leq\; f^*+\epsilon\sum_{i=1}^n \mathrm{e}^{(x_i^*)^2},$$

provided that r is sufficiently large.

Proof. Observe that the polynomial $f-f^*$ is nonnegative on V. Therefore, by Theorem 4.1, for every ϵ there exists $r(\epsilon)\in\mathbb{N}$ and $\lambda(\epsilon)\in\mathbb{R}^m_+$ such that

$$f-f^*+\epsilon\theta_r+\sum_{j=1}^m \lambda_j(\epsilon)g_j^2 \;=\; q_\epsilon$$

for some s.o.s. polynomial $q_\epsilon\in\mathbb{R}[x]$. But this shows that $(\lambda(\epsilon),f^*,q_\epsilon)\in\mathbb{R}^m_+\times\mathbb{R}\times\mathbb{R}[x]$ is a feasible solution of $\mathbb{Q}^*_{\epsilon r}$ as soon as $r\geq r(\epsilon)$, in which case $\sup\mathbb{Q}^*_{\epsilon r}\geq f^*$. Moreover, we have seen that $\inf \mathbb{Q}_{\epsilon r} \leq f_{\epsilon r}(x)$ for any feasible solution $x \in V$. In particular, $\inf \mathbb{Q}_{\epsilon r} \leq f^* + \epsilon\theta_r(x^*)$, from which (4.10) follows. $\square$

Theorem 4.4 has a nice feature. Suppose that one knows some bound ρ on the norm $\|x^*\|_\infty$ of a global minimizer of f on V. Then, one may fix a priori the error bound η on $|\inf \mathbb{Q}_{\epsilon r} - f^*|$. Indeed, let η be fixed, and fix $\epsilon > 0$ such that $\epsilon \leq \eta(ne^{\rho^2})^{-1}$. By Theorem 4.4, one has $f^* \leq \inf \mathbb{Q}_{\epsilon r} \leq f^* + \eta$, provided that r is large enough.

The same approach works to approximate the global minimum of a polynomial f on a semialgebraic set $\mathbb{K}$, as defined in (4.2). In view of Corollary 4.3, and via a lifting in $\mathbb{R}^{n+m}$, one is reduced to the case of a real algebraic set $V \subset \mathbb{R}^{n+m}$, so that Theorem 4.4 still applies. It is important to emphasize that one requires no assumption on $\mathbb{K}$ or on the g_j that define $\mathbb{K}$. This is to be compared with previous SDP relaxation techniques developed in, e.g., [8, 9, 10, 14, 21], where the set $\mathbb{K}$ is supposed to be compact, and with an additional assumption on the g_j to ensure that Putinar's representation [16] holds.

5. Proofs.

5.1. Proof of Theorem 3.3.

To prove the absence of a duality gap, we first rewrite $\mathcal{P}_M^r$ (resp., $\mathcal{Q}_M^r$) as a linear program in (standard) form

$$\min_x\{\langle x, c\rangle \mid \quad Gx = b, x \in C\}, \quad (\text{resp., } \max_w\{\langle w, b\rangle \mid \quad c - G^*w \in C^*\})$$

on appropriate dual pairs of vector spaces, with associated convex cone C (and its dual C^*), and associated linear map G (and its adjoint G^*). Then, we will prove that G is continuous, and the set $D := \{(Gx, \langle x, c\rangle) \mid x \in C\}$ is closed, in some appropriate weak topology. This permits us to conclude by invoking standard results in infinite dimensional linear programming, that one may find in, e.g., Anderson and Nash [1]. For a brief account see section 6.1, and for more details, see, e.g., Robertson and Robertson [17] and Anderson and Nash [1].

Let θ_r be as in (3.2), and let $M(\mathbb{R}^n)$ be the $\mathbb{R}$-vector space of finite signed Borel measures μ on $\mathbb{R}^n$, such that $\int \theta_r \, d|\mu| < \infty$ (where $|\mu|$ denotes the total variation of μ). Similarly, let H^r be the $\mathbb{R}$-vector space of continuous functions $h : \mathbb{R}^n \to \mathbb{R}$, such that $\sup_{x\in\mathbb{R}^n} |h(x)|/\theta_r(x) < \infty$. With the bilinear form $\langle .,. \rangle : M(\mathbb{R}^n) \times H^r$, defined as

$$(\mu, h) \mapsto \langle \mu, h\rangle = \int h \, d\mu, \qquad (\mu, h) \in M(\mathbb{R}^n) \times H^r,$$

$(M(\mathbb{R}^n), H^r)$ forms a dual pair of vector spaces (see section 6.1). Introduce the dual pair of vector spaces $(\mathcal{X}, \mathcal{Y})$,

$$\mathcal{X} := M(\mathbb{R}^n) \times \mathbb{R}^m \times \mathbb{R}, \quad \mathcal{Y} := H^r \times \mathbb{R}^m \times \mathbb{R},$$

and $(\mathcal{Z}, \mathcal{W})$

$$\mathcal{Z} := \mathbb{R}^m \times \mathbb{R} \times \mathbb{R}, \quad \mathcal{W} := \mathbb{R}^m \times \mathbb{R} \times \mathbb{R}.$$

Recall that $2r > \deg g_j^2$ for all $j = 1, \ldots, m$, and let $G : \mathcal{X} \to \mathcal{Z}$ be the linear map

$$(\mu, u, v) \mapsto G(\mu, u, v) := \begin{bmatrix} \langle \mu, g_1^2\rangle + u_1 \\ \cdots \\ \langle \mu, g_m^2\rangle + u_m \\ \langle \mu, \theta_r\rangle + v \\ \langle \mu, 1\rangle \end{bmatrix}$$

with associated adjoint linear map $G^*:\mathcal{W}\to\mathcal{X}^*$

$$(\lambda,\delta,\gamma)\,\mapsto\,G^*(\lambda,\delta,\gamma)\,:=\,\left[\begin{array}{l}\sum_{j=1}^m\lambda_j g_j^2+\delta\theta_r+\gamma\\ \lambda\\ \delta\end{array}\right].$$

Notice that $G^*(\mathcal{W})\subseteq\mathcal{Y}$ because $2r>\deg g_j^2$ for all $j=1,\dots,m$, and so $g_j^2\in H^r$ for all $j=1,\dots,m$.

Next, let $M(\mathbb{R}^n)_+\subset M(\mathbb{R}^n)$ be the convex cone of nonnegative finite Borel measures of $M(\mathbb{R}^n)$, so that the set $C:=M(\mathbb{R}^n)_+\times\mathbb{R}^m_+\times\mathbb{R}_+\subset\mathcal{X}$ is a convex cone in $\mathcal{X}$. If H^r_+ denotes the cone of nonnegative functions of H^r, then

$$C^*\,=\,H^r_+\times\mathbb{R}^m_+\times\mathbb{R}_+\subset\mathcal{Y}$$

is the dual cone of C in $\mathcal{Y}$.

As $2r>\max[2\mathrm{deg}f,\deg g_j^2]$ it follows that $f\in H^r$ and $g_j^2\in H^r$ for all $j=1,\dots,m$. Then, by introducing slack variables $u\in\mathbb{R}^m_+,v\in\mathbb{R}_+$, rewrite the infinite dimensional linear program $\mathcal{P}^r_M$ defined in (3.5), in equality form, that is,

$$\text{(5.1)}\qquad \mathcal{P}^r_M:\left\{\begin{array}{rl} & \inf\limits_{\mu,u,v}\,\langle(\mu,u,v),(f,0,0)\rangle\\ \text{s.t.} & G(\mu,u,v)\,=\,\left[\begin{array}{l}0\\ n\mathrm{e}^{M^2}\\ 1\end{array}\right],\\ & (\mu,u,v)\in C.\end{array}\right.$$

The LP dual $(\mathcal{P}^r_M)^*$ of $\mathcal{P}^r_M$ now reads

$$\text{(5.2)}\qquad (\mathcal{P}^r_M)^*:\left\{\begin{array}{rl} & \sup\limits_{\lambda,\delta,\gamma}\,\langle(\lambda,\delta,\gamma),(0,n\mathrm{e}^{M^2},1)\rangle\\ \text{s.t.} & (f,0,0)-G^*(\lambda,\delta,\gamma)\in C^*.\end{array}\right.$$

Hence, every feasible solution (λ,δ,γ) of $(\mathcal{P}^r_M)^*$ satisfies

$$\text{(5.3)}\qquad f-\sum_{j=1}^m\lambda_j\,g_j^2-\delta\,\theta_r-\gamma\geq 0,\quad \lambda,\delta\leq 0.$$

As $\lambda,\delta\leq 0$ in (5.2), one may see that the two formulations (5.2) and (3.6) are identical, i.e., $\mathcal{Q}^r_M=(\mathcal{P}^r_M)^*$.

As $2r>\max[2\mathrm{deg}f,\deg g_j^2]$, it follows that $f-\sum_{j=1}^m\lambda_j\,g_j^2-\delta\,\theta_r-\gamma\in H^r$, for all $(\lambda,\delta,\gamma)\in\mathcal{W}$. As $G^*(\mathcal{W})\subset\mathcal{Y}$, by Proposition 6.2, the linear map G is weakly continuous (i.e., is continuous with respect to the weak topologies $\sigma(\mathcal{X},\mathcal{Y})$ and $\sigma(\mathcal{Z},\mathcal{W})$). We next prove that the set $D\subset\mathcal{Z}\times\mathbb{R}$, defined as

$$\text{(5.4)}\qquad D\,:=\,\{(G(\mu,u,v),\langle(\mu,u,v),(f,0,0)\rangle)\mid\quad(\mu,u,v)\in C\},$$

is weakly closed.

For some directed set $(A,\geq)$, let $\{(\mu_\beta,u_\beta,v_\beta)\}_{\beta\in A}$ be a net in C, such that

$$(G(\mu_\beta,u_\beta,v_\beta),\langle(\mu_\beta,u_\beta,v_\beta),(f,0,0)\rangle)\,\to\,((a,b,c),d),$$

weakly, for some element $((a,b,c),d)\in\mathcal{Z}\times\mathbb{R}$. (In fact, as $\mathcal{Z}\times\mathbb{R}\equiv\mathbb{R}^{m+3}$ it suffices to consider sequences instead of nets.) In particular,

$$\mu_\beta(\mathbb{R}^n)\,\to\,c;\quad\langle\mu_\beta,\theta_r\rangle+v_\beta\,\to\,b;\quad\langle\mu_\beta,g_j^2\rangle+(u_\beta)_j\,\to\,a_j,\;j=1,\dots,m,$$

and $\langle \mu_\beta, f\rangle \to d$. As $(\mu_\beta, u_\beta, v_\beta) \in C$, and $\theta_r, g_j^2 \geq 0$, it follows immediately that $a, b, c \geq 0$. We need to consider the two cases $c = 0$ and $c > 0$.

Case $c = 0$. From $\mu_\beta(\mathbb{R}^n) \to c$, it follows that $\mu_\beta \to \mu := 0$ in the total variation norm. But in this case, observe that $G(\mu, a, b) = (a, b, c)$. It remains to prove that we also have $\langle \mu_\beta, f\rangle \to d = 0$, in which case, $(G(\mu, a, b), \langle \mu, f\rangle) = ((a, b, c), d)$, as desired.

Recall that $r \geq \mathrm{deg} f$. Denote by $\{y_\alpha(\beta)\}_{|\alpha|\leq 2r}$ the sequence of moments of the measure μ_β, i.e.,

$$y_\alpha(\beta) \;=\; \int x^\alpha \, d\mu_\beta, \quad \alpha \in \mathbb{N}^n, \quad |\alpha| \leq 2r.$$

In particular, $y_0(\beta) = \mu_\beta(\mathbb{R}^n)$. From $\langle \mu_\beta, \theta_r\rangle + v_\beta \to b$, there is some $\beta_0 \in A$, such that $\langle \mu_\beta, \theta_r\rangle \leq 2b$ for all $\beta \geq \beta_0$. But this implies that

$$y_{2k}(i, \beta) \;:=\; \int x_i^{2k} \, d\mu_\beta \leq 2r!b, \quad k \leq r, \quad i = 1, \ldots, n.$$

By Lemma 6.6, it follows that $y_{2\alpha}(\beta) \leq 2br!$ for all $\alpha \in \mathbb{N}^n$ with $|\alpha| \leq r$ and $|y_\alpha(\beta)| \leq \sqrt{2y_0(\beta)\, br!}$ for all $|\alpha| \leq r$. But then, as $y_0(\beta) = \mu_\beta(\mathbb{R}^n) \to c = 0$, we thus obtain $y_\alpha(\beta) \to 0$ for all $|\alpha| \leq r$. Therefore,

$$\langle \mu_\beta, f\rangle \;=\; \int f \, d\mu_\beta \;=\; \sum_{|\alpha|\leq r} f_\alpha \int x^\alpha \, d\mu_\beta \;=\; \sum_{|\alpha|\leq r} f_\alpha y_\alpha(\beta) \to 0,$$

the desired result.

Case $c > 0$. From $\mu_\beta(\mathbb{R}^n) \to c$ and $\langle \mu_\beta, \theta_r\rangle + v_\beta \to b$, there is some $\beta_0 \in A$, such that $\mu_\beta(\mathbb{R}^n) \leq 2c$ and $\langle \mu_\beta, \theta_r\rangle \leq 2b$ for all $\beta \geq \beta_0$. But, as θ_r is a moment function, this implies that the family $\Delta := \{\nu_\beta := \mu_\beta/\mu_\beta(\mathbb{R}^n)\}_{\beta\geq\alpha_0}$ is a tight family of probability measures, and as Δ is a set of probability measures on a metric space, by Prohorov's theorem, Δ is relatively compact (see [7, Chap. 1] and section 6.2). Therefore, there is some probability measure $\nu^* \in M(\mathbb{R}^n)$, and a sequence $\{n_k\} \subset \Delta$, such that ν_{n_k} converges to ν^* for the weak convergence of probability measures, i.e.,

$$\langle \nu_{n_k}, h\rangle \;\to\; \langle \nu^*, h\rangle \qquad \forall h \in C_b(\mathbb{R}^n)$$

(where $C_b(\mathbb{R}^n)$ denotes the space of bounded continuous functions on $\mathbb{R}^n$); see, e.g., Billingsley [5]. Hence, with $\mu^* := c\,\nu^*$, we also conclude

$$\langle \mu_{n_k}, h\rangle \;\to\; \langle \mu^*, h\rangle \qquad \forall h \in C_b(\mathbb{R}^n). \tag{5.5}$$

Next, as $2r > \max[2\mathrm{deg}\, f, \deg g_j^2]$, the functions f/θ_{r-1} and g_j^2/θ_{r-1}, $j = 1, \ldots, m$, are all in $C_b(\mathbb{R}^n)$. Therefore, using Lemma 6.5, we obtain

$$\langle \nu_{n_k}, f\rangle \;\to\; \langle \nu^*, f\rangle, \;\text{ and }\; \langle \nu_{n_k}, g_j^2\rangle \;\to\; \langle \nu^*, g_j^2\rangle,\; j = 1, \ldots, m.$$

And, therefore,

$$\langle \mu_{n_k}, f\rangle \;\to\; \langle \mu^*, f\rangle \;=\; d, \;\text{ and }\; \langle \mu_{n_k}, g_j^2\rangle \;\to\; \langle \mu^*, g_j^2\rangle,\; j = 1, \ldots, m.$$

Finally, from the weak convergence (5.5), and as θ_r is continuous and nonnegative,

$$\langle \mu^*, \theta_r\rangle \;\leq\; \liminf_{k\to\infty} \langle \mu_{n_k}, \theta_r\rangle \;\leq\; b;$$

see, e.g., [7, Prop. 1.4.18].

So, let $v := b - \langle \mu^*, \theta_r \rangle \geq 0$ and $u_j := a_j - \langle \mu^*, g_j^2 \rangle \geq 0$, $j = 1, \ldots, m$, and recalling that $c = \mu^*(\mathbb{R}^n)$, we conclude that $G(\mu^*, u, v) = (a, b, c)$ and $\langle (\mu^*, u, v), (f, 0, 0) \rangle = d$, which proves that the set D in (5.4) is weakly closed.

Finally, by Proposition 3.2, $\mathcal{P}_M^r$ is consistent with finite value as soon as M is large enough to ensure that $B_M \cap V \neq \emptyset$. Therefore, one may invoke Theorem 6.4 and conclude that there is no duality gap between $\mathcal{P}_M^r$ and its dual $\mathcal{Q}_M^r$, the desired result.

5.2. Proof of Theorem 4.1. It suffices to prove the result for the case where $\inf_{x \in V} f(x) = f^* > 0$. Indeed, suppose that $f^* = 0$. Then with $\epsilon > 0$ fixed, arbitrary, $f^* + n\epsilon > 0$ and so suppose that (4.1) holds for $\hat{f} := f + n\epsilon$. There is some $r(\epsilon) \in \mathbb{N}$ such that for all $r \geq r(\epsilon)$,

$$\hat{f} \,=\, f + n\epsilon + \epsilon\,\theta_r \,=\, q_{\epsilon r} - \sum_{j=1}^{m} \lambda_j g_j^2$$

for some s.o.s. polynomial $q_{\epsilon r}$ and some nonnegative scalars $\{\lambda_j\}$. Equivalently,

$$f + 2\epsilon\,\theta_r \,=\, q_{\epsilon r} + \epsilon \sum_{k=1}^{r} \sum_{j=1}^{n} \frac{x_j^{2k}}{k!} - \sum_{j=1}^{m} \lambda_j g_j^2 \,=\, \hat{q}_{\epsilon r} - \sum_{j=1}^{m} \lambda_j g_j^2,$$

where $\hat{q}_{\epsilon r}$ is an s.o.s. polynomial. Equivalently, $f_{2\epsilon r} = \hat{q}_{\epsilon r} - \sum_{j=1}^{m} \lambda_j g_j^2$, so that (4.1) also holds for f. Therefore, from now on, we will assume that $f^* > 0$.

So let $\epsilon > 0$ (fixed) be such that $f^* - \epsilon > 0$, and let $r \geq r_0$ with r_0 as in Theorem 3.3. Next, by Proposition 3.2, let M be such that $f^* \leq \inf \mathcal{P}_M^r \leq f^* + \epsilon$. By Theorem 3.3, we then have $\sup \mathcal{Q}_M^r \geq f^*$. So, by considering a maximizing sequence of $\mathcal{Q}_M^r$, there is some $(\lambda, \delta, \gamma) \in \mathbb{R}_+^m \times \mathbb{R}_+ \times \mathbb{R}$, such that

$$0 < f^* - \epsilon < \gamma - n\delta \mathrm{e}^{M^2} \leq f^* + \epsilon; \quad f + \delta\theta_r + \sum_{j=1}^{m} \lambda_j\, g_j^2 \geq \gamma, \tag{5.6}$$

and so,

$$f - (\gamma - n\delta \mathrm{e}^{M^2}) + \sum_{j=1}^{m} \lambda_j\, g_j^2 \geq \delta(n\mathrm{e}^{M^2} - \theta_r). \tag{5.7}$$

By Proposition 3.1, we may choose M such that there is some $x_M \in B_{M/2} \cap V$ such that $f(x_M) \leq f^* + \epsilon$. Evaluating (5.7) at $x = x_M$ yields

$$2\epsilon \,\geq\, f(x_M) - (\gamma - n\delta \mathrm{e}^{M^2}) \geq \delta(n\mathrm{e}^{M^2} - \theta_r(x_M)), \tag{5.8}$$

and so, using $\|x_M\|_\infty \leq M/2$,

$$2\epsilon \geq \delta n(\mathrm{e}^{M^2} - \mathrm{e}^{M^2/4}), \tag{5.9}$$

which yields $\delta \leq 2\epsilon/n(\mathrm{e}^{M^2} - \mathrm{e}^{M^2/4})$. Therefore, given $\epsilon > 0$, one may pick $(\lambda, \delta, \gamma)$ in a maximizing sequence of $\mathcal{Q}_M^r$, in such a way that $\delta \leq \epsilon$.

For such a choice of $(\lambda, \delta, \gamma)$, and in view of (5.6), we have

$$f + \delta\theta_r + \sum_{j=1}^{m} \lambda_j\, g_j^2 \geq (\gamma - n\delta \mathrm{e}^{M^2}) + n\delta \mathrm{e}^{M^2} \,\geq\, f^* - \epsilon + n\delta \mathrm{e}^{M^2} \geq 0,$$

so that the polynomial $h := f + \delta\theta_r + \sum_{j=1}^{m} \lambda_j\, g_j^2$ is nonnegative.

Therefore, invoking Theorem 3.4 proved in Lasserre [11], there is some $r(\epsilon) \in \mathbb{N}$ such that for all $s \geq r(\epsilon)$, the polynomial $q_\epsilon := h + \epsilon\theta_s$ is an s.o.s. But then, take $s > \max[r, r(\epsilon)]$ and observe that

$$\delta\theta_r + \epsilon\theta_s \;=\; (\delta+\epsilon)\theta_s - \delta \sum_{k=r+1}^{s} \sum_{j=1}^{n} \frac{x_i^2}{k!},$$

and so

$$q_\epsilon = h + \epsilon\theta_s = f + \sum_{j=1}^{m} \lambda_j g_j^2 + (\delta+\epsilon)\theta_s - \delta \sum_{k=r+1}^{s} \sum_{i=1}^{n} \frac{x_i^2}{k!},$$

or, equivalently,

$$f + \sum_{j=1}^{m} \lambda_j\, g_j^2 + (\delta+\epsilon)\theta_s \;=\; q_\epsilon + \delta \sum_{k=r+1}^{s} \sum_{j=1}^{n} \frac{x_i^2}{k!} \;=\; \hat{q}_\epsilon,$$

where $\hat{q}_\epsilon$ is an s.o.s. polynomial.

As δ was chosen to satisfy $\delta \leq \epsilon$, we obtain

$$f + \sum_{j=1}^{m} \lambda_j\, g_j^2 + 2\epsilon\theta_s \;=\; \hat{q}_\epsilon + (\epsilon-\delta)\theta_s \;=\; \hat{\hat{q}}_\epsilon,$$

where again, $\hat{\hat{q}}_\epsilon$ is an s.o.s. polynomial.

6. Appendix.

In this section, we first briefly recall some basic results of linear programming in infinite dimensional spaces and then present auxiliary results that are used in some of the proofs in section 5.

6.1. Linear programming in infinite dimensional spaces.

6.1.1. Dual pairs. Let $\mathcal{X}, \mathcal{Y}$ be two arbitrary (real) vector spaces, and let $\langle ., . \rangle$ be a bilinear form on $\mathcal{X} \times \mathcal{Y}$, that is, a real-valued function on $\mathcal{X} \times \mathcal{Y}$ such that

- the map $x \mapsto \langle x, y\rangle$ is linear on $\mathcal{X}$ for every $y \in \mathcal{Y}$,
- the map $y \mapsto \langle x, y\rangle$ is linear on $\mathcal{Y}$ for every $x \in \mathcal{X}$.

Then the pair $(\mathcal{X}, \mathcal{Y})$ is called a dual pair if the bilinear form separates points in $\mathcal{X}$ and $\mathcal{Y}$, that is,

- for each $0 \neq x \in \mathcal{X}$, there is some $y \in \mathcal{Y}$ such that $\langle x, y\rangle \neq 0$, and
- for each $0 \neq y \in \mathcal{Y}$, there is some $x \in \mathcal{X}$ such that $\langle x, y\rangle \neq 0$.

Given a dual pair $(\mathcal{X}, \mathcal{Y})$, we denote by $\sigma(\mathcal{X}, \mathcal{Y})$ the weak topology on $\mathcal{X}$ (also referred to as the σ-topology on $\mathcal{X}$), namely, the coarsest—or weakest—topology on $\mathcal{X}$, under which all the elements of $\mathcal{Y}$ are continuous when regarded as linear forms $\langle ., y\rangle$ on $\mathcal{X}$. In addition, let $\mathcal{X}^*$ be the algebraic dual of $\mathcal{X}$.

Equivalently, the base of neighborhoods of the origin of the σ-topology is the family of all sets of the form

$$N(I, \epsilon) \;:=\; \{x \in \mathcal{X} \mid \quad |\langle x, y\rangle \leq \epsilon \quad \forall y \in I\},$$

where $\epsilon > 0$ and I is a finite subset of $\mathcal{Y}$. (See, for instance, Robertson and Robertson [17, p. 32].) In this case, if $\{x_n\}$ is a net or a sequence in $\mathcal{X}$, then x_n converges to x in the weak topology $\sigma(\mathcal{X}, \mathcal{Y})$ if

$$\langle x_n, y\rangle \;\to\; \langle x, y\rangle \qquad \forall y \in \mathcal{Y}.$$

Definition 6.1. *Let $(\mathcal{X},\mathcal{Y})$ and $(\mathcal{Z},\mathcal{W})$ be two dual pairs of vector spaces, and let $G:\mathcal{X}\to\mathcal{Z}$ be a linear map.*

(a) *G is said to be weakly continuous if it is continuous with respect to the weak topologies $\sigma(\mathcal{X},\mathcal{Y})$ and $\sigma(\mathcal{Z},\mathcal{W})$; that is, if $\{x_n\}$ is a net in $\mathcal{X}$ such that $x_n\to x$ in the weak topology $\sigma(\mathcal{X},\mathcal{Y})$, then $Gx_n\to Gx$ in the weak topology $\sigma(\mathcal{X},\mathcal{Y})$, i.e.,*

$$\langle Gx_n,v\rangle\;\to\;\langle Gx,v\rangle\qquad\forall v\in\mathcal{W}.$$

(b) *The adjoint $G^*:\mathcal{W}\to\mathcal{X}^*$ of G is defined by the relation*

$$\langle Gx,v\rangle\;=\;\langle x,G^*v\rangle\qquad\forall x\in\mathcal{X},\,v\in\mathcal{W}.$$

The following proposition gives a well-known (easy to use) criterion for the map G in Definition 6.1, to be weakly continuous.

Proposition 6.2. *The linear map G is weakly continous if and only if its adjoint G^* maps $\mathcal{W}$ into $\mathcal{Y}$, that is, $G^*(\mathcal{W})\subseteq\mathcal{Y}$.*

6.1.2. Positive and dual cones. Let $(\mathcal{X},\mathcal{Y})$ be a dual pair of vector spaces, and C a convex cone in $\mathcal{X}$, that is, $x+x'$ and λx belong to C whenever x and x' are in C and $\lambda>0$. Unless explicitly stated otherwise, we shall assume that C is not the whole space, that is, $C\neq\mathcal{X}$, and that the origin (the zero vector in $\mathcal{X}$) is in C. In this case, C defines a partial order $\geq$ in $\mathcal{X}$, such that

$$x\;\geq x'\quad\Leftrightarrow\quad x-x'\;\in C,$$

and C is referred to as a positive cone in $\mathcal{X}$. The dual cone of C is the convex cone C^* in $\mathcal{Y}$ defined by

$$C^*\;:=\;\{y\in\mathcal{Y}\,|\quad\langle x,y\rangle\;\geq\;0\quad\forall x\in C\}.$$

6.1.3. Infinite linear programming. An infinite linear program requires the following components:

- two dual pairs of vector spaces $(\mathcal{X},\mathcal{Y})$;
- a weakly continuous linear map $G:\mathcal{X}\to\mathcal{Z}$, with adjoint $G^*:\mathcal{W}\to\mathcal{Y}$;
- a positive cone C in $\mathcal{X}$, with dual cone C^* in $\mathcal{Y}$; and
- vectors $b\in\mathcal{Z}$ and $c\in\mathcal{Y}$.

Then the primal linear program is

$$\text{(6.1)}\qquad\mathbb{P}:\quad\begin{array}{l}\text{minimize }\langle x,c\rangle\\ \text{subject to: }Gx\;=\;b,\quad x\in C.\end{array}$$

The corresponding dual linear program is

$$\text{(6.2)}\qquad\mathbb{P}^*:\quad\begin{array}{l}\text{maximize }\langle b,w\rangle\\ \text{subject to: }c-G^*w\;\in\;C^*,\quad w\in\mathcal{W}.\end{array}$$

An element of $x\in\mathcal{X}$ is called feasible for $\mathbb{P}$ if it satisfies (6.1), and $\mathbb{P}$ is said to be consistent, if it has a feasible solution. If $\mathbb{P}$ is consistent, then its value is defined as

$$\inf\mathbb{P}\;:=\;\inf\{\langle x,c\rangle\,|\quad x\text{ is feasible for }\mathbb{P}\};$$

otherwise, $\inf\mathbb{P}\;=\;+\infty$. The linear program $\mathbb{P}$ is solvable if there is some feasible solution $x^*\in\mathcal{X}$ that achieves the value $\inf\mathbb{P}$; then x^* is an optimal solution of $\mathbb{P}$, and one then writes $\inf\mathbb{P}=\min\mathbb{P}$. The same definitions apply for the dual linear program $\mathbb{P}^*$.

The next result can be proved as in elementary (finite-dimensional) linear programming.

PROPOSITION 6.3 (weak duality). *If $\mathbb{P}$ and $\mathbb{P}^*$ are both consistent, then their values are finite and satisfy* $\sup \mathbb{P}^* \leq \inf \mathbb{P}$.

There is no duality gap if $\sup \mathbb{P}^* = \inf \mathbb{P}$, and strong duality holds if $\max \mathbb{P}^* = \min \mathbb{P}$, i.e., if there is no duality gap, and both $\mathbb{P}^*$ and $\mathbb{P}$ are solvable.

THEOREM 6.4. *Let D be the set in $\mathcal{Z} \times \mathbb{R}$, defined as*

$$D := \{(Gx, \langle x, c\rangle) \mid \quad x \in C\}. \tag{6.3}$$

If $\mathbb{P}$ is consistent with finite value, and D is weakly closed (i.e., closed in the weak topology $\sigma(\mathcal{Z}\times\mathbb{R}, \mathcal{W}\times\mathbb{R})$), then $\mathbb{P}$ is solvable and there is no duality gap, i.e., $\sup \mathbb{P}^* = \min \mathbb{P}$. *(See Anderson and Nash* [1, Theorems 3.10 *and* 3.22]*.)*

6.2. Auxiliary results. Let X be a metric space. A nonnegative function $f : X \to \mathbb{R}_+$ is said to be a moment if there is a sequence of compact sets $\mathbb{K}_n \uparrow X$ as $n \to \infty$, such that

$$\lim_{n\to\infty} \inf_{x\in X\setminus\mathbb{K}_n} f(x) = +\infty.$$

For instance, with $X := \mathbb{R}^n$, the function $x \mapsto f(x) = x'Qx$ for some positive definite symmetric matrix $Q \in \mathbb{R}^{n\times n}$ is a moment function; take, for instance, $\mathbb{K}_m := \{x \in \mathbb{R}^n : x'Qx \leq m\}$ for all $m = 1, 2, \ldots$.

Moment functions are very useful because they provide a sufficient criterion for tightness of a set of probability measures. Indeed, let $\mathcal{P}(X)$ be the space of probability measures on X, and let $\Pi \subset \mathcal{P}(X)$ be a given set of probability measures. If

$$\sup_{\mu\in\Pi} \int f\, d\mu < \infty$$

for some moment function f, then Π is tight, and therefore, by Prohorov's theorem, Π is relatively compact for the topology of weak convergence of probability measures. That is, for any sequence $\{\mu_n\} \subset \Pi$, there exists a probability measure $\mu \in \mathcal{P}(X)$, and a subsequence $\{n_k\}$ such that

$$\lim_{k\to\infty} \int h\, d\mu_{n_k} = \int h\, d\mu$$

for all bounded continuous functions $h : X \to \mathbb{R}$; see, e.g., Hernández-Lerma and Lasserre [7, pp. 10–11].

Let $\mathcal{B}$ be the Borel sigma-algebra of $\mathbb{R}^n$, $C_b(\mathbb{R}^n)$ be the space of bounded continuous functions on $\mathbb{R}^n$, and let θ_r be as in (3.2). Let $M(\mathbb{R}^n)$ be the space of finite signed Borel measures on $\mathbb{R}^n$.

LEMMA 6.5. *Let $r \geq 1$, and let $\{\mu_j\}_{j\in J} \subset M(\mathbb{R}^n)$ be a sequence of probability measures such that*

$$\sup_{j\in J} \int \theta_r\, d\mu_j < \infty. \tag{6.4}$$

Then there is a subsequence $\{j_k\} \subset J$ and a probability measure μ on $\mathbb{R}^n$ (not necessarily in $\mathcal{M}$) such that

$$\lim_{k\to\infty} \int f\, d\mu_{j_k} = \int f\, d\mu$$

for all continuous functions $f : \mathbb{R}^n \to \mathbb{R}$, such that $f/\theta_{r-1} \in C_b(\mathbb{R}^n)$.

Proof. θ_r is a moment function (see (3.3)), and so, (6.4) implies that the sequence $\{\mu_j\}$ is tight. Hence, as $\mathbb{R}^n$ is a metric space, by Prohorov's theorem [7, Theorem 1.4.12], there is a subsequence $\{j_k\} \subset J$ and a measure $\mu \in M(\mathbb{R}^n)$ such that $\mu_{j_k} \Rightarrow \mu$, i.e.,

$$\int h\, \mu_{j_k} \,\to\, \int h\, d\mu, \tag{6.5}$$

for all $h \in C_b(\mathbb{R}^n)$. Next, let ν_{j_k} be the measure obtained from μ_{j_k} by

$$\nu_{j_k}(B) \,:=\, \int_B \theta_{r-1}\, d\mu_{j_k}, \qquad B \in \mathcal{B}.$$

Observe that from the definition of θ_r, the function θ_r/θ_{r-1} is a moment function, for every $r \geq 1$. And one has

$$\sup_k \int \theta_r/\theta_{r-1}\, d\nu_{j_k} \,=\, \sup_k \int \theta_r\, d\mu_{j_k} \,<\, \infty$$

because of (6.4). Observe that $\nu_{j_k}(\mathbb{R}^n) \leq \rho_0$ for some ρ_0 and all k, and so we may consider a subsequence of $\{j_k\}$ (still denoted $\{j_k\}$ for simplicy of notation) such that $\nu_{j_k}(\mathbb{R}^n) \to \rho\,(> 0)$ as $k \to \infty$. With $\hat{\nu}_{j_k} := \nu_{j_k}/\nu_{j_k}(\mathbb{R}^n)$ for all k, it follows that the sequence of probability measures $\{\hat{\nu}_{j_k}\}_k$ is tight, which implies that there is a subsequence $\{j_n\}$ of $\{j_k\}$ and a measure $\hat{\nu} \in M(\mathbb{R}^n)$ such that

$$\text{as } n \to \infty, \quad \int h\, d\hat{\nu}_{j_n} \,\to\, \int h\, d\hat{\nu} \quad \forall h \in C_b(\mathbb{R}^n).$$

Since $\nu_{j_k}(\mathbb{R}^n) \to \rho$ as $k \to \infty$, we immediately get

$$\int h\, d\nu_{j_n} \,=\, \int h\,(\rho + \nu_{j_n}(\mathbb{R}^n) - \rho)\, d\hat{\nu}_{j_n} \,\to\, \int h\rho\, d\hat{\nu} \qquad \text{as } n \to \infty$$

for all $h \in C_b(\mathbb{R}^n)$. Equivalently, with $\nu := \rho\hat{\nu}$,

$$\text{as } n \to \infty, \quad \int h\, d\nu_{j_n} \,\to\, \int h\, d\nu \quad \forall h \in C_b(\mathbb{R}^n). \tag{6.6}$$

But as $h/\theta_{r-1} \in C_b(\mathbb{R}^n)$ whenever $h \in C_b(\mathbb{R}^n)$, (6.6) yields

$$\int h/\theta_{r-1}\, d\nu \,=\, \lim_{n\to\infty} \int h/\theta_{r-1}\, d\nu_{j_n} \,=\, \lim_{n\to\infty} \int h\, d\mu_{j_n} \,=\, \int h\, d\mu$$

for all $h \in C_b(\mathbb{R}^n)$.

As both μ and $\theta_{r-1}^{-1} d\nu$ are finite measures, this implies that

$$\mu(B) \,:=\, \int_B (1/\theta_{r-1})\, d\nu, \qquad B \in \mathcal{B}. \tag{6.7}$$

As the subsequence $\{j_n\}$ was arbitrary, it thus follows that the whole subsequence $\{\nu_{j_k}\}$ converges weakly to ν.

Next, let $f : \mathbb{R}^n \to \mathbb{R}$ be continuous and such that $f/\theta_{r-1} \in C_b(\mathbb{R}^n)$. As $k \to \infty$, from (6.6),

$$\int (f/\theta_{r-1})\, d\nu_{j_k} \,\to\, \int (f/\theta_{r-1})\, d\nu,$$

and so

$$\begin{aligned}\int f\, d\mu_{j_k} &= \int (f/\theta_{r-1})\, \theta_{r-1}\, d\mu_{j_k} = \int (f/\theta_{r-1})\, d\nu_{j_k}\\ &\to \int (f/\theta_{r-1})\, d\nu = \int f\, d\mu, \quad \text{[by (6.7)]},\end{aligned}$$

the desired result. □

LEMMA 6.6. *Let μ be a measure on $\mathbb{R}^n$ (with $\mu(\mathbb{R}^n) = y_0$) be such that*

$$\sup_{i=1,\ldots,n}\ \sup_{0\leq k\leq r} \int x_i^{2k}\, d\mu \leq S. \tag{6.8}$$

Then,

$$\sup_{\alpha\in\mathbb{N}^n;|\alpha|\leq r} \left| \int x^\alpha\, d\mu \right| \leq \sqrt{y_0 S}. \tag{6.9}$$

Proof. Let $\mathbf{y} = \{y_\alpha\}_{|\alpha|\leq 2r}$ be the sequence of moments, up to order $2r$, of the measure μ, and let $M_r(\mathbf{y})$ be the moment matrix defined in (2.3), associated with μ. Then, (6.8) means that those diagonal elements of $M_r(\mathbf{y})$, denoted $y_{2k}^{(i)}$ in Lasserre [11], are all bounded by S. Therefore, by Lemma 6.2 in [11], all diagonal elements of $M_r(\mathbf{y})$ are also bounded by S, i.e.,

$$y_{2\alpha} \leq S \quad \forall \alpha \in \mathbb{N}^n,\ |\alpha| \leq r, \tag{6.10}$$

and so are all elements of $M_r(\mathbf{y})$ (because $M_r(\mathbf{y}) \succeq 0$). Next, consider the two columns (and rows) 1 and j, associated with the monomials 1 and x^α, respectively, and with $|\alpha| \leq r$, that is, $M_r(\mathbf{y})(1,1) = y_0$ and $M_r(\mathbf{y})(1,j) = y_\alpha$. As $M_r(\mathbf{y}) \succeq 0$, we immediately have

$$M_r(\mathbf{y})(1,1) \times M_r(\mathbf{y})(j,j) \geq M_r(\mathbf{y})(1,j) M_r(\mathbf{y})(j,1) = M_r(\mathbf{y})(1,j)^2,$$

that is, $y_0 y_{2\alpha} \geq y_\alpha^2$. Using that $|\alpha| \leq r$ and (6.10), we obtain $y_0 S \geq y_\alpha^2$ for all α, $|\alpha| \leq r$, the desired result (6.9). □

REFERENCES

[1] E.J. ANDERSON AND P. NASH, *Linear Programming in Infinite-Dimensional Spaces*, John Wiley, Chichester, UK, 1987.

[2] R. ASH, *Real Analysis and Probability*, Academic Press, San Diego, 1972.

[3] C. BERG, J.P.R. CHRISTENSEN, AND P. RESSEL, *Positive definite functions on abelian semigroups*, Math. Ann., 223 (1976), pp. 253–274.

[4] C. BERG, *The multidimensional moment problem and semi-groups*, Proc. Sympos. Appl. Math., 37 (1980), pp. 110–124.

[5] P. BILLINGSLEY, *Convergence of Probability Measures*, John Wiley, New York, 1968.

[6] T. JACOBI AND A. PRESTEL, *Distinguished representations of strictly positive polynomials*, J. Reine. Angew. Math., 532 (2001), pp. 223–235.

[7] O. HERNÁNDEZ-LERMA AND J.B. LASSERRE, *Markov Chains and Invariant Probabilities*, Birkhäuser Verlag, Basel, 2003.

[8] J.B. LASSERRE, *Global optimization with polynomials and the problem of moments*, SIAM J. Optim., 11 (2001), pp. 796–817.

[9] J.B. LASSERRE, *Polynomials nonnegative on a grid and discrete optimization*, Trans. Amer. Math. Soc., 354 (2002), pp. 631–649.

[10] J.B. LASSERRE, *Semidefinite programming vs. LP relaxations for polynomial programming*, Math. Oper. Res., 27 (2002), pp. 347–360.

[11] J.B. Lasserre, *A sum of squares approximation of nonnegative polynomials*, SIAM J. Optim., to appear.

[12] M. Marshall, *Approximating positive polynomials using sums of squares*, Canad. Math. Bull., 46 (2003), pp. 400–418.

[13] A.E. Nussbaum, *Quasi-analytic vectors*, Ark. Mat., 6 (1966), pp. 179–191.

[14] P.A. Parrilo, *Semidefinite programming relaxations for semialgebraic problems*, Math. Progr. Ser. B, 96 (2003), pp. 293–320.

[15] A. Prestel and C.N. Delzell, *Positive Polynomials*, Springer, Berlin, 2001.

[16] M. Putinar, *Positive polynomials on compact semi-algebraic sets*, Indiana Univ. Math. J., 42 (1993), pp. 969–984.

[17] A.P. Robertson and W. Robertson, *Topological Vector Spaces*, Cambridge University Press, Cambridge, UK, 1964.

[18] C. Scheiderer, *Positivity and sums of squares: A guide to some recent results*, Department of Mathematics, University of Duisburg, Germany, 2003.

[19] C. Scheiderer, *Sums of squares on real algebraic curves*, Math. Z., 245 (2003), pp. 725–760.

[20] K. Schmüdgen, *The K-moment problem for compact semi-algebraic sets*, Math. Ann., 289 (1991), pp. 203–206.

[21] M. Schweighofer, *Optimization of polynomials on compact semialgebraic sets*, SIAM J. Optim., 15 (2005), pp. 805–825.

[22] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95.

SIAM J. OPTIM.
Vol. 16, No. 3, pp. 629–649

DISTANCE TO SOLVABILITY/UNSOLVABILITY IN LINEAR OPTIMIZATION*

M. J. CÁNOVAS[†], M. A. LÓPEZ[‡], J. PARRA[†], AND F. J. TOLEDO[†]

Abstract. In this paper we measure how much a linear optimization problem, in $\mathbb{R}^n$, has to be perturbed in order to lose either its solvability (i.e., the existence of optimal solutions) or its unsolvability property. In other words, if we consider as ill-posed those problems in the boundary of the set of solvable ones, then we can say that this paper deals with the associated distance to ill-posedness. Our parameter space is the set of all the linear semi-infinite programming problems with a fixed, but arbitrary, index set. In this framework, which includes as a particular case the ordinary linear programming, we obtain a formula for the distance from a solvable problem to unsolvability in terms of the nominal problem's coefficients. Moreover, this formula also provides the exact expression, or a lower bound, of the distance from an unsolvable problem to solvability. The relationship between the solvability and the primal-dual consistency is analyzed in the semi-infinite context, underlining the differences with the finite case.

Key words. stability, ill-posedness, distance to ill-posedness, solvability, duality, linear semi-infinite programming

AMS subject classifications. 65F22, 90C34, 15A39, 52A40

DOI. 10.1137/040612981

1. Introduction.

Different concepts of distance to ill-posedness have recently acquired remarkable prominence in different settings related to linear programming. Besides providing quantitative measures of the stability of a problem, they are related to several theoretical and numerical issues, namely, stability of the feasible set [3], [6], [19]; measures of conditioning [10], [17], [20]; complexity analysis of certain algorithms for computing solutions [9], [11]; size of the feasible set [3], [8]; metric regularity of mappings [6], [7], [15]; etc.

An instance of a problem is *ill-posed* with respect to a certain property if arbitrarily small perturbations of the data defining the problem instance can yield problem instances with and without the property. In this way, the respective boundaries of the sets of consistent problems (i.e., with nonempty solution set), bounded problems (i.e., with finite optimal value), or solvable problems (i.e., having optimal solutions) can be seen as examples of sets of ill-posed problems. The distance from a problem to any of these boundaries is referred to as its *distance to ill-posedness* with respect to the considered property.

The distance to ill-posedness with respect to consistency has been thoroughly studied, for example, in the contexts of conic linear systems [16], [18], [19] and linear semi-infinite inequality systems [3].

This paper is concerned with the distance to ill-posedness with respect to the boundedness and solvability of linear optimization problems, in $\mathbb{R}^n$, of the following

*Received by the editors August 6, 2004; accepted for publication (in revised form) May 12, 2005; published electronically January 6, 2006. This research has been partially supported by grants BFM2002-04114-C02 (01-02) from MCYT (Spain) and FEDER (E.U.), and GV04B-648 and GRUPOS04/79 from Generalitat Valenciana (Spain).

http://www.siam.org/journals/siopt/16-3/61298.html

[†] Operations Research Center, Miguel Hernández University of Elche, 03202 Elche (Alicante), Spain (canovas@umh.es, parra@umh.es, javier.toledo@umh.es).

[‡] Department of Statistics and Operations Research, University of Alicante, 03071 Alicante, Spain (marco.antonio@ua.es).

form:

$$\pi: \begin{array}{ll} \text{Inf} & c'x \\ \text{s.t.} & a_t'x \geq b_t,\ t \in T, \end{array} \tag{1}$$

where c, x, $a_t \in \mathbb{R}^n$, $b_t \in \mathbb{R}$, and y' denotes the transpose of $y \in \mathbb{R}^n$. The *index set*, T, of the *constraint system*, $\sigma = \{a_t'x \geq b_t\ ,\ t \in T\}$, is arbitrary. The *feasible set* of π is denoted by F, its *optimal value* by v, and its *optimal set* by F^{op}, adopting the convention $v = +\infty$ when $F = \emptyset$.

When T is finite π is nothing else but an *ordinary linear programming problem*, whereas π is a *linear semi-infinite programming problem* when T is infinite. In the latter case the problem of determining the distance to the ill-posedness with respect to the solvability is not reducible to the problem of determining the distance to the ill-posedness with respect the consistency of the combined system of primal and dual constraints [5], since the dual problem is infinite-dimensional and there might exist a duality gap (see [12, Chap. 8]). Nevertheless, in section 6 the relationship between solvability and primal-dual consistency is explored.

The *parameter space* of all the linear optimization problems $\pi = (c, \sigma)$ in the form (1), and whose constraint systems have the same index set T, is denoted by Π. When different problems are considered in Π, they and their associated elements will be distinguished by means of sub- or superscripts. Thus, if π_1 also belongs to Π, we write $\pi_1 = (c^1, \sigma_1)$ and $\sigma_1 := \{(a_t^1)'x \geq b_t^1\ ,\ t \in T\}$, and its feasible set, optimal value, and optimal set are accordingly denoted by F_1, v_1, and F_1^{op}, respectively.

Π_c will denote the subset of Π formed by all the *consistent problems*, while $\Pi_i := \Pi \backslash \Pi_c$ represents the subset of all the *inconsistent problems*. Π_b denotes the subset of the *bounded problems*, and Π_s the subset of the *solvable* ones, that is, those problems with nonempty optimal set ($F^{op} \neq \emptyset$). Obviously $\Pi_s \subset \Pi_b \subset \Pi_c$.

Associated with two arbitrary norms in $\mathbb{R}^n$ and $\mathbb{R}^{n+1}$, both denoted by $\|\cdot\|$, the *extended distance* $\delta : \Pi \times \Pi \to [0, +\infty]$ given by

$$\delta(\pi_1, \pi) := \max\left\{\left\|c^1 - c\right\|, d(\sigma_1, \sigma)\right\}, \tag{2}$$

where

$$d(\sigma_1, \sigma) := \sup_{t \in T} \left\| \begin{pmatrix} a_t^1 \\ b_t^1 \end{pmatrix} - \begin{pmatrix} a_t \\ b_t \end{pmatrix} \right\|.$$

This extended distance endows Π with the topology of the uniform convergence of the coefficients vectors (see [12, Chapter 10] for details). The space Π locally behaves as a normed space.

Given $\pi \in \Pi$ and $\widetilde{\Pi} \subset \Pi$, we will write, as usual,

$$\delta(\pi, \widetilde{\Pi}) := \inf\left\{\delta(\pi, \widetilde{\pi})\ ,\ \widetilde{\pi} \in \widetilde{\Pi}\right\} \in [0, +\infty].$$

If $\varnothing \neq \widetilde{\Pi}$ and $\pi \notin \widetilde{\Pi}$, one has $\delta(\pi, \widetilde{\Pi}) = \delta(\pi, bd(\widetilde{\Pi}))$.

If X is a subset of any topological space, $int(X)$, $cl(X)$, and $bd(X)$ denote the *interior set*, the *closure*, and the *boundary* of X, respectively. By $ext(X)$ we represent the *exterior* of X, i.e., the complementary set of $cl(X)$.

In [4, Thm. 1], it is proved that the set of ill-posed problems with respect to solvability, $bd(\Pi_s)$, coincides with $bd(\Pi_b)$. Moreover, this set is characterized there by means of some results which we gather, among other preliminaries, in section 2.

In fact, in order to achieve our goal of finding an expression for the distance to ill-posedness $\delta(\pi, bd(\Pi_s))$, we appeal to a collection of results about the stability and well-posedness established in [2], [3], [4], [12], and [13]. In section 3 we provide (via Theorems 1 and 2) an explicit formula (5) for the distance to unsolvability from a solvable problem. Specifically, this formula consists of the minimum of two distances in $\mathbb{R}^n$ and $\mathbb{R}^{n+1}$, respectively, which depend only on the problem's data. We point out that the first of these distances turns out to be the distance to (primal) inconsistency for the given problem, whereas the second one is, as we prove in Theorem 6, the distance to dual inconsistency. On the other hand, Theorems 3 and 4 establish that the previous formula can be extended to certain unsolvable problems. Theorem 1 gathers all the cases for which the formula holds. In the remaining cases we show that the right-hand side of (5) still stands as a lower bound for the distance to ill-posedness, and a general upper bound is also given (Theorem 5), again in terms of the problem's data. Section 4 provides more precise upper bounds under certain additional hypotheses. Section 5 is devoted to presenting some examples and counterexamples which delimit and illustrate the main results of the paper. Specifically, Examples 5 and 6 show the difficulties in providing a formula of $\delta(\pi, bd(\Pi_s))$ when Theorem 1 does not apply. Section 6 approaches the ill-posedness with respect to the dual consistency and analyzes the relationship between the ill-posedness with respect to the solvability and with respect to primal-dual consistency. This section, together with section 7, integrates the contributions of the paper within the related literature on conditioning in linear optimization, paying attention to the backgrounds in the finite case (with T finite) traced from [5], [10], and [20]. We emphasize the differences between the finite case and the general one (T arbitrary). These two last sections show how formula (5) generalizes to our semi-infinite context the corresponding result for finite solvable linear programming problems. Moreover, (5) extends to a certain subset of unsolvable problems, providing new results even for finite linear programming.

2. Preliminaries. This section presents the necessary notation and some basic definitions, results, and tools used in this paper. Given $\varnothing \neq X \subset \mathbb{R}^k$, $conv(X)$ and $cone(X)$ denote the *convex hull* of X and the *conical convex hull* of X, respectively. It is assumed that $cone\,(X)$ always contains the zero-vector 0_k, and thus $cone(\varnothing) = \{0_k\}$. If $\Lambda \subset \mathbb{R}$, we introduce the set $\Lambda X := \{\lambda x : \lambda \in \Lambda \text{ and } x \in X\}$.

If we consider any norm in $\mathbb{R}^k$, $\|.\|$, the corresponding *open unit ball* will be represented by B. Given a sequence $\{\mu_r\}$, $\lim_r \mu_r$ should be interpreted as $\lim_{r\to+\infty} \mu_r$.

Associated with $\pi = (c, \sigma)$, the following sets are relevant in our analysis:

$$A := conv\left(\{a_t,\ t \in T\}\right), \qquad M := cone\left(\{a_t,\ t \in T\}\right) = \mathbb{R}_+ A,$$

$$Z^+ := conv\left(\{a_t,\ t \in T;\, c\}\right), \qquad Z^- := conv\left(\{a_t\,,\ t \in T\,;\ -c\}\right),$$

$$C := conv\left(\left\{\begin{pmatrix} a_t \\ b_t \end{pmatrix}, t \in T\right\}\right), \quad H := C + \mathbb{R}_+\left\{\begin{pmatrix} 0_n \\ -1 \end{pmatrix}\right\},$$

where $\mathbb{R}_+ := [0, +\infty[$. The sets M and H are, respectively, called the *first moment cone* and the *hypographical set*.

The existence of infinitely many coefficient vectors when T is infinite gives rise to the following pathological subset of problems (see [3, sect. 3]):

$$\Pi_\infty := \{\pi \in \Pi \mid \delta(\pi, bd(\Pi_c)) = +\infty\}.$$

The problems in Π_∞ are characterized by the property that $\binom{0_n}{1}$ belongs to the recession cone of $cl(C)$; in other words, $\binom{0_n}{1} = \lim_r \mu_r z^r$, with $\{z^r\}_{r=1}^\infty \subset C$ and $\{\mu_r\} \downarrow 0$. Moreover, $\Pi_\infty \subset \Pi_i$.

The following proposition gathers different results which are applied throughout the paper.

Proposition 1. *Given $\pi = (c, \sigma) \in \Pi$, the following statements hold:*

(i) [2, *Lem.* 4.1] *If $\pi \in int\,(\Pi_c)$, then $\pi \in int\,(\Pi_s)$ if and only if $c \in int\,(M)$.*

(ii) [12, *Thm.* 8.1(iv)] *If $\pi \in \Pi_b$, then $c \in cl\,(M)$.*

(iii) [4, *Lem.* 1(iv)] *If $\pi \in \Pi_c$ and $c \in cl\,(M)$, then $\pi \in cl\,(\Pi_s)$.*

(iv) [4, *Lem.* 1(i) *and Prop.* 5] *If $\pi \in bd\,(\Pi_c) \cup (\Pi_i \backslash \Pi_\infty)$, then $0_n \in cl\,(A)$.*

(v) [4, *Lem.* 1(ii)] *If $\pi \in bd\,(\Pi_c) \cap \Pi_i$, then $0_n \in bd\,(A)$.*

(vi) [12, *Thm.* 6.3] *If $\pi \in \Pi_i$ and $M = \mathbb{R}^n$, then $\pi \in int\,(\Pi_i)$.*

Along the lines of [8] and [10] (which deal with conic linear systems), $bd(\Pi_c)$ is considered as the set of *ill-posed problems* with respect to the consistency, and according to [19], the *distance to ill-posedness* is $\delta(\pi, bd(\Pi_c))$. The following proposition describes the position of $\pi \in \Pi$ relative to $bd\,(\Pi_c)$ in terms of the relative position between 0_{n+1} and the boundary of the hypographical set, $bd\,(H)$.

Proposition 2 (see [3, Thms. 4, 5, and 6]). *Let $\pi \in \Pi \diagdown \Pi_\infty$. Then, the following statements hold:*

(i) $\pi \in int\,(\Pi_i) \Leftrightarrow 0_{n+1} \in int\,(H)$;

(ii) $\pi \in int\,(\Pi_c) \Leftrightarrow 0_{n+1} \in ext(H)$;

(iii) $\pi \in bd\,(\Pi_c) \Leftrightarrow 0_{n+1} \in bd(H)$;

(iv) $\delta(\pi, bd(\Pi_c)) = d\,(0_{n+1}, bd\,(H))$.

Observe that (iv) translates the problem of measuring the distance to ill-posedness with respect to the consistency, posed in the infinite-dimensional space Π, into the problem of calculating a distance in the $(n+1)$-dimensional Euclidean space.

The following proposition describes the position of $\pi \in int\,(\Pi_c)$ relative to $bd\,(\Pi_s)$ in terms of the relative position between 0_n and the boundary of the set Z^-.

Proposition 3 (see [4, Thm. 2]). *Given $\pi \in int\,(\Pi_c)$, one has*

(i) $\pi \in int\,(\Pi_s) \Leftrightarrow 0_n \in int(Z^-)$;

(ii) $\pi \in bd\,(\Pi_s) \Leftrightarrow 0_n \in bd(Z^-)$;

(iii) $\pi \in ext\,(\Pi_s) \Leftrightarrow 0_n \in ext\,(Z^-)$.

The next result characterizes those problems that, being ill-posed with respect to the consistency, are also ill-posed with respect to the solvability.

Proposition 4 (see [4, Thm. 3]). *Let $\pi \in bd\,(\Pi_c)$. Then $\pi \in bd\,(\Pi_s)$ if and only if either $\pi \in cl\,(bd\,(\Pi_c) \cap \Pi_c)$ or $0_n \in bd(Z^+)$.*

The following proposition explores the relationship between the condition $\pi \in cl\,(bd\,(\Pi_c) \cap \Pi_c)$ and the data-set C.

Proposition 5 (see [4, Thm. 4]). *Let $\pi \in bd\,(\Pi_c)$. If $\pi \in cl\,(bd\,(\Pi_c) \cap \Pi_c)$, then $0_{n+1} \in bd\,(C)$. The converse statement holds when $\{b_t\,,\, t \in T\}$ is bounded.*

The following proposition, which is a straightforward consequence of Propositions 3, 4, and 5, provides a complete characterization of the ill-posed problems whose constraint systems have a bounded right-hand side.

Proposition 6 (see [4, Thm. 5]). *Let $\pi \in \Pi$, and suppose that the set $\{b_t\,,\, t \in T\}$ is bounded. Then, $\pi \in bd\,(\Pi_s)$ if and only if some of the following statements hold:*

(i) $0_{n+1} \in ext\,(H)$ *and* $0_n \in bd\,(Z^-)$;

(ii) $0_{n+1} \in bd\,(H) \cap bd\,(C)$;

(iii) $0_{n+1} \in bd\,(H)$ *and* $0_n \in bd\,(Z^+)$.

The following results admit straightforward proofs.

Proposition 7. *The sets M, A, Z^+, and Z^- satisfy the following relations:*

$$c \in int\,(M) \Leftrightarrow 0_n \in int\left(Z^-\right) \text{ and } -c \in int\,(M) \Leftrightarrow 0_n \in int\left(Z^+\right).$$

In particular, if $c = 0_n$, then $Z^+ = Z^-$ and

$$0_n \in int\left(Z^-\right) \Leftrightarrow 0_n \in int\,(A) \Leftrightarrow 0_n \in int\,(M) \Leftrightarrow M = \mathbb{R}^n.$$

Proposition 8. *Let $S \neq \varnothing$ be an arbitrary index set and let $X := \{x^s\,,\, s \in S\}$ and $Y := \{y^s\,,\, s \in S\}$ be two subsets of $\mathbb{R}^k$ such that $\sup_{s\in S}\|x^s - y^s\| \leq \varepsilon$ for certain $\varepsilon \geq 0$. Then one has the following:*

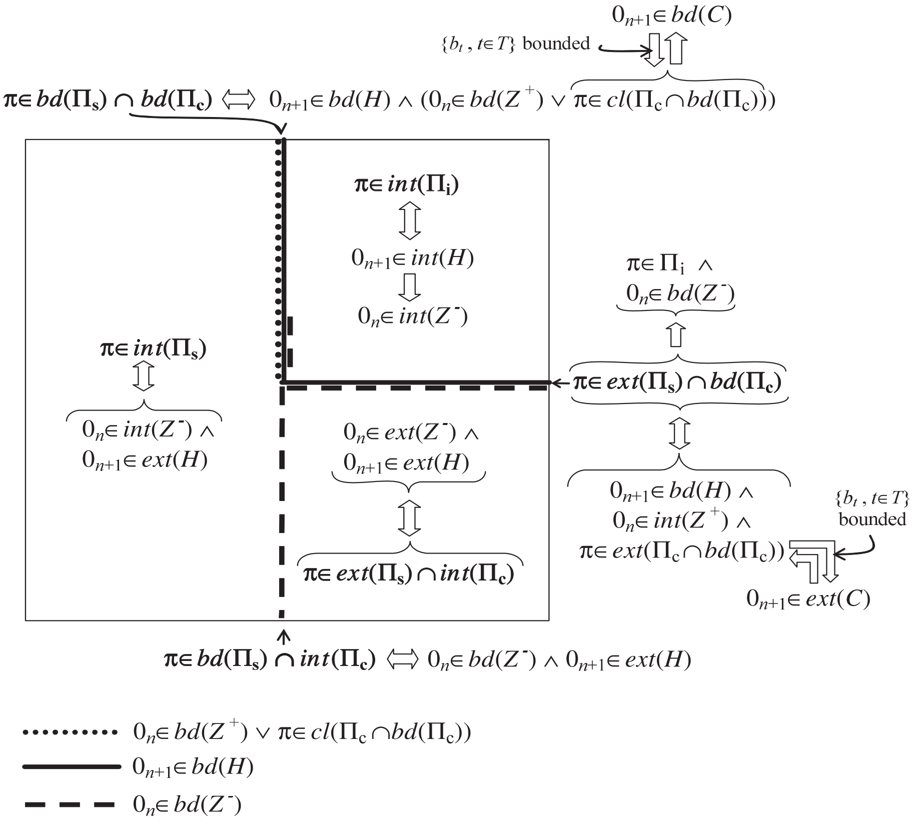
(i) *If $\rho cl\,(B) \subset cl\,(conv\,(X))$ for some $\rho \geq \varepsilon$, then*

$$(\rho - \varepsilon)\, cl\,(B) \subset cl\,(conv\,(Y)).$$

(ii) *If $\rho cl\,(B) \cap cl\,(conv\,(X)) = \varnothing$ for some $\rho \geq \varepsilon$, then*

$$(\rho - \varepsilon)\, cl\,(B) \cap cl\,(conv\,(Y)) = \varnothing.$$

Figure 1 summarizes the information we have already presented about the structure of $\Pi\backslash\Pi_\infty$ in relation to the properties of consistency and solvability.



$0_{n+1} \in bd(C)$
$\{b_t\,,\, t\in T\}$ bounded
$\pi \in bd(\Pi_s) \cap bd(\Pi_c) \Longleftrightarrow 0_{n+1} \in bd(H) \wedge (0_n \in bd(Z^+) \vee \pi \in cl(\Pi_c \cap bd(\Pi_c)))$
$\pi \in int(\Pi_i)$
$0_{n+1} \in int(H)$
$0_n \in int(Z^-)$
$\pi \in \Pi_i \wedge$
$0_n \in bd(Z^-)$
$\pi \in int(\Pi_s)$
$0_n \in int(Z^-) \wedge$
$0_{n+1} \in ext(H)$
$\pi \in ext(\Pi_s) \cap bd(\Pi_c)$
$0_n \in ext(Z^-) \wedge$
$0_{n+1} \in ext(H)$
$0_{n+1} \in bd(H) \wedge$
$0_n \in int(Z^+) \wedge$
$\pi \in ext(\Pi_c \cap bd(\Pi_c))$
$\{b_t\,,\, t\in T\}$ bounded
$\pi \in ext(\Pi_s) \cap int(\Pi_c)$
$0_{n+1} \in ext(C)$
$\pi \in bd(\Pi_s) \cap int(\Pi_c) \Longleftrightarrow 0_n \in bd(Z^-) \wedge 0_{n+1} \in ext(H)$
$0_n \in bd(Z^+) \vee \pi \in cl(\Pi_c \cap bd(\Pi_c))$
$0_{n+1} \in bd(H)$
$0_n \in bd(Z^-)$

Fig. 1. *Structure of* $\Pi\backslash\Pi_\infty$.

Throughout the paper we assume that the norm $\|\cdot\|$ considered in $\mathbb{R}^{n+1}$ verifies

$$\left\|\begin{pmatrix} a \\ b \end{pmatrix}\right\| = \left\|\begin{pmatrix} a \\ -b \end{pmatrix}\right\| \text{ for all } \begin{pmatrix} a \\ b \end{pmatrix} \in \mathbb{R}^{n+1}. \tag{3}$$

Observe that any p-norm, but not any norm (see [21, Thm. 15.2]), verifies this condition. In $\mathbb{R}^n$ the norm (also denoted by $\|\cdot\|$) given by

$$\|a\| := \left\|\begin{pmatrix} a \\ 0 \end{pmatrix}\right\| \text{ for all } a \in \mathbb{R}^n \tag{4}$$

will be considered. Note that if the norm considered in $\mathbb{R}^{n+1}$ is a p-norm, $p \in [1, +\infty]$, the norm in $\mathbb{R}^n$ is also a p-norm (with the same p).

Remark 1. Property (3) implies that $\|\binom{a}{b_1}\| \leq \|\binom{a}{b_2}\|$ when $|b_1| \leq |b_2|$. The proof is a straightforward consequence of the fact that $\binom{a}{b_1}$ is a convex combination of $\binom{a}{b_2}$ and $\binom{a}{-b_2}$.

3. Distance to solvability/unsolvability. In the present section we approach the problem of determining the distance to ill-posedness, $\delta(\pi, bd(\Pi_s))$, for a given problem $\pi \in \Pi \backslash \Pi_\infty$. The case $\pi \in \Pi_\infty$ is obvious as far as $\Pi_\infty \subset \Pi_i$, and then $\delta(\pi, bd(\Pi_s)) \geq \delta(\pi, bd(\Pi_c)) = +\infty$. We will analyze different cases obtaining either an exact expression (see Theorem 1) or lower and upper bounds (see Theorem 5 and the subsequent results) for this distance in terms of the problem's data.

The following theorem is the main result in this paper and partially synthesizes the statements of Theorems 2, 3, 4, and 5 in relation to the exact formula for the distance to ill-posedness mentioned in the previous paragraph.

THEOREM 1. *Let $\pi = (c, \sigma) \in \Pi \backslash \Pi_\infty$. Suppose that at least one of the following conditions holds:*

(i) $\pi \in cl(\Pi_s)$;

(ii) $\pi \in ext(\Pi_s)$ *and* $d(0_{n+1}, bd(H)) \neq d(0_n, bd(Z^-))$;

(iii) $d(0_{n+1}, bd(H)) = d(0_n, bd(Z^-)) \geq \|c\|$.

Then one has

$$\delta(\pi, bd(\Pi_s)) = \min\left\{d(0_{n+1}, bd(H)), d\left(0_n, bd\left(Z^-\right)\right)\right\}. \tag{5}$$

Proof. (i) See Theorem 2.

(ii) $\pi \in ext(\Pi_s)$ implies either $\pi \in int(\Pi_c)$, in which case Theorem 3 applies, or $\pi \in \Pi_i$, and Theorem 4 applies, since otherwise one has $\pi \in bd(\Pi_c) \cap \Pi_c$ and Proposition 4 yields a contradiction.

(iii) See Theorem 5. □

Remark 2. Note that, by virtue of Proposition 2, one has $d(0_{n+1}, bd(H)) = \delta(\pi, bd(\Pi_c))$; that is, the distance to ill-posedness with respect to the solvability depends on the distance to ill-posedness with respect to the consistency, as one would expect.

Remark 3. Formula (5) for the distance to ill-posedness does not hold, in general, in the remaining case corresponding to the problems $\pi \in ext(\Pi_s)$ such that

$$d(0_{n+1}, bd(H)) = d\left(0_n, bd\left(Z^-\right)\right) < \|c\|,$$

even in ordinary linear programming in $\mathbb{R}$ with "few" constraints, as we can see in Examples 3, 4, and 5. Nevertheless, in this case one has, as a straightforward consequence of Theorem 5, that

$$d(0_{n+1}, bd(H)) \leq \delta(\pi, bd(\Pi_s)) \leq \|c\|.$$

The next lemma will be used later on.

Lemma 1. *Given $\pi \in \Pi$, one has the following:*

(i) *If $0_n \in bd\,(A)$ and $c \in cl\,(M)$, then $0_n \in bd(Z^+)$.*

(ii) *If $\pi \in bd\,(\Pi_c)$ and $c \in cl\,(M)$, then $\pi \in bd\,(\Pi_s)$.*

(iii) *If $0_n \in int\,(Z^- \cap Z^+)$, then $0_n \in int\,(A)$.*

Proof. (i) Since $0_n \in bd\,(A)$, we have $0_n \in cl\,(Z^+)$. If $0_n \in int\,(Z^+)$, then $-c \in int\,(M)$ (see Proposition 7) and, since $c \in cl\,(M)$ by assumption, Theorem 6.1 in [21] ensures that $0_n \in int\,(M)$ and, then, we get the contradiction $0_n \in int\,(A)$ (again by Proposition 7).

(ii) We distinguish two cases. If $\pi \in \Pi_c$, one has $\pi \in bd\,(\Pi_s)$ by virtue of Proposition 4. If $\pi \in \Pi_i$, then $0_n \in bd\,(A)$ from Proposition 1(v); thus, the previous statement implies $0_n \in bd\,(Z^+)$ and therefore $\pi \in bd\,(\Pi_s)$, again by virtue of Proposition 4.

(iii) Under this hypothesis we have, by Proposition 7, $c \in int\,(M)$ and $-c \in int\,(M)$; thus, by convexity of $int\,(M)$, $0_n \in int\,(M)$ and Proposition 7 again leads us to $0_n \in int\,(A)$. □

The following theorem establishes that (5) is valid for a problem in the closure of the set of solvable problems.

Theorem 2. *Let $\pi \in cl\,(\Pi_s)$. Then* (5) *holds, i.e.,*

$$\delta\left(\pi, bd\,(\Pi_s)\right) = \min\left\{d\left(0_{n+1}, bd\,(H)\right), d\left(0_n, bd\left(Z^-\right)\right)\right\}.$$

Proof. First let us consider the case $\pi \in bd(\Pi_s)$ and let us see that the right-hand side in (5) is zero. Indeed, if $\pi \in int(\Pi_c)$, Proposition 3(ii) ensures that $d(0_n, bd(Z^-)) = 0$, and if $\pi \in bd(\Pi_c)$, Proposition 2(iii) guarantees that $d(0_{n+1}, bd(H)) = 0$.

From now on we will suppose $\pi \in int\,(\Pi_s)$. In order to establish the $\leq$ inequality, let us see that the following inequalities are simultaneously satisfied:

(a) $\delta\left(\pi, bd\,(\Pi_s)\right) \leq d\left(0_{n+1}, bd\,(H)\right)$, and

(b) $\delta\left(\pi, bd\,(\Pi_s)\right) \leq d\left(0_n, bd\,(Z^-)\right)$.

Since $\pi \in int\,(\Pi_s)$, we have

$$\delta\left(\pi, bd\,(\Pi_s)\right) \leq \delta\left(\pi, bd\,(\Pi_c)\right) = d\left(0_{n+1}, bd\,(H)\right)$$

and (a) holds. On the other hand, (b) is trivial if $Z^- = \mathbb{R}^n$. Otherwise, the distance $d\left(0_n, bd\,(Z^-)\right)$ will be attained at certain $a \in bd\,(Z^-)$. Consequently we have $0_n \in bd\,(Z^- - a)$.

If we consider the problem $\pi_0 := (c + a, \sigma_0)$, where

$$\sigma_0 := \left\{(a_t - a)' x \geq b_t,\ t \in T\right\},$$

then $Z_0^- = Z^- - a$ and Proposition 7 entails $c + a \notin int(M_0)$ (with $M_0 = cone(\{a_t - a, t \in T\})$). Now Proposition 1(i) ensures that $\pi_0 \notin int\,(\Pi_s)$. Therefore

$$\delta\left(\pi, bd\,(\Pi_s)\right) \leq \delta\left(\pi, \pi_0\right) = \|a\|,$$

which establishes (b).

Now let $\alpha := \min\left\{d\left(0_{n+1}, bd\,(H)\right), d\left(0_n, bd\,(Z^-)\right)\right\}$, and take as before a point $a \in bd\,(Z^-)$ at which the distance $d\left(0_n, bd\,(Z^-)\right)$ is attained, supposing for the moment that $Z^- \neq \mathbb{R}^n$. From (a) and (b) we have $\delta\left(\pi, bd\,(\Pi_s)\right) \leq \alpha$ (for all $\pi \in int\,(\Pi_s)$). To see that equality (5) holds, it is sufficient to prove that every problem $\pi_1 := (c^1, \sigma_1) \in \Pi$ such that $\delta\left(\pi, \pi_1\right) < \alpha$ is still in $int\,(\Pi_s)$.

Take $\pi_1 \in \Pi$ in the previous conditions. Since $\pi \in int(\Pi_c)$ and $\delta(\pi, \pi_1) < d(0_{n+1}, bd(H))$, we have $\pi_1 \in int(\Pi_c)$ (Proposition 2). On the other hand, since $\pi \in int(\Pi_s)$, one has $c \in int(M)$ by virtue of Proposition 1(i), and then $0_n \in int(Z^-)$ (by Proposition 7). Indeed $\|a\| \, cl(B) \subset cl(Z^-)$ and $\alpha \leq \|a\|$. Writing $\delta(\pi, \pi_1) = \alpha - \varepsilon$ for some $0 < \varepsilon \leq \alpha$, Proposition 8(i) entails $\varepsilon \, cl(B) \subset cl\left(Z_1^-\right)$, and then $0_n \in int(cl(Z_1^-)) = int(Z_1^-)$. So, taking into account that $\pi_1 \in int(\Pi_c)$, Proposition 3(i) ensures that $\pi_1 \in int(\Pi_s)$.

Finally, in the case $Z^- = \mathbb{R}^n$ one has $\alpha = d(0_{n+1}, bd(H))$, and then if $\delta(\pi, \pi_1) < \alpha$, one has $\pi_1 \in int(\Pi_c)$ and, trivially, $0_n \in int(Z_1^-)$, because $Z_1^- = \mathbb{R}^n$ (by Proposition 8). Thus $\pi_1 \in int(\Pi_s)$. □

Examples 1 and 2 illustrate formula (5) for the problem $\pi \in cl(\Pi_s)$. In the first example, one has $\delta(\pi, bd(\Pi_s)) = d(0_n, bd(Z^-)) < d(0_{n+1}, bd(H))$, while in the second example $\delta(\pi, bd(\Pi_s)) = d(0_{n+1}, bd(H)) < d(0_n, bd(Z^-))$. In both cases, a perturbation for obtaining a problem where the distance to ill-posedness is attained will be indicated.

Now we approach the distance to ill-posedness for the problem $\pi \in ext(\Pi_s) \cap int(\Pi_c)$.

Theorem 3. *Let* $\pi \in ext(\Pi_s) \cap int(\Pi_c)$. *Then*

(i) $d(0_n, bd(Z^-)) \leq d(0_{n+1}, bd(H))$;

(ii) $\delta(\pi, bd(\Pi_s)) \geq d(0_n, bd(Z^-))$;

(iii) *if* $d(0_n, bd(Z^-)) < d(0_{n+1}, bd(H))$, *one has*

$$\delta(\pi, bd(\Pi_s)) = d\left(0_n, bd\left(Z^-\right)\right).$$

Proof. (i) Since $\pi \in ext(\Pi_s) \cap int(\Pi_c)$, Proposition 3(iii) ensures that $0_n \in ext(Z^-)$. Take $a \in bd(Z^-)$ such that $d(0_n, bd(Z^-)) = \|a\|$.

For every $\binom{\overline{a}}{b} \in bd(H)$ we have $\overline{a} \in cl(A) \subset cl(Z^-)$ and $\|\binom{\overline{a}}{b}\| \geq \|\overline{a}\| \geq \|a\|$ (see Remark 1). The arbitrariness of $\binom{\overline{a}}{b}$ entails $d(0_{n+1}, bd(H)) \geq \|a\|$ as we aimed to prove.

(ii) $\pi \in ext(\Pi_s) \cap int(\Pi_c)$ entails, by virtue of Propositions 2(ii) and 3(iii), $0_{n+1} \in ext(H)$ and $0_n \in ext(Z^-)$. Let $\pi_1 \in \Pi$ with $\delta(\pi_1, \pi) < d(0_n, bd(Z^-))$; then Proposition 8(ii) and part (i) ensure that $0_n \in ext(Z_1^-)$ and $0_{n+1} \in ext(H_1)$. Now, again by Propositions 2(ii) and 3(iii), $\pi_1 \in ext(\Pi_s) \cap int(\Pi_c)$ and therefore $\delta(\pi, bd(\Pi_s)) \geq d(0_n, bd(Z^-))$.

(iii) From (ii) we need only prove $\delta(\pi, bd(\Pi_s)) \leq d(0_n, bd(Z^-))$. Take again $a \in bd(Z^-)$ such that $d(0_n, bd(Z^-)) = \|a\|$ and consider the same perturbation as in the proof of Theorem 2; i.e., we consider the problem $\pi_0 = (c + a, \sigma_0)$, where

$$\sigma_0 := \left\{(a_t - a)' x \geq b_t, \ t \in T\right\}.$$

On the one hand, we have $0_n \in bd(Z_0^-) = bd(Z^- - a)$. On the other hand, since $\pi \in int(\Pi_c)$, Proposition 2(ii) guarantees that $0_{n+1} \in ext(H)$, and the fact that $\delta(\pi, \pi_0) = \|a\| < d(0_{n+1}, bd(H))$ by the current assumption, together with Proposition 8(ii), ensures that $0_{n+1} \in ext(H_0)$. Again by virtue of Proposition 2(ii) one has $\pi_0 \in int(\Pi_c)$. Therefore $\pi_0 \in bd(\Pi_s)$ by Proposition 3(ii) and, consequently, $\delta(\pi, bd(\Pi_s)) \leq \delta(\pi, \pi_0) = \|a\|$. □

Example 3 shows that formula (5) for the problem $\pi \in ext(\Pi_s) \cap int(\Pi_c)$ does not hold in general when $d(0_n, bd(Z^-)) = d(0_{n+1}, bd(H))$.

The next result is devoted to approaching the distance $\delta(\pi, bd(\Pi_s))$ for the problem $\pi \in \Pi_i \backslash \Pi_\infty$.

THEOREM 4. *Let* $\pi \in \Pi_i \backslash \Pi_\infty$*. Then*
(i) $d\left(0_{n+1}, bd\left(H\right)\right) \leq d\left(0_n, bd\left(Z^-\right)\right)$*;*
(ii) $\delta\left(\pi, bd\left(\Pi_s\right)\right) \geq d\left(0_{n+1}, bd\left(H\right)\right)$*;*
(iii) *if* $d\left(0_{n+1}, bd\left(H\right)\right) < d\left(0_n, bd\left(Z^-\right)\right)$*, one has*

$$\delta\left(\pi, bd\left(\Pi_s\right)\right) = d\left(0_{n+1}, bd\left(H\right)\right).$$

Proof. (i) If $\pi \in \Pi_i \backslash \Pi_\infty$, Proposition 1(iv) ensures that $0_n \in cl\left(A\right) \subset cl\left(Z^-\right)$. Moreover, Proposition 2 implies $0_{n+1} \in cl\left(H\right)$. Thus,

$$\begin{aligned} d\left(0_{n+1}, bd\left(H\right)\right) &= d\left(0_{n+1}, ext\left(H\right)\right) \leq d\left(0_{n+1}, ext\left(A \times \mathbb{R}\right)\right) \\ &= d\left(0_n, ext\left(A\right)\right) \leq d\left(0_n, ext\left(Z^-\right)\right) \\ &= d\left(0_n, bd\left(Z^-\right)\right). \end{aligned}$$

(ii) It is immediate as far as $\Pi_s \subset \Pi_c$, and then if $\pi \in \Pi_i \backslash \Pi_\infty$, one has, by virtue of Proposition 2(iv),

$$\delta\left(\pi, bd\left(\Pi_s\right)\right) \geq \delta\left(\pi, bd\left(\Pi_c\right)\right) = d\left(0_{n+1}, bd\left(H\right)\right).$$

(iii) From the previous statement it is sufficient to prove that

$$\delta\left(\pi, bd\left(\Pi_s\right)\right) \leq d\left(0_{n+1}, bd\left(H\right)\right).$$

If $H = \mathbb{R}^{n+1}$, the reader can easily prove that $\pi \in \Pi_\infty$, and thus we can take $\binom{a}{b} \in bd\left(H\right)$ such that $d\left(0_{n+1}, bd\left(H\right)\right) = \left\|\binom{a}{b}\right\|$. Consider, then, the problem $\pi_1 := \left(c + a, \sigma_1\right)$, where

$$\sigma_1 := \left\{\left(a_t - a\right)' x \geq b_t - b;\ t \in T\right\}.$$

Let us see that $\pi_1 \in bd\left(\Pi_s\right)$. We have $\pi_1 \in bd\left(\Pi_c\right)$ (see Proposition 2(iii)). We shall distinguish two possibilities:
(a) In the case when $\pi_1 \in \Pi_c$, Proposition 4 ensures that $\pi_1 \in bd\left(\Pi_s\right)$.
(b) If $\pi_1 \in \Pi_i$, by Proposition 1(v) $0_n \in bd\left(A_1\right) \subset cl(Z_1^+)$.

Let us proceed by supposing that, in the latter case, $0_n \in int(Z_1^+)$. Since $\pi \in \Pi_i$, Proposition 1(iv) gives $0_n \in cl\left(A\right) \subset cl\left(Z^-\right)$ and, because of the current assumption

$$\|a\| \leq \left\|\begin{pmatrix} a \\ b \end{pmatrix}\right\| < d\left(0_n, bd\left(Z^-\right)\right),$$

it must be the case that $a \in int\left(Z^-\right)$ and, then, $0_n \in int\left(Z_1^-\right)$. Thus, from Lemma 1(iii) we obtain $0_n \in int\left(A_1\right)$, which is a contradiction and makes us conclude $0_n \in bd(Z_1^+)$. Therefore $\pi_1 \in bd\left(\Pi_s\right)$, again by virtue of Proposition 4. Thus, we have

$$\delta\left(\pi, bd\left(\Pi_s\right)\right) \leq \delta\left(\pi, \pi_1\right) = \max\left\{\|a\|, \left\|\begin{pmatrix} a \\ b \end{pmatrix}\right\|\right\} = \left\|\begin{pmatrix} a \\ b \end{pmatrix}\right\|. \qquad \square$$

Example 4 shows that formula (5) for a problem $\pi \in \Pi_i \backslash \Pi_\infty$ does not hold in general when $d\left(0_{n+1}, bd\left(H\right)\right) = d\left(0_n, bd\left(Z^-\right)\right)$.

Now we establish an upper bound for the distance to ill-posedness, which coincides with it under certain conditions. We emphasize the fact that this bound works for any $\pi \in \Pi \backslash \Pi_\infty$.

PROPOSITION 9. *Let* $\pi = (c, \sigma) \in \Pi \backslash \Pi_\infty$. *Then*

$$\delta\left(\pi, bd\left(\Pi_s\right)\right) \leq \max\left\{d\left(0_{n+1}, bd\left(H\right)\right), \|c\|\right\}.$$

Proof. We know that $H \neq \mathbb{R}^{n+1}$ because $\pi \in \Pi \backslash \Pi_\infty$. Let $\binom{a}{b} \in bd\left(H\right)$ be such that $d\left(0_{n+1}, bd\left(H\right)\right) = \|\binom{a}{b}\|$. Consider the problem $\pi_2 := (0_n, \sigma_2)$, where

$$\sigma_2 := \left\{(a_t - a)' x \geq b_t - b;\ t \in T\right\}.$$

Let us see that $\pi_2 \in bd\left(\Pi_s\right)$. We have that $\pi_2 \in bd\left(\Pi_c\right)$ (see Proposition 2(iii)). In the case that $\pi_2 \in \Pi_c$, Proposition 4 ensures that $\pi_2 \in bd(\Pi_s)$. Assuming $\pi_2 \in \Pi_i$, Proposition 1(v) ensures $0_n \in bd\left(A_2\right) \subset cl(Z_2^+)$. If $0_n \in int(Z_2^+)$, the second statement in Proposition 7 entails $0_n \in int\left(A_2\right)$, which is a contradiction. Thus, $0_n \in bd(Z_2^+)$, and then $\pi_2 \in bd(\Pi_s)$, again by Proposition 4. Thus we have

$$\delta\left(\pi, bd\left(\Pi_s\right)\right) \leq \delta\left(\pi, \pi_2\right) = \max\left\{d\left(0_{n+1}, bd\left(H\right)\right), \|c\|\right\}. \qquad \square$$

The following result can be obtained as a straightforward consequence of the previous statements and establishes bounds on the distance to ill-posedness.

THEOREM 5. *Let* $\pi = (c, \sigma) \in \Pi \backslash \Pi_\infty$. *If we denote*

$$\alpha := \min\left\{d\left(0_{n+1}, bd\left(H\right)\right), d\left(0_n, bd\left(Z^-\right)\right)\right\} \text{ and}$$
$$\beta := \max\left\{d\left(0_{n+1}, bd\left(H\right)\right), \|c\|\right\},$$

then we have

$$\alpha \leq \delta\left(\pi, bd\left(\Pi_s\right)\right) \leq \beta.$$

Figure 2 synthesizes the main results about the distance to ill-posedness with respect to the solvability provided in this paper. The reader should analyze the information provided in this figure together with that given in Figure 1. We use the same notation as in Theorem 5 for α and β.

4. Other upper bounds. The goal of the following result is to obtain, under additional hypotheses, some refinements of the upper bound given in Proposition 9.

PROPOSITION 10. *Let* $\pi = (c, \sigma) \in ext\left(\Pi_s\right)$. *The following statements hold:*

(i) *If* $\pi \in cl\left(\Pi_c\right)$, *then*

$$\delta\left(\pi, bd\left(\Pi_s\right)\right) \leq d\left(c, bd\left(M\right)\right) \leq \|c\|.$$

(ii) *If* $\pi \in bd\left(\Pi_c\right)$, *then*

$$\delta\left(\pi, bd\left(\Pi_s\right)\right) \leq d\left(c, bd\left(-M\right)\right) \leq d\left(c, bd\left(M\right)\right).$$

(iii) *If* $\pi \in bd\left(\Pi_c\right)$ *and there exists* $t_0 \in T$ *such that* $a_{t_0} = 0_n$, *then*

$$\delta\left(\pi, bd\left(\Pi_s\right)\right) \leq d\left(0_n, bd\left(Z^+\right)\right) \leq d\left(c, bd\left(-M\right)\right).$$

Proof. (i) First observe that $M \neq \mathbb{R}^n$ since, otherwise, either $\pi \in cl(\Pi_s)$ (when $\pi \in \Pi_c$ and applying Proposition 1(iii)) or $\pi \in int(\Pi_i)$ (when $\pi \in \Pi_i$ and applying Proposition 1(vi)).

Let $a \in bd\left(M\right)$ such that $d\left(c, bd\left(M\right)\right) = \|c - a\|$ and consider the problem $\pi_3 := (a, \sigma)$. We have $\pi_3 \in cl\left(\Pi_c\right)$ and $a \in bd\left(M_3\right) = bd\left(M\right)$. If $\pi_3 \in \Pi_c$, Proposition 1(iii)

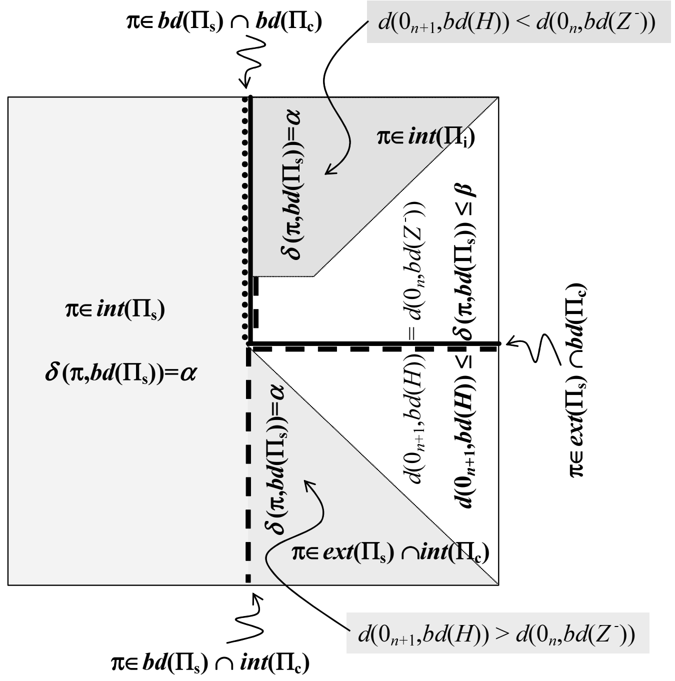


FIG. 2. *Distance to ill-posedness in* $\Pi\backslash\Pi_\infty$.

ensures that $\pi_3 \in cl\,(\Pi_s)$. If $\pi_3 \in bd\,(\Pi_c)$, then $\pi_3 \in bd\,(\Pi_s)$ by Lemma 1(ii). In any case, $\pi_3 \in cl\,(\Pi_s)$, and then

$$\delta\,(\pi, bd\,(\Pi_s)) \leq \delta\,(\pi, \pi_3) = \|c - a\| = d\,(c, bd\,(M))\,.$$

Finally, and since $M \neq \mathbb{R}^n$, one has $0_n \in bd\,(M)$, which implies

$$d\,(c, bd\,(M)) \leq \|c\|\,.$$

(ii) Obviously we have again $M \neq \mathbb{R}^n$. Let $d \in bd\,(-M)$ satisfying that $d\,(c, bd\,(-M)) = \|c - d\|$ and consider the problem $\pi_4 := (d, \sigma)$. Obviously $\pi_4 \in bd\,(\Pi_c)$ and we see that $\pi_4 \in bd\,(\Pi_s)$. In the case when $\pi_4 \in \Pi_c$, Proposition 4 ensures $\pi_4 \in bd\,(\Pi_s)$. If $\pi_4 \in \Pi_i$, from Proposition 1(v) one has $0_n \in bd\,(A_4) \subset cl(Z_4^+)$.

If $0_n \in int(Z_4^+)$, then $-d \in int\,(M_4) = int\,(M)$ (see Proposition 7), i.e., $d \in int\,(-M)$, which is a contradiction. Then $0_n \in bd(Z_4^+)$ and, again by Proposition 4, $\pi_4 \in bd\,(\Pi_s)$. So, in any case, $\pi_4 \in bd\,(\Pi_s)$, and then

$$\delta\,(\pi, bd\,(\Pi_s)) \leq \delta\,(\pi, \pi_4) = \|c - d\| = d\,(c, bd\,(-M))\,.$$

Because of Proposition 1(iv) we have $0_n \in cl\,(A) \subset cl\,(Z^+)$, and $0_n \in bd\,(Z^+)$ leads us to the contradiction $\pi \in bd(\Pi_s)$ (by Proposition 4). Hence we have that, under the current hypothesis, $0_n \in int\,(Z^+)$ or, equivalently, $-c \in int\,(M) \Leftrightarrow c \in int\,(-M)$. Since $M \neq \mathbb{R}^n$, there will exist $u \neq 0_n$ such that M and $-M$ are, respectively, contained in the half-spaces $S_+ := \{x \in \mathbb{R}^n \mid u'x \leq 0\}$ and $S_- := \{x \in \mathbb{R}^n \mid u'x \geq 0\}$. Therefore

$$d\,(c, bd\,(-M)) \leq d\,(c, S_+) \leq d\,(c, bd\,(M))\,.$$

(iii) Remember that under the current hypotheses $0_n \in int\,(Z^+)$. Take $a \in bd\,(Z^+)$ such that $d\,(0_n, bd\,(Z^+)) = \|a\|$. Since $a \in bd\,(Z^+)$, one has $0_n \in bd(Z^+ - a)$.

Consider the problem $\pi_5 := (c - a, \sigma_5)$ where $\sigma_5 := \left\{(a_t^5)'x \geq b_t;\ t \in T\right\}$ with $a_t^5 := a_t - a$ if $t \in T\backslash\{t_0\}$ and $a_{t_0}^5 := a_{t_0} = 0_n$. From the definition of π_5 one has, obviously, $0_n \in cl(Z_5^+)$. If we had $0_n \in int(Z_5^+)$, then we would obtain the following contradiction:

$$0_n \in int\,(conv\,(\{a_t - a\,,\, t \in T\backslash\{t_0\}\ ;\ c - a\})) \subset int\left(Z^+ - a\right).$$

Then $0_n \in bd(Z_5^+)$ and, in particular, $0_{n+1} \notin int(H_5)$ (otherwise $0_n \in int(A_5) \subset int(Z_5^+)$). Thus, Proposition 2 yields $\pi_5 \in cl\,(\Pi_c)$ and we have the following discussion. If $\pi_5 \in bd\,(\Pi_c)$, then Proposition 4 leads us to conclude that $\pi_5 \in bd\,(\Pi_s)$. If $\pi_5 \in int\,(\Pi_c)$, since $0_n \in A_5 \subset cl(Z_5^-)$, Proposition 3 entails $\pi_5 \in cl\,(\Pi_s)$. In any case, we have $\pi_5 \in cl\,(\Pi_s)$, and then

$$\delta\,(\pi, bd\,(\Pi_s)) \leq \delta\,(\pi, \pi_5) = \|a\| = d\left(0_n, bd\left(Z^+\right)\right).$$

Remember that under the current hypotheses we have $-c \in int\,(M)$, i.e., $0_n \in int\,(M + c)$, and the latter implies $\mu\,(M + c) \subset M + c$ for every $0 \leq \mu \leq 1$, due to the convexity of $M + c$. Now we will prove that $Z^+ \subset M + c$. In fact, if $x \in Z^+$ there exist $a \in A$ and $\lambda \geq 0$, $\mu \geq 0$, with $\lambda + \mu = 1$, such that $x = \lambda a + \mu c$. If $\mu > 0$, we can write

$$x = \mu\left(\frac{\lambda}{\mu}a + c\right) \in \mu\,(M + c) \subset M + c,$$

and if $\mu = 0$, we have $x = a \in A \subset M \subset M + c$, where the latter inclusion is due to the fact that $-c \in M$. Thus, we have

$$\begin{aligned} d\left(0_n, bd\left(Z^+\right)\right) &\leq d\,(0_n, bd\,(M + c)) = d\,(0_n, bd\,(M) + c) \\ &= d\,(-c, bd\,(M)) = d\,(c, bd\,(-M)). \quad \square \end{aligned}$$

5. Examples and counterexamples. In this section we present different examples in order to show the usefulness of formula (5) for obtaining the distance to ill-posedness for problems satisfying the hypothesis of Theorem 1. Moreover, some of these examples show that this formula does not generally provide the desired distance when the hypotheses of Theorem 1 are not fulfilled. Different situations illustrate that certain redundant constraints may considerably affect the distance to ill-posedness, even for ordinary linear programming problems with "few" constraints. In fact, we provide two linear optimization problems in $\mathbb{R}^2$ (Examples 5 and 6) with the same associated sets A, M, Z^-, Z^+, C, and H, and different distances to ill-posedness. These examples show that the sets A, M, Z^-, Z^+, C, and H alone are not sufficient to characterize the distance to ill-posedness in all cases. The norm considered (in $\mathbb{R}^n$ and $\mathbb{R}^{n+1}$) in all the examples is $\|\cdot\|_\infty$.

The following examples illustrate formula (5) of Theorem 2 for the problem $\pi \in cl\,(\Pi_s)$.

Example 1. Consider the linear optimization problem in $\mathbb{R}^2$,

$$\begin{array}{lll} \pi: & \text{Inf} & \tfrac{1}{2}x_2 \\ & \text{s.t.} & -x_1 + x_2 \geq 0, \\ & & x_1 + x_2 \geq 0. \end{array}$$

One has

$$H = conv\left\{(-1,1,0)', (1,1,0)'\right\} + \mathbb{R}_{+}\{(0,0,-1)'\}$$

and

$$Z^{-} = conv\left\{(-1,1)', (1,1)', (0,-1/2)'\right\}.$$

Since $0_3 \in ext\,(H)$, then $\pi \in int\,(\Pi_c)$ (Proposition 2(iii)) and then, since $0_2 \in int\,(Z^{-})$, one obtains $\pi \in int\,(\Pi_s)$ (Proposition 3(iii)). It is easy to check that

$$\frac{1}{5} = \left\| \begin{pmatrix} \frac{1}{5} \\ \frac{-1}{5} \end{pmatrix} \right\|_{\infty} = d\left(0_2, bd\left(Z^{-}\right)\right) < d\left(0_3, bd\,(H)\right) = 1.$$

Then, $\delta\,(\pi, bd\,(\Pi_s)) = d\,(0_2, bd\,(Z^{-})) = 1/5$.

As in the proof of Theorem 2, a problem belonging to $bd\,(\Pi_s)$, where this distance is attained, is given by

$$\begin{array}{lll} \pi_0: & \text{Inf} & \left(0+\frac{1}{5}\right)x_1 + \left(\frac{1}{2}-\frac{1}{5}\right)x_2 \\ & \text{s.t.} & \left(-1-\frac{1}{5}\right)x_1 + \left(1+\frac{1}{5}\right)x_2 \;\geq\; 0-0, \\ & & \left(1-\frac{1}{5}\right)x_1 + \left(1+\frac{1}{5}\right)x_2 \;\geq\; 0-0. \end{array}$$

We have $0_2 \in bd(Z_0^{-})$ and $0_3 \in ext\,(H_0)$. Consequently $\pi_0 \in int\,(\Pi_c) \cap bd\,(\Pi_s)$ (Propositions 2(ii) and 3(ii)).

Example 2. Consider now the linear optimization problem in $\mathbb{R}^2$,

$$\begin{array}{llrcl} \pi: & \text{Inf} & 2x_2 & & \\ & \text{s.t.} & -x_1 + x_2 & \geq & 0, \\ & & x_1 + x_2 & \geq & 0, \\ & & \frac{1}{4}x_2 & \geq & 0. \end{array}$$

Now we have

$$H = conv\left\{(-1,1,0)', (1,1,0)', (0,1/4,0)'\right\} + \mathbb{R}_{+}\{(0,0,-1)'\}$$

and

$$Z^{-} = conv\left\{(-1,1)', (1,1)', (0,1/4)', (0,-2)'\right\}.$$

As in the previous example one has $\pi \in int\,(\Pi_s)$. Again it is easy to check that

$$\frac{1}{2} = d\left(0_2, bd\left(Z^{-}\right)\right) > d\left(0_3, bd\,(H)\right) = \left\| \left(0, \frac{1}{4}, 0\right)' \right\|_{\infty} = \frac{1}{4},$$

and $\delta\,(\pi, bd\,(\Pi_s)) = d\,(0_3, bd\,(H)) = 1/4$.

Now, the problem $\pi_0 \in bd\,(\Pi_s)$ where this distance is attained, is given by

$$\begin{array}{lll} \pi_0: & \text{Inf} & 2x_2 \\ & \text{s.t.} & (-1-0)\,x_1 + \left(1-\frac{1}{4}\right)x_2 \;\geq\; 0-0, \\ & & (1-0)\,x_1 + \left(1-\frac{1}{4}\right)x_2 \;\geq\; 0-0, \\ & & (0-0)\,x_1 + \left(\frac{1}{4}-\frac{1}{4}\right)x_2 \;\geq\; 0-0. \end{array}$$

We have $0_3 \in bd\,(H_0)$. In fact $0_3 \in bd\,(C_0)$, and then we conclude that $\pi_0 \in bd\,(\Pi_s)$ (by virtue of Proposition 6(ii)).

Examples 3 and 4 (see also Example 5) show that formula (5) of Theorem 1 cannot be extended to the case $\pi \in ext\,(\Pi_s)$ when $d\,(0_n, bd\,(Z^-)) = d\,(0_{n+1}, bd\,(H)) < \|c\|$, not even in the context of ordinary linear programming in $\mathbb{R}$. The characterization of $bd(\Pi_s)$ given in Proposition 6 is used in the remaining examples.

Example 3. Consider now the ordinary linear programming problem in $\mathbb{R}$,

$$\begin{array}{rll} \pi: & \text{Inf} & -10\,x \\ & \text{s.t.} & x \geq 9, \quad x \geq 10, \quad 4x \geq 9, \quad 4x \geq 10. \end{array}$$

One has that $\pi \in int\,(\Pi_c) \cap ext\,(\Pi_s)$ and $d\,(0, bd\,(Z^-)) = 1 = d\,(0_2, bd\,(H))$. Moreover, $\delta\,(\pi, bd\,(\Pi_s)) = 4 > d\,(0_2, bd\,(H))$, and this is attained in the problem

$$\begin{array}{rll} \pi_1: & \text{Inf} & -10\,x \\ & \text{s.t.} & -3x \geq 9, \quad -3x \geq 10, \quad 0x \geq 9, \quad 0x \geq 10. \end{array}$$

Indeed, if $\pi_2 \in \Pi$ is such that $\delta\,(\pi, \pi_2) < 4$, Proposition 8 guarantees that $0 \in int(Z_2^+)$ and $0_2 \in ext\,(C_2)$. The coefficients of the problem imply that $0_2 \in ext\,(H_2)$ only if $H_2 \subset]0, +\infty[\times \mathbb{R}$, in which case one also has $0_2 \in ext(Z_2^-)$. So it is impossible to have $\pi_2 \in bd\,(\Pi_s)$ according to Proposition 6.

Observe that, in this case, the problems of $bd\,(\Pi_s)$ at which the distance $\delta\,(\pi, bd\,(\Pi_s))$ is attained verify the third condition in Proposition 6.

Example 4. Consider the linear programming problem in $\mathbb{R}$,

$$\begin{array}{rll} \pi: & \text{Inf} & -3x \\ & \text{s.t.} & -x \geq 5, \quad 2x \geq 5, \quad -x \geq 4, \quad 2x \geq 4. \end{array}$$

One has that $\pi \in int\,(\Pi_i)$ and $d\,(0_2, bd\,(H)) = d\,(0, bd\,(Z^-)) = 1$. It is easy to check that

$$\delta\,(\pi, bd\,(\Pi_s)) = \delta\,(\pi, \pi_1) = 2 > d\,(0_2, bd\,(H)),$$

where

$$\begin{array}{rll} \pi_1: & \text{Inf} & -3x \\ & \text{s.t.} & -3x \geq 5, \quad 0x \geq 5, \quad -3x \geq 4, \quad 0x \geq 4. \end{array}$$

In the following examples, the considered problems have the same sets A, M, Z^-, Z^+, C, and H, but their distances to ill-posedness are different. Moreover, in the second example the distance to ill-posedness obeys formula (5), although the problem does not satisfy the hypotheses of Theorem 1.

Example 5. Consider the problem

$$\begin{array}{rlrcll} \pi: & \text{Inf} & 6\,x_1 - 3x_2 & & & \\ & \text{s.t.} & 3x_1 + x_2 & \geq & 10, & t = 1, \\ & & 3x_1 + 3x_2 & \geq & 10, & t = 2, \\ & & -2x_1 + 3x_2 & \geq & 10, & t = 3, \\ & & -2x_1 + x_2 & \geq & 10, & t = 4. \end{array}$$

It is easy to check that $\pi \in int\,(\Pi_c) \cap ext\,(\Pi_s)$ and $d\,(0_2, bd\,(Z^-)) = 1 = d\,(0_3, bd\,(H))$. Now let us see that $\delta\,(\pi, bd\,(\Pi_s)) > 1$. The choice of $b_t = 10$, $t = 1, \ldots, 4$, allows us

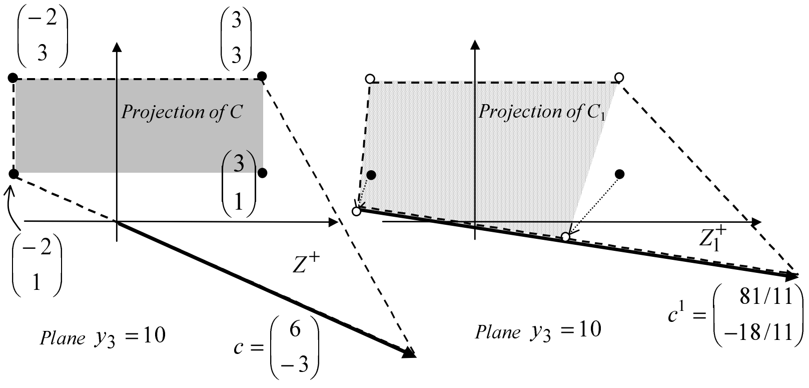

$\begin{pmatrix}-2\\3\end{pmatrix}$
$\begin{pmatrix}3\\3\end{pmatrix}$
Projection of C
$\begin{pmatrix}3\\1\end{pmatrix}$
$\begin{pmatrix}-2\\1\end{pmatrix}$
Z^+
Plane $y_3 = 10$
$c = \begin{pmatrix}6\\-3\end{pmatrix}$
Projection of C_1
Z_1^+
Plane $y_3 = 10$
$c^1 = \begin{pmatrix}81/11\\-18/11\end{pmatrix}$

FIG. 3. *Perturbation of π to obtain condition* (iii) *of Proposition* 6.

to see that the nearest problem to π, $\pi_1 \in bd(\Pi_s)$ does not verify the condition $0_3 \in bd(C_1)$. Thus, we will look for a problem $\pi_1 = (c^1, \sigma_1)$, with $\sigma_1 = \{(a_t^1)'x \geq b_t^1,\ t = 1, \dots, 4\}$, as the one presented in Figure 3 below, where we illustrate graphically the sets C and Z^+, associated with π, as well as the ones associated with π_1. It can be proved that the minimum perturbation (in $\|\cdot\|_\infty$) for which c^1 is a multiple of a_1^1 corresponds to the vectors $c^1 = \binom{81/11}{-18/11}$ and $a_1^1 = \binom{18/11}{-4/11}$, both colinear with $\binom{9}{-2}$. Now we modify a_4 to get $0_3 \in bd(H_1)$ and $0_2 \in bd(Z_1^+)$, taking for example $a_4^1 = \binom{-27/11}{6/11}$. The remaining coefficients stay unchanged. For simplicity, and since $b_t^1 = b_t = 10$, for $t = 1, \dots, 4$, we represent the figures projected in the plane $\{y \in \mathbb{R}^3 \mid y_3 = 10\}$.

Next we will check that if $\pi_2 = (c^2, \sigma_2) \in \Pi$ verifies $\delta(\pi_2, \pi) < \delta(\pi_1, \pi) = \frac{15}{11}$, then $\pi_2 \notin \Pi_s$. Indeed, in other case and appealing to Proposition 1(ii), we will obtain $c^2 \in M_2$ (which is a closed cone because it is finitely generated). Moreover, one can easily check that the following inequalities relative to $\sigma_2 = \{(a_t^2)'x \geq b_t^2, t = 1, \dots, 4\}$ hold:

$$(a_1^2)'\tbinom{9}{-2} > 0,\ (a_2^2)'\tbinom{9}{-2} > 0,\ (a_3^2)'\tbinom{9}{-2} < 0,\ (a_4^2)'\tbinom{9}{-2} < 0, \tag{6}$$

$$(a_1^2)'\tbinom{2}{9} > 0, \quad (a_2^2)'\tbinom{2}{9} > 0, \quad (a_3^2)'\tbinom{2}{9} > 0, \tag{7}$$

$$(c^2)'\tbinom{9}{-2} > 0, \tag{8}$$

$$(c^2)'\tbinom{2}{9} < 0. \tag{9}$$

The condition $c^2 \in M_2$, together with (7) and (9), implies that $(a_4^2)'\binom{2}{9} < 0$; then the vectors a_1^2, a_3^2, a_4^2, and c^2 are, respectively, in the interior of the first second, third, and fourth quadrants determined by the (orthogonal) vectors $\binom{9}{-2}$ and $\binom{2}{9}$, from where one deduces that $M_2 = \mathbb{R}^2$. That is, $0_2 \in int(conv(\{a_t^2,\ t = 1, \dots, 4\}))$, and since $b_t^2 \geq 10 - \frac{15}{11} > 0$ for all $t = 1, \dots, 4$, we conclude that $0_3 \in int(H_2)$, and then $\pi_2 \in int(\Pi_i)$ (Proposition 2(i)), in contradiction with the assumption $\pi_2 \in \Pi_s$. Thus we conclude that $\delta(\pi, bd(\Pi_s)) = \delta(\pi, \pi_1) = \frac{15}{11}$.

Example 6. Consider the linear programming problem π in $\mathbb{R}^2$, obtained by adding to the problem of the previous example the constraint $0x_1 + x_2 \geq 10$ (for $t = 5$), whose coefficient vector is a convex combination of the vectors associated with

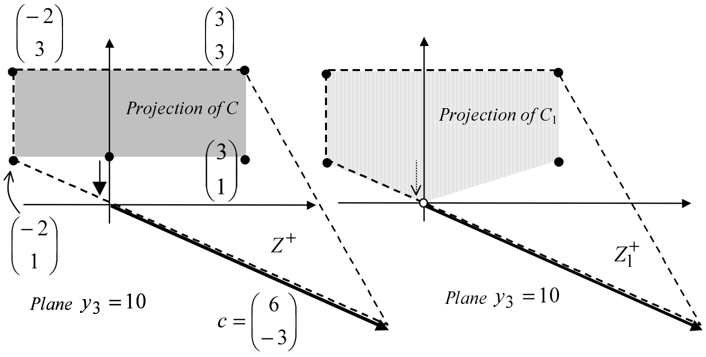

$\begin{pmatrix}-2\\3\end{pmatrix}$
$\begin{pmatrix}3\\3\end{pmatrix}$
Projection of C
$\begin{pmatrix}3\\1\end{pmatrix}$
$\begin{pmatrix}-2\\1\end{pmatrix}$
Z^+
Plane $y_3=10$
$c=\begin{pmatrix}6\\-3\end{pmatrix}$
Projection of C_1
Z_1^+
Plane $y_3=10$

FIG. 4. *Perturbation of* π *to obtain condition* (iii) *of Proposition* 6.

the first and fourth constraints (in particular, the new constraint is redundant). This problem still verifies the conditions $\pi \in int(\Pi_c) \cap ext(\Pi_s)$ and $d(0_2, bd(Z^-)) = 1 = d(0_3, bd(H))$. However, now one has $\delta(\pi, bd(\Pi_s)) = d(0_3, bd(H))$, since $d(0_3, bd(H))$ is a lower bound for the distance from π to $bd(\Pi_s)$ (see Theorem 3) and the problem π_1, coming from replacing the last constraint of π by the new one "$0x_1 + 0x_2 \geq 10$," verifies that $\pi_1 \in bd(\Pi_s)$ (since $0_{n+1} \in bd(H_1)$ and $0_n \in bd(Z_1^+)$) and $\delta(\pi, \pi_1) = 1$. In Figure 4 we illustrate graphically these elements, projecting them on the plane $y_3 = 10$ (note that $b_t = 10$, for all $t = 1, \ldots, 4$).

6. A primal-dual approach to the distance to solvability/unsolvability.

Let us consider the dual of problem (1), which is given by

$$(10)\qquad \begin{aligned} \pi^d:\quad & \text{Sup} \sum_{t\in T} \lambda_t b_t \\ & \text{s.t.} \sum_{t\in T} \lambda_t a_t = c, \\ & \lambda \in \mathbb{R}_+^{(T)}, \end{aligned}$$

where $\mathbb{R}_+^{(T)}$ is the convex cone of all the functions $\lambda : T \to \mathbb{R}_+$ taking positive values only at finitely many points of T. When T is infinite, π^d is also a linear semi-infinite programming problem having, in this case a finite number of constraints but an infinite number of variables. In this case, π^d is called the dual of π in the sense of Haar. The subset of Π formed by those problems whose dual is consistent will be denoted by Π_c^d; in other words, $\Pi_c^d := \{\pi \in \Pi \mid c \in M\}$.

In the finite case (T finite) it is well known from linear programming duality that the problem π is solvable if and only if it is both primal and dual feasible; i.e., $\Pi_s = \Pi_c \cap \Pi_c^d$. Hence, for solvable instances the distance to unsolvability is given by

$$(11)\qquad \delta(\pi, bd(\Pi_s)) = \min\left\{\delta(\pi, bd(\Pi_c)), \delta\left(\pi, bd\left(\Pi_c^d\right)\right)\right\}.$$

The finite case is actually a particular case of the conic linear context studied in [5], [10], [16], [18], [19], and [20], among others. In section 7, which is specifically devoted to the finite case, we show how some results of [10], [19], and [20] may be used to derive, for $\pi \in \Pi_s$, the expressions

$$\delta(\pi, bd(\Pi_c)) = d(0_{n+1}, bd(H)) \text{ and } \delta\left(\pi, bd\left(\Pi_c^d\right)\right) = d\left(0_n, bd\left(Z^-\right)\right).$$

The first equality is a particular case of Proposition 2(iv), and the second is extended, in Theorem 6, to the semi-infinite case and for problems without any consistency requirements.

First of all, let us point out that when T is infinite, solvability may not be identified with primal-dual feasibility. In fact, none of the inclusions $\Pi_s \supset \Pi_c \cap \Pi_c^d$ and $\Pi_s \subset \Pi_c \cap \Pi_c^d$ holds, as the following examples show. Despite this fact, Corollary 1 will show that (11) remains valid for problems in $cl\,(\Pi_s)$ as well as for some subset of insolvable problems.

Example 7. $\Pi_s \not\supset \Pi_c \cap \Pi_c^d$. Let us consider the linear semi-infinite programming problem in $\mathbb{R}^2$,

$$\begin{aligned}\pi: \quad & \operatorname{Inf} x_1 \\ & \text{s.t.} \quad t\,x_1 + \frac{1}{t}x_2 \geq 2,\ t \in \left]0, +\infty\right[\ , \\ & \qquad\ \ x_1 \geq 0.\end{aligned}$$

One can easily check that $F = \{(x_1, x_2)' \mid x_2 \geq x_1^{-1}, x_1 > 0\}$ and then π is bounded but not solvable. However, $\pi \in \Pi_c \cap \Pi_c^d$.

Example 8. $\Pi_s \not\subset \Pi_c \cap \Pi_c^d$. Let us consider the problem in $\mathbb{R}^2$,

$$\begin{aligned}\pi: \quad & \operatorname{Inf} x_1 \\ & \text{s.t.} \quad x_1 + t\,x_2 \geq 0,\ t \in \left]0, +\infty\right[\ .\end{aligned}$$

The feasible set of π coincides with $[0, +\infty[^2$ and then it is immediate that $\pi \in \Pi_s$. However, $M = \left]0, +\infty\right[^2 \cup \{0_2\}$ and then $c = (1, 0)' \notin M$, i.e., $\pi \notin \Pi_c^d$.

The following theorem is a dual version of Proposition 2. It characterizes the ill-posedness with respect to the dual consistency and provides the associated distance to ill-posedness.

THEOREM 6. *Let $\pi \in \Pi$. The following statements hold:*

(i) $\pi \in int\left(\Pi_c^d\right)$ *if and only if* $0_n \in int\,(Z^-)$;

(ii) $\pi \in bd\left(\Pi_c^d\right)$ *if and only if* $0_n \in bd\,(Z^-)$;

(iii) $\pi \in ext\left(\Pi_c^d\right)$ *if and only if* $0_n \in ext\,(Z^-)$;

(iv) $\delta\left(\pi, bd\left(\Pi_c^d\right)\right) = d\,(0_n, bd\,(Z^-))$.

Proof. (i) Theorem 5 in [14] establishes that $\pi \in int(\Pi_c^d)$ if and only if $c \in int\,(M)$, which is equivalent to $0_n \in int\,(Z^-)$ (see Proposition 7).

(ii) Let us start with the "only if" part. Take $\pi \in bd(\Pi_c^d)$ and a sequence $\{\pi_r\} \subset \Pi_c^d$ converging to π. We have, on the one hand, $c^r \in M_r$ for each $r \in \mathbb{N}$ and consequently $0_n \in Z_r^-$. On the other hand, $0_n \notin int\,(Z^-)$ because $\pi \in bd\left(\Pi_c^d\right)$. Assume by contradiction that $0_n \notin bd\,(Z^-)$. Then $0_n \in ext\,(Z^-)$ and, since $\{\pi_r\}$ converges to π, Proposition 8 ensures that $0_n \in ext\,(Z_r^-)$ for r large enough, which represents a contradiction. Therefore, $0_n \in bd\,(Z^-)$.

In order to prove the "if" condition, assume that $0_n \in bd\,(Z^-)$ and take a sequence $\{u^r\} \subset Z^-$ converging to 0_n. We can write

$$u^r = \sum_{t \in T} \lambda_t^r a_t - \mu_r c, \qquad r = 1, 2, \ldots,$$

for some sequences $\{\lambda^r\} \subset \mathbb{R}_+^{(T)}$ and $\{\mu_r\} \subset \mathbb{R}_+$ verifying $\sum_{t \in T} \lambda_t^r + \mu_r = 1$. For each $r \in \mathbb{N}$, define the problem $\pi_r := (c^r, \sigma_r)$ by distinguishing two cases:

(1) If $\mu_r = 0$, let $c^r := c$ and

$$\sigma_r := \left\{ \left(a_t - u^r + \frac{1}{r}c \right)' x \geq b_t \, , \; t \in T \right\}.$$

Note that in this case $\sum_{t\in T} \lambda_t^r = 1$ and $\sum_{t\in T} \lambda_t^r (a_t - u^r + \frac{1}{r}c) = \frac{1}{r}c$, so $c^r = c \in M_r$.

(2) If $\mu_r > 0$, defining $c^r := c + u^r$ and

$$\sigma_r := \left\{ (a_t - u^r)' x \geq b_t \, , \; t \in T \right\},$$

one has $\sum_{t\in T} \lambda_t^r \, (a_t - u^r) = \mu_r \, (c + u^r)$ and, again, $c^r \in M_r$.

Thus, in any case, $\pi_r \in \Pi_c^d$ and, since $\{\pi_r\}$ converges to π, then $\pi \in cl(\Pi_c^d)$. Finally, under the current hypothesis, $\pi \notin int(\Pi_c^d)$ and then $\pi \in bd(\Pi_c^d)$.

(iii) This is a straightforward consequence of (i) and (ii).

(iv) If $\pi \in bd(\Pi_c^d)$, the desired equality comes trivially from (ii). Assume that $\pi \in int(\Pi_c^d)$, and let us see first the inequality "$\geq$." If $\pi_1 \in \Pi$ satisfies $\delta(\pi, \pi_1) < d(0_n, bd(Z^-))$, then Proposition 8 implies $0_n \in int(Z_1^-)$ and thus $\pi_1 \in int(\Pi_c^d)$. In order to establish the inequality "$\leq$," take $u \in bd\,(Z^-)$ such that $d\,(0_n, bd\,(Z^-)) = \|u\|$ and define $\pi_1 := (c + u, \sigma_{-u})$, where $\sigma_{-u} := \left\{ (a_t - u)' \, x \geq b_t \, , \; t \in T \right\}$. In such a way $Z_1^- = Z^- - u$, and then $0_n \in bd(Z_1^-)$, which entails $\pi_1 \in bd\left(\Pi_c^d\right)$. Thus,

$$\delta\left(\pi, bd\left(\Pi_c^d\right)\right) \leq \delta\left(\pi, \pi_1\right) = \|u\| = d\left(0_n, bd\left(Z^-\right)\right).$$

In the case $\pi \in ext\left(\Pi_c^d\right)$ one obtains the desired equality just by replacing *int* by *ext* in the previous argument. □

The following corollary comes directly from Theorem 1, Proposition 2, and statement (iv) of Theorem 6.

COROLLARY 1. *Let* $\pi = (c, \sigma) \in \Pi \backslash \Pi_\infty$. *Suppose that at least one of the following conditions holds:*

(i) $\pi \in cl\,(\Pi_s)$;

(ii) $\pi \in ext\,(\Pi_s)$ *and* $d\,(0_{n+1}, bd\,(H)) \neq d\,(0_n, bd\,(Z^-))$;

(iii) $d\,(0_{n+1}, bd\,(H)) = d\,(0_n, bd\,(Z^-)) \geq \|c\|$.

Then one has

$$\delta\left(\pi, bd\left(\Pi_s\right)\right) = \min\left\{ \delta\left(\pi, bd\left(\Pi_c\right)\right), \delta\left(\pi, bd\left(\Pi_c^d\right)\right) \right\}. \tag{12}$$

Bounds on $\delta\left(\pi, bd\left(\Pi_s\right)\right)$ in terms of $\delta\left(\pi, bd\left(\Pi_c\right)\right)$ and $\delta\left(\pi, bd\left(\Pi_c^d\right)\right)$ may be obtained by reformulating Theorems 3 and 4. The results in these theorems also provide the inequalities $\delta\left(\pi, bd\left(\Pi_c^d\right)\right) \leq \delta\left(\pi, bd\left(\Pi_c\right)\right)$, for $\pi \in ext\left(\Pi_s\right) \cap int\left(\Pi_c\right)$, and $\delta(\pi, bd(\Pi_c)) \leq \delta(\pi, bd(\Pi_c^d))$, for $\pi \in \Pi_i \backslash \Pi_\infty$. See [5, section 2.4] for a counterpart in the finite case.

The following proposition clarifies the relationship between Π_s and $\Pi_c \cap \Pi_c^d$ in our context. It shows that although both sets do not coincide, the ill-posedness with respect to the solvability may be identified to the ill-posedness with respect to the primal-dual consistency. We make use of the inclusion

$$\Pi_c \cap \Pi_c^d \subset \Pi_b,$$

which comes from the following standard argument on duality.

If $\pi \in \Pi_c \cap \Pi_c^d$ and we take any feasible point $\lambda \in \mathbb{R}_+^{(T)}$ of the dual problem, i.e., $\sum_{t \in T} \lambda_t a_t = c$, we have

$$c'x = \sum_{t \in T} \lambda_t a_t' x \geq \sum_{t \in T} \lambda_t b_t \text{ for all } x \in F. \tag{13}$$

Proposition 11. *The following statements hold:*

(i) $int\,(\Pi_s) = int\,(\Pi_c \cap \Pi_c^d)$*;*

(ii) $bd\,(\Pi_s) = bd\,(\Pi_c \cap \Pi_c^d)$*;*

(iii) $ext\,(\Pi_s) = ext\,(\Pi_c \cap \Pi_c^d)$*.*

Proof. We shall prove conditions (i) and (ii), and then (iii) follows. Condition (i) is a consequence of Proposition 3 and Theorem 6, taking into account that $int\,(\Pi_c) \cap int\,(\Pi_c^d) = int\,(\Pi_c \cap \Pi_c^d)$.

(ii) If $\pi \in bd\,(\Pi_c \cap \Pi_c^d) \subset cl\,(\Pi_b) = cl\,(\Pi_s)$, where the last equality comes from [4, Thm. 1], then it must be $\pi \in bd\,(\Pi_s)$, taking into account the previous condition. Assume now that $\pi \in bd\,(\Pi_s)$ and take $\pi_r = (c^r, \sigma_r) \in \Pi_s$ with $\{\pi_r\}$ converging to π. By Proposition 1(ii), $c^r \in cl\,(M_r)$ for each r. Then, for each r, there exists $\widetilde{c}^r \in M_r$ with $\|\widetilde{c}^r - c^r\| \leq \frac{1}{r}$. Therefore the problem $\widetilde{\pi}_r = (\widetilde{c}^r, \sigma_r) \in \Pi_c \cap \Pi_c^d$ for all r. Since $\{\widetilde{\pi}_r\}$ converges to π, we have $\pi \in cl(\Pi_c \cap \Pi_c^d)$ and then, under the current hypothesis, $\pi \in bd(\Pi_c \cap \Pi_c^d)$. □

Remark 4. In general $cl(\Pi_c \cap \Pi_c^d) \subset cl(\Pi_c) \cap cl(\Pi_c^d)$, but the opposite inclusion does not hold, even in the finite case. Just consider the problem in $\mathbb{R}$ given by $\pi := \text{Inf}\,\{-x \mid 0x \geq 1, x \geq 1\}$. Since $d\,(0_2, bd\,(H)) = 0 = d\,(0, bd\,(Z^-))$, $\pi \in \Pi_i \cap cl\,(\Pi_c) \cap cl\,(\Pi_c^d)$. However, $\pi \notin bd\,(\Pi_s)$ (see Proposition 6), and then $\pi \notin cl\,(\Pi_s)$.

7. The finite case. In the finite case (T finite), the distances to inconsistency (primal and/or dual) are studied in [5], [10], [16], [18], [19], [20], etc. In particular [10], following the steps of [20], deals with a conic linear system $\sigma = (A, b)$,

$$\sigma : \quad \begin{array}{l} b - Ax \in C_Y, \\ x \in C_X, \end{array} \tag{14}$$

where $C_X \subset X$ and $C_Y \subset Y$ are closed convex cones in X and Y, respectively. X and Y are an n-dimensional and an m-dimensional normed space, respectively, and the norms in both spaces are represented by $\|\cdot\|$. Here $b \in Y$ and $A : X \longrightarrow Y$ is a linear operator, with norm $\|A\| := \sup\{\|Ax\| \mid \|x\| \leq 1\}$. The parameter space of all systems (14) is endowed with the product norm

$$\|\sigma\| = \|(A, b)\| := \max\{\|A\|, \|b\|\}. \tag{15}$$

This model includes our primal constraint system if T is finite (particularly if $|T| = m$) just by taking $C_X := \mathbb{R}^n$ and $C_Y := -\mathbb{R}_+^m$. As a consequence of this fact, finite-dimensional versions of our distances to inconsistency are obtained by applying Theorems 1 and 2 in [10]. Unfortunately the tools developed in [10] and [20] do not apply in our context, in which $Y = \mathbb{R}^T$ is not a normed space when T is infinite and arbitrary.

With respect to the aim of relating our results to [10, Thms. 1 and 2] and [20, Thm. 3.5], and if we identify $a \in \mathbb{R}^n$ with the linear operator $x \mapsto a'x$, a suitable norm to be used in $\mathbb{R}^{n+1}$ would be

$$\left\| \begin{pmatrix} a \\ b \end{pmatrix} \right\| = \max\{\|a\|_*, |b|\}. \tag{16}$$

Observe that this norm satisfies conditions (3) and (4). Moreover, the norm $\|\cdot\|$ in $X = \mathbb{R}^n$ will be arbitrary, whereas in $Y = \mathbb{R}^m$ we shall use the norm $\|\cdot\|_\infty$.

Specifically, when T is finite, our system $\{a_t' x \geq b_t\ ,\ t \in T\}$ can be rewritten in the matrix form $Ax \geq b$ (where the tth row of the matrix A is a_t', and the tth component of the vector b is b_t). Then, and thanks to the fact that in Y we chose the infinite-norm $\|\cdot\|_\infty$,

$$\|\sigma\| = \max\left\{\max_{\|x\|\leq 1}\|Ax\|, \|b\|\right\} = \max_{t\in T}\max\left\{\|a_t\|_*, |b_t|\right\} = \max_{t\in T}\left\|\begin{pmatrix} a_t \\ b_t\end{pmatrix}\right\|.$$

For system σ in (14), [10] presents different mathematical programs, each of whose optimal values provides either the exact distance to inconsistency, denoted by $\rho(\sigma)$, or an approximation of $\rho(\sigma)$ to within certain constants. In particular, if $X = \mathbb{R}^n$ and $Y = \mathbb{R}^m$, Theorem 2 in [10] establishes, when σ is consistent, that $\rho(\sigma)$ coincides with the optimal value of the program,

$$\begin{array}{ll} \mathrm{Inf}_{y,q,g} & \max\left\{\|A'y - q\|_*, |b'y + g|\right\} \\ \text{s.t.} & y \in C_Y^*,\ \|y\|_* = 1,\ q \in C_X^*,\ g \geq 0, \end{array} \tag{17}$$

where A' and b' are the transposes of A and b, respectively.

In our framework $C_X := \mathbb{R}^n$, $C_Y := -\mathbb{R}_+^m$ and, writing $\lambda := -y$, program (17) is equivalent to

$$\begin{array}{ll} \mathrm{Inf}_{\lambda,g} & \max\left\{\left\|\sum_{t\in T}\lambda_t a_t\right\|_*, \left|\sum_{t\in T}\lambda_t b_t - g\right|\right\} \\ \text{s.t.} & \lambda \geq 0_m,\ \|\lambda\|_1 = 1,\ g \geq 0. \end{array}$$

By defining $w_{n+1} := \sum_{t\in T}\lambda_t b_t - g$, and according to (16), we get another equivalent program:

$$\begin{array}{ll} \mathrm{Inf}_{\lambda} & \left\|\begin{pmatrix} w \\ w_{n+1}\end{pmatrix}\right\| \\ \text{s.t.} & \lambda \geq 0_m,\ \|\lambda\|_1 = 1, \\ & w = \sum_{t\in T}\lambda_t a_t,\ w_{n+1} \leq \sum_{t\in T}\lambda_t b_t. \end{array}$$

In this way we conclude that, for σ consistent, the distance to the primal inconsistency $\rho(\sigma)$ coincides with $d(0_{n+1}, H)$, where H is the hypographical set introduced in section 2. Hence, we recover a partial result in Proposition 2(iv), limited to (primal) consistent problems in the finite case.

On the other hand, the dual problem (10) may be rewritten in the finite case as

$$\begin{array}{ll} \pi^d: & \text{Sup } b'\lambda \\ & \text{s.t. } A'\lambda = c, \\ & \qquad \lambda \in \mathbb{R}_+^m. \end{array}$$

Assuming that this dual problem is consistent (i.e., $\pi \in \Pi_c^d$), the distance to dual inconsistency $\delta(\pi, bd(\Pi_c^d))$ is, according to Theorem 3.5 in [20],

$$\begin{aligned} &\inf\left\{\|q\| \mid \{q = A'\lambda - sc,\ \lambda \geq 0_m,\ s \geq 0,\ \|\lambda\|_1 + |s| \leq 1\} \text{ is inconsistent}\right\} \\ &= \inf\left\{\|q\| \mid q \notin conv\left(\{a_t,\ t \in T;\ -c,\ 0_n\}\right)\right\} = d\left(0_n, bd\left(Z^-\right)\right), \end{aligned}$$

where we have taken into account the fact that $0_n \in Z^-$ (due to the consistency of π^d). Hence, we also recover a partial result in Theorem 6(iv), limited to dual-consistent problems in the finite case.

Finally, let us recall that the specification of inequality (i) in Theorems 3 and 4 for the finite case can be traced from section 2.4 in [5]. Concerning the distance from unsolvable instances to solvability, statements (ii) and (iii) in Theorem 1 are new even in the finite case.

REFERENCES

[1] M. J. Cánovas, A. L. Dontchev, M. A. López, and J. Parra, *Metric regularity of semi-infinite constraint systems*, Math. Program., Ser. B, to appear.

[2] M. J. Cánovas, M. A. López, J. Parra, and M. I. Todorov, *Stability and well-posedness in linear semi-infinite programming*, SIAM J. Optim., 10 (1999), pp. 82–98.

[3] M. J. Cánovas, M. A. López, J. Parra, and F. J. Toledo, *Distance to ill-posedness and the consistency value of linear semi-infinite inequality systems*, Math. Program., 103A (2005), pp. 95–126.

[4] M. J. Cánovas, M. A. López, J. Parra, and F. J. Toledo, *Ill-Posedness with Respect to the Solvability in Linear Optimization*, preprint, 2004.

[5] D. Cheung, F. Cucker, and J. Peña, *Unifying condition numbers for linear programming*, Math. Oper. Res., 28 (2003), pp. 609-624.

[6] A. L. Dontchev, A. S. Lewis, and R. T. Rockafellar, *The radius of metric regularity*, Trans. Amer. Math. Soc., 355 (2002), pp. 493–517.

[7] A. L. Dontchev and R. T. Rockafellar, *Regularity and conditioning of solution mappings in variational analysis*, Set-Valued Anal., 12 (2004), pp. 79–109.

[8] M. Epelman and R. M. Freund, *Condition number complexity of an elementary algorithm for computing a reliable solution of a conic linear system*, Math. Program., 88A (2000), pp. 451–485.

[9] M. Epelman and R. M. Freund, *A new condition measure, preconditioners, and relations between different measures of conditioning for conic linear systems*, SIAM J. Optim., 12 (2002), pp. 627–655.

[10] R. M. Freund and J. R. Vera, *Some characterizations and properties of the "distance to ill-posedness" and the condition measure of a conic linear system*, Math. Program., 86A (1999), pp. 225–260.

[11] R. M. Freund and J. R. Vera, *Condition-based complexity of convex optimization in conic linear form via the ellipsoid algorithm*, SIAM J. Optim., 10 (1999), pp. 155–176.

[12] M. A. Goberna and M. A. López, *Linear Semi-infinite Optimization*, John Wiley, London, 1998.

[13] M. A. Goberna, M. A. Lopez, and M. I. Todorov, *Stability theory for linear inequality systems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 730–743.

[14] M. A. Goberna, M. A. López, and M. I. Todorov, *On the stability of the feasible set in linear optimization*, Set-Valued Anal., 9 (2001), pp. 75–99.

[15] D. Klatte and B. Kummer, *Nonsmooth Equations in Optimization: Regularity, Calculus, Methods and Applications*, Kluwer Academic Publishers, Dordrecht, NL, 2002.

[16] M. A. Nunez, *A characterization of ill-posed data instances for convex programming*, Math. Program., 91 (2002), pp. 375–390.

[17] M. A. Nunez and R. M. Freund, *Condition measures and properties of the central trajectory of a linear program*, Math. Program., 83 (1998), pp. 1–28.

[18] J. Peña, *Understanding the geometry of infeasible perturbations of a conic linear system*, SIAM J. Optim., 10 (2000), pp. 534–550.

[19] J. Renegar, *Some perturbation theory for linear programming*, Math. Program., 65A (1994), pp. 73–91.

[20] J. Renegar, *Linear programming, complexity theory and elementary functional analysis*, Math. Program., 70 (1995), pp. 279–351.

[21] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

SIAM J. OPTIM.
Vol. 16, No. 3, pp. 650–669

PRECISION CONTROL FOR GENERALIZED PATTERN SEARCH ALGORITHMS WITH ADAPTIVE PRECISION FUNCTION EVALUATIONS*

ELIJAH POLAK† AND MICHAEL WETTER‡

Abstract. In the literature on generalized pattern search algorithms, convergence to a stationary point of a once continuously differentiable cost function is established under the assumption that the cost function can be evaluated exactly. However, there is a large class of engineering problems where the numerical evaluation of the cost function involves the solution of systems of differential algebraic equations. Since the termination criteria of the numerical solvers often depend on the design parameters, computer code for solving these systems usually defines a numerical approximation to the cost function that is discontinuous with respect to the design parameters. Standard generalized pattern search algorithms have been applied heuristically to such problems, but no convergence properties have been stated.

In this paper we extend a class of generalized pattern search algorithms to include a subprocedure that adaptively controls the precision of the approximating cost functions. The numerical approximations to the cost function need not define a continuous function. Our algorithms can be used for solving linearly constrained problems with cost functions that are at least locally Lipschitz continuous.

Assuming that the cost function is smooth, we prove that our algorithms converge to a stationary point. Under the weaker assumption that the cost function is only locally Lipschitz continuous, we show that our algorithms converge to points at which the Clarke generalized directional derivatives are nonnegative in predefined directions.

An important feature of our adaptive precision scheme is the use of coarse approximations in the early iterations, with the approximation precision controlled by a test. We show by numerical experiments that such an approach leads to substantial time savings in minimizing computationally expensive functions.

Key words. algorithm implementation, approximations, generalized pattern search, Hooke–Jeeves, Clarke's generalized directional derivative, nonsmooth optimization

AMS subject classifications. 90C30, 90C56, 90C59, 65K05

DOI. 10.1137/040605527

1. Introduction.

Generalized pattern search (GPS) algorithms are derivative free methods for the minimization of smooth functions, possibly with linear inequality constraints. Examples of pattern search algorithms are the coordinate search algorithm [22], the pattern search algorithm of Hooke and Jeeves [14], and the multidirectional search algorithm of Dennis and Torczon [9]. What they all have in common is that they define the construction of a mesh, which is then explored according to some rule, and if no decrease in cost is obtained on mesh points around the current iterate, then the mesh is refined and the process is repeated.

*Received by the editors March 23, 2004; accepted for publication (in revised form) May 20, 2005; published electronically January 6, 2006.

http://www.siam.org/journals/siopt/16-3/60552.html

†Department of Electrical Engineering, University of California at Berkeley, Berkeley, CA 94720 (polak@eecs.berkeley.edu). This author's research was supported by the National Science Foundation under grant ECS-9900985.

‡United Technologies Research Center, East Hartford, CT 06108 (WetterM@utrc.utc.com). This author's research was performed at the Simulation Research Group, Building Technologies Department, Environmental Energy Technologies Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720. This author's research was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of the Building Technologies Program of the U.S. Department of Energy, under contract DE-AC03-76SF00098.

In 1997, Torczon [26] was the first to show that all the existing pattern search algorithms are specific implementations of an abstract pattern search scheme and to establish that for unconstrained problems with smooth cost functions $f\colon \mathbb{R}^n \to \mathbb{R}$, the gradient of the cost function vanishes at accumulation points of sequences constructed by this scheme. Lewis and Torczon extended her theory to address bound constrained problems [20] and problems with linear inequality constraints [21]. In both cases, convergence to a feasible point x^* satisfying $\langle \nabla f(x^*), x - x^* \rangle \geq 0$ for all feasible x is proven under the condition that $f(\cdot)$ is once continuously differentiable. Audet and Dennis [3] present a simpler abstraction of GPS algorithms, and, in addition to reestablishing the Torczon and the Lewis and Torczon results, they relax the assumption that the cost function is smooth to that it is locally Lipschitz continuous.

In principle, a natural area for the application of GPS algorithms is engineering optimization, where the cost functions are defined on the solution of complex systems of equations including implicit equations, ordinary differential equations, and partial differential equations. However, in such cases, obtaining an accurate approximation to the cost function often takes many hours, and there is no straightforward way of approximating gradients. Furthermore, it is not uncommon that the termination criteria of the numerical solvers introduce discontinuities in the approximations to the cost function. Hence, standard GPS algorithms can only be used heuristically in this context.

In this paper we present a modified class of GPS algorithms which adjust the precision of the cost function evaluations adaptively: low precision in the early iterations, with precision progressively increasing as a solution is approached. The modified GPS algorithms converge to stationary points of the cost function even though the cost function is approximated by a family of discontinuous functions.

We assume that the cost function $f(\cdot)$ is at least locally Lipschitz continuous and that it can be approximated by a family of functions, say $\{f^*(\epsilon, \cdot)\}_{\epsilon \in \mathbb{R}^q_+}$, with fixed $q \in \mathbb{N}$, where $\epsilon \in \mathbb{R}^q_+$ denotes the tolerance settings of the PDE, ODE, and algebraic equation solvers, and each $f^*(\epsilon, \cdot)$ may be discontinuous but converges to $f(\cdot)$ uniformly on compact sets. A test in the algorithm determines when precision must be increased. This test includes parameters that can be used to control the speed with which precision is increased. We will show by numerical experiments that this flexibility can be exploited to obtain a significant reduction in computation times, as compared to using high precision throughout the computation.

Under the assumption that the cost function is continuously differentiable, all the accumulation points constructed by our GPS algorithms are stationary, while under the assumption that $f(\cdot)$ is only locally Lipschitz continuous, our algorithms converge to points at which the Clarke generalized directional derivatives are nonnegative in predefined directions. Thus, we regain the results of [3].

To prove that our GPS algorithms construct accumulation points at which Clarke's generalized directional derivatives are nonnegative in predefined directions, we made the assumption that $f(\cdot)$ is Lipschitz continuous on $\mathbb{R}^n$ rather than only near the accumulation points, even though local Lipschitz continuity suffices for proving Theorem 5.5. The reason is that in setting up engineering optimization problems, it is easier to construct $f(\cdot)$ to be Lipschitz continuous on $\mathbb{R}^n$, rather than only near the accumulation points, because they are unknown at the time the optimization problem is set up. Furthermore, as shown in [3], strict differentiability, as defined in [8], near an accumulation point x^* suffices to prove that $\nabla f(x^*) = 0$. But since the implicit function theorem and standard theory of differential equations [23] state smoothness of solutions in terms of continuous differentiability rather than strict differentiability, we assume in proving Theorem 5.6 that $f(\cdot)$ is continuously differentiable.

Contrary to the model management framework with generalized pattern search algorithms [10, 27, 25, 4], we do not assume that function values of $f(\cdot)$ are available, and consequently we do not construct surrogate models of increasing accuracy that are based on support points at which $f(\cdot)$ has been evaluated. Our algorithms construct an infinite sequence of approximating cost functions $\{f^*(\epsilon,\cdot)\}_{\epsilon\in\mathbb{R}^q_+}$ so that $f^*(\epsilon,\cdot)$ converges to $f(\cdot)$ fast enough near stationary points. However, since our GPS algorithms include global search and local search stages, as is typical in GPS algorithms, our GPS algorithms allow the use of surrogate models of $f^*(\epsilon,\cdot)$ to obtain points for the global search.

In implicit filtering [16], Kelley accounts for the situation where $f(\cdot)$ is approximated numerically using a computer program that contains adaptive solvers. In implicit filtering, however, one does not adaptively control the error of the cost function evaluations, but rather assumes in proving convergence to a stationary point of $f(\cdot)$ that the error of the approximating cost function decays faster to zero than the step size used in the finite difference approximation of the gradient of the cost function. Because in our convergence analysis we establish a lower bound for Clarke's generalized directional derivative, which we bound by a sequence of finite difference approximations, we need to assume the same rate of error decay as is assumed in implicit filtering. However, in contrast to implicit filtering, our algorithms adaptively control the approximation error. This allows us to construct a simulation precision control algorithm that causes the optimization to use computationally cheap coarse precision approximations to the cost function in the early iterations, and progressively use higher precision cost function evaluations as needed when the algorithm approaches a stationary point.

2. Notation.

1. We denote by $\mathbb{Z}$ the set of integers, by $\mathbb{Q}$ the set of rational numbers, and by $\mathbb{N} \triangleq \{0, 1, \dots\}$ the set of natural numbers. The set $\mathbb{N}_+$ is defined as $\mathbb{N}_+ \triangleq \{1, 2, \dots\}$. Similarly, vectors in $\mathbb{R}^n$ with strictly positive elements are denoted by $\mathbb{R}^n_+ \triangleq \{x \in \mathbb{R}^n \mid x^i > 0, \text{for all } i \in \{1,\dots,n\}\}$ and the set $\mathbb{Q}_+$ is defined as $\mathbb{Q}_+ \triangleq \{q \in \mathbb{Q} \mid q > 0\}$.
2. The inner product in $\mathbb{R}^n$ is denoted by $\langle\cdot,\cdot\rangle$ and for $x, y \in \mathbb{R}^n$ it is defined by $\langle x, y\rangle \triangleq \sum_{i=1}^n x^i\, y^i$.
3. For $\epsilon \in \mathbb{R}^q_+$, by $\epsilon \le \epsilon_{\mathbf{S}}$, we mean that $0 < \epsilon^i \le \epsilon^i_{\mathbf{S}}$ for all $i = \{1,\dots,q\}$.
4. If a subsequence $\{x_i\}_{i\in\mathbf{K}} \subset \{x_i\}_{i=0}^\infty$ converges to some point x, we write $x_i \to^{\mathbf{K}} x$.
5. Let $\mathbb{W}$ be a set containing a sequence $\{w_i\}_{i=0}^k$. Then, we denote by $\underline{w}_k$ the sequence $\{w_i\}_{i=0}^k$ and by $\underline{\mathbf{W}}_k$ the set of all $k+1$ element sequences in $\mathbb{W}$.
6. We denote by $\{e_i\}_{i=1}^n$ the unit vectors in $\mathbb{R}^n$.
7. If $\mathbf{X}$ is a set, we denote by $\partial\mathbf{X}$ its boundary.
8. If $\mathbf{S}$ is a set, we denote by $2^{\mathbf{S}}$ the set of all nonempty subsets of $\mathbf{S}$.
9. If $\mathcal{H}$ is a set, we denote by $\operatorname{card}(\mathcal{H})$ its cardinality.
10. If $D \in \mathbb{Q}^{n\times q}$ is a matrix, we will use the notation $d \in D$ to denote the fact that $d \in \mathbb{Q}^n$ is a column vector of the matrix D.
11. For $s \in \mathbb{R}$, we define $\lfloor s \rfloor \triangleq \max\{k \in \mathbb{Z} \mid k \le s\}$.

3. Minimization problem. To best explain our precision control algorithm without having to discuss technical details of constructing search directions that conform, in the sense of [3], to the feasible set of design parameters, and to avoid having to discuss the problem of degenerate search directions, we restrict our discussion to box-constrained problems rather than to problems with general linear constraints. The construction of search directions for linearly constrained problems is discussed in [17, 3, 24].

We will consider box-constrained problems

$$\min_{x \in \mathbf{X}} f(x), \tag{1a}$$

$$\mathbf{X} \triangleq \big\{x \in \mathbb{R}^n \mid l^i \leq x^i \leq u^i,\ i \in \{1, \ldots, n\}\big\}, \tag{1b}$$

with $-\infty \leq l^i < u^i \leq \infty$ for $i \in \{1, \ldots, n\}$, where the cost function $f\colon \mathbb{R}^n \to \mathbb{R}$ is (at least) Lipschitz continuous.

We assume that the function $f(\cdot)$ cannot be evaluated exactly, but that it can be approximated by functions $f^*\colon \mathbb{R}^q_+ \times \mathbb{R}^n \to \mathbb{R}$, and that $\epsilon \in \mathbb{R}^q_+$ is a vector of fixed dimension $q \in \mathbb{N}$ that contains the tolerance settings of the PDE, ODE, and algebraic equation solvers. We will assume that $f(\cdot)$ and its approximating functions $\{f^*(\epsilon, \cdot)\}_{\epsilon \in \mathbb{R}^q_+}$ have the following properties.

Assumption 3.1.

1. There exists an error bound function $\varphi\colon \mathbb{R}^q_+ \to \mathbb{R}_+$ such that for any compact set $\mathbf{S} \subset \mathbf{X}$, there exists an $\epsilon_{\mathbf{S}} \in \mathbb{R}^q_+$ and a scalar $K_{\mathbf{S}} \in (0, \infty)$ such that for all $x \in \mathbf{S}$ and for all $\epsilon \in \mathbb{R}^q_+$, with $\epsilon \leq \epsilon_{\mathbf{S}}$,

$$|\, f^*(\epsilon, x) - f(x)| \leq K_{\mathbf{S}}\, \varphi(\epsilon). \tag{2a}$$

Furthermore,

$$\lim_{\|\epsilon\| \to 0} \varphi(\epsilon) = 0. \tag{2b}$$

2. The function $f\colon \mathbb{R}^n \to \mathbb{R}$ is at least locally Lipschitz continuous on $\mathbf{X}$.

Remark 3.2. The functions $\{f^*(\epsilon, \cdot)\}_{\epsilon \in \mathbb{R}^q_+}$ may be discontinuous.

Examples of error bound functions $\varphi(\cdot)$ can be found in section 6 and in [24].

Next, we state an assumption on the level sets of the family of approximating cost functions. To do so, we first define the notion of a level set.

Definition 3.3 (level set). *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$ and an $\alpha \in \mathbb{R}$, we will say that the set $\mathbf{L}_\alpha(f) \subset \mathbb{R}^n$, defined as*

$$\mathbf{L}_\alpha(f) \triangleq \{x \in \mathbb{R}^n \mid f(x) \leq \alpha\}, \tag{3}$$

is a level set *of $f(\cdot)$, parametrized by α.*

Assumption 3.4 (compactness of level sets). Let $\{f^*(\epsilon, \cdot)\}_{\epsilon \in \mathbb{R}^q_+}$ be as in Assumption 3.1 and let $\mathbf{X} \subset \mathbb{R}^n$ be the constraint set. Let $x_0 \in \mathbf{X}$ be the initial iterate and $\epsilon_0 \in \mathbb{R}^q_+$ be the initial solver tolerance. Then, we assume that there exists a compact set $\mathbf{C} \subset \mathbb{R}^n$ such that for all $\epsilon \in \mathbb{R}^q_+$, with $\epsilon \leq \epsilon_0$,

$$\mathbf{L}_{f^*(\epsilon_0, x_0)}\big(f^*(\epsilon, \cdot)\big) \cap \mathbf{X} \subset \mathbf{C}. \tag{4}$$

4. Precision control for generalized pattern search algorithms.

4.1. Characterization of generalized pattern search algorithms. There exist different geometrical characterizations for pattern search algorithms, and a general framework is presented in the review [17]. To focus on the explanation of our precision control algorithms without having to repeat the excellent discussions in [17], we will use a simple implementation of pattern search algorithms to explain our precision control algorithms. In particular, we will assume that the search directions are the columns of the matrix

$$D \triangleq [-e_1,\, +e_1,\, \dots,\, -e_n,\, +e_n] \in \mathbb{Z}^{n \times 2n}, \tag{5}$$

which suffices for box-constrained problems. Furthermore, we assume that the sequence of mesh size parameters, which parameterize the minimum distance between iterates, is constructed as follows.

Assumption 4.1 (*k*th mesh size parameter). Let $r, s_0, k \in \mathbb{N}$, with $r > 1$, and $\{t_i\}_{i=0}^{k-1} \subset \mathbb{N}$. We assume that the sequence of mesh size parameters $\{\Delta_k\}_{k=0}^{\infty}$ satisfies

$$\Delta_k \triangleq \frac{1}{r^{s_k}}, \tag{6a}$$

where for $k > 0$

$$s_k \triangleq s_0 + \sum_{i=0}^{k-1} t_i. \tag{6b}$$

With this construction, all iterates lie on nested rational meshes of the form

$$\mathbb{M}_k \triangleq \{x_0 + \Delta_k\, D\, m \mid m \in \mathbb{N}^{2n}\}. \tag{7}$$

We will now characterize the set-valued maps that determine the mesh points for the "global" and "local" searches.

DEFINITION 4.2. *Let $\underline{\mathbf{X}}_k \subset \mathbb{R}^n$ and $\underline{\mathbf{\Delta}}_k \subset \mathbb{Q}_+$ be the sets of all sequences containing $k+1$ elements, let $\mathbb{M}_k$ be the current mesh as defined in* (7), *and let $\epsilon_k \in \mathbb{R}^q_+$ be the solver tolerance.*

1. *We define the* global search set map *to be any set-valued map*

   $$\gamma_k \colon \underline{\mathbf{X}}_k \times \underline{\mathbf{\Delta}}_k \times \mathbb{R}^q_+ \to \left(2^{\mathbb{M}_k} \cap \mathbf{X}\right) \cup \emptyset \tag{8a}$$

   whose image $\gamma_k(\underline{x}_k, \underline{\Delta}_k, \epsilon_k)$ contains only a finite number of mesh points.
2. *We will call $\mathcal{G}_k \triangleq \gamma_k(\underline{x}_k, \underline{\Delta}_k, \epsilon_k)$ the* global search set.
3. *We define the directions for the local search as $D \triangleq [-e_1,\, +e_1,\, \dots,\, -e_n,\, +e_n]$.*
4. *We will call*

   $$\mathcal{L}_k \triangleq \bigl\{x_k \pm \Delta_k\, e_i \mid i \in \{1, \dots, n\}\bigr\} \cap \mathbf{X} \tag{8b}$$

   the local search set.

Remark 4.3.

1. As we shall see, the global search affects only the efficiency of the algorithm but not its convergence properties. Any heuristic procedure that leads to a finite number of function evaluations can be used for $\gamma_k(\cdot,\cdot,\cdot)$. Thus, the elements in $\mathcal{G}_k$ can be determined using a search procedure on surrogate cost functions, as in [10, 27, 25, 4], if the search procedure is a finite process.
2. The empty set is included in the range of $\gamma_k(\cdot,\cdot,\cdot)$ to allow omitting the global search.

4.2. Adaptive precision GPS algorithm models. We will now present our GPS algorithm models with adaptive precision cost function evaluations. We will first present an algorithm that simultaneously decreases Δ_k and ϵ_k.

ALGORITHM 4.4 (GPS algorithm model with simultaneous decrease of ϵ_k and Δ_k).

```
Data:    Sufficient decrease parameter $\zeta \geq 0$;
         Initial iterate $x_0 \in \mathbf{X}$;
         Mesh size divider $r \in \mathbb{N}$, with $r > 1$;
         Initial mesh size exponent $s_0 \in \mathbb{N}$.
Maps:    Global search set map $\gamma_k \colon \underline{\mathbf{X}_k} \times \underline{\mathbf{\Delta}}_k \times \mathbb{R}^q_+ \to \big(2^{\mathbb{M}_k} \cap \mathbf{X}\big) \cup \emptyset$;
         $\varphi \colon \mathbb{R}^q_+ \to \mathbb{R}_+$ as in Assumption 3.1;
         Function $\rho \colon \mathbb{R}_+ \to \mathbb{R}^q_+$ (to assign $\epsilon_k$), such that the composition
         $\varphi \circ \rho \colon \mathbb{R}_+ \to \mathbb{R}_+$ is strictly monotone increasing and satisfies
         $\varphi(\rho(\Delta))/\Delta \to 0$, as $\Delta \to 0$.
Step 0:  Initialize $k = 0$, $\Delta_0 = 1/r^{s_0}$, and $\epsilon_0 = \rho(1)$.
Step 1:  Global Search
         Construct the global search set $\mathcal{G}_k = \gamma_k(\underline{x}_k, \underline{\Delta}_k, \epsilon_k)$.
         If $f^*(\epsilon_k, x') - f^*(\epsilon_k, x_k) < -\zeta\,\varphi(\epsilon_k)$ for any $x' \in \mathcal{G}_k$, go to Step 3;
         else, go to Step 2.
Step 2:  Local Search
         Evaluate $f^*(\epsilon_k, \cdot)$ for any $x' \in \mathcal{L}_k$ until some $x' \in \mathcal{L}_k$
         satisfying $f^*(\epsilon_k, x') - f^*(\epsilon_k, x_k) < -\zeta\,\varphi(\epsilon_k)$ is obtained, or until all points
         in $\mathcal{L}_k$ are evaluated.
Step 3:  Parameter Update
         If there exists an $x' \in \mathcal{G}_k \cup \mathcal{L}_k$ satisfying $f^*(\epsilon_k, x') - f^*(\epsilon_k, x_k) < -\zeta\,\varphi(\epsilon_k)$,
         set $x_{k+1} = x'$, $s_{k+1} = s_k$, $\Delta_{k+1} = \Delta_k$, and $\epsilon_{k+1} = \epsilon_k$;
         else, set $x_{k+1} = x_k$, $s_{k+1} = s_k + t_k$, with $t_k \in \mathbb{N}_+$ arbitrary,
         $\Delta_{k+1} = 1/r^{s_{k+1}}$, and $\epsilon_{k+1} = \rho(\Delta_{k+1}/\Delta_0)$.
Step 4:  Replace $k$ by $k + 1$, and go to Step 1.
```

Remark 4.5.

1. We allow setting $\zeta = 0$ to obtain a GPS algorithm without imposing a sufficient decrease condition. In proving that $\lim_{k\to\infty} \Delta_k = 0$, we will make use of the fact that the iterates x_k are contained in a compact set and lie on a rational lattice in which the spacing of the elements depends on Δ_k, and hence simple decrease suffices to accept an iterate [17].
2. To ensure that ϵ_0 does not depend on the scaling of Δ_0, we normalize the argument of $\rho(\cdot)$.
3. In Step 2, once a (sufficient) decrease in cost is obtained, one can proceed to Step 3. But note that one is allowed to evaluate the approximating cost function at more points in $\mathcal{L}_k$ in an attempt to obtain a larger decrease in cost. However, one is allowed to proceed to Step 3 only after either a (sufficient) decrease in cost has been obtained or after *all* points in $\mathcal{L}_k$ were tested.
4. In Step 3, one is not restricted to accept the point $x' \in \mathcal{G}_k \cup \mathcal{L}_k$ that gives lowest cost. But the mesh size parameter Δ_k is reduced *only* if there exists no $x' \in \mathcal{G}_k \cup \mathcal{L}_k$ satisfying $f^*(\epsilon_k, x') - f^*(\epsilon_k, x_k) < -\zeta\,\varphi(\epsilon_k)$.
5. To simplify the explanation of our precision control algorithms, we do not increase the mesh size parameter if the cost has been reduced. However, our global search allows searching on a coarser mesh $\widehat{\mathbb{M}} \subset \mathbb{M}_k$, and hence our algorithm can easily be extended to include a rule for increasing Δ_k for a

finite number of iterations (see [24]).

6. Audet and Dennis [3] update the mesh size parameter using the formula $\Delta_{k+1} = \tau^{w_k}\,\Delta_k$, where $\tau > 1$ is a rational number that remains constant over all iterations, and where $w_k \in [w^-, w^+] \subset \mathbb{Z}$ for some constants $w^-, w^+ \in \mathbb{Z}$, with $w^- \leq -1$ and $0 \leq w^+$. In the notation of Audet and Dennis, our algorithm sets $\tau = r \in \mathbb{N}_+$, $r > 1$ and $w_k \leq 0$, $w_k \in \mathbb{Z}$. Our GPS algorithms do not require $t_k \in \mathbb{N}_+$ to be bounded from above (i.e., in the notation of Audet and Dennis [3], w_k need not be bounded from below by w^-). However, it is obvious that from a computational point of view, a small t_k should be selected to avoid too fine a mesh in early iterations. We prefer presenting our GPS algorithms with our construction of mesh size dividers because it leads to simpler convergence proofs that better highlight our precision control scheme.

In [24], we use the GPS Algorithm model 4.4 with $\zeta = 0$ to extend the Hooke–Jeeves algorithm for use with adaptive precision cost function evaluations.

We will now present a modification of the GPS Algorithm model 4.4 in which we decrease Δ_k only after $\varphi(\epsilon_k)$ has been sufficiently decreased.

Algorithm 4.6 (GPS algorithm model).

```
Data:    Parameters $\alpha \in (0, 1)$ and $\zeta \geq 0$;
         Initial iterate $x_0 \in \mathbf{X}$;
         Mesh size divider $r \in \mathbb{N}$, with $r > 1$;
         Initial mesh size exponent $s_0 \in \mathbb{N}$.
Maps:    Global search set map $\gamma_k \colon \underline{\mathbf{X}_k} \times \underline{\mathbf{\Delta}_k} \times \mathbb{R}^q_+ \to \left(2^{\mathbb{M}_k} \cap \mathbf{X}\right) \cup \emptyset$;
         $\varphi \colon \mathbb{R}^q_+ \to \mathbb{R}_+$ as in Assumption 3.1;
         $\rho \colon \mathbb{N} \to \mathbb{R}^q_+$ (to assign $\epsilon_k$) such that the
         composition $\varphi \circ \rho \colon \mathbb{N} \to \mathbb{R}_+$ is strictly monotone decreasing and
         satisfies $\varphi(\rho(N)) \to 0$, as $N \to \infty$.
Step 0:  Initialize $k = 0$, $\Delta_0 = 1/r^{s_0}$, $N = 1$, and $\epsilon_0 = \rho(1)$.
Step 1:  Global Search
         Construct the global search set $\mathcal{G}_k = \gamma_k(\underline{x}_k, \underline{\Delta}_k, \epsilon_k)$.
         If $f^*(\epsilon_k, x') - f^*(\epsilon_k, x_k) < -\zeta\,\varphi(\epsilon_k)$ for any $x' \in \mathcal{G}_k$, go to Step 3;
         else, go to Step 2.
Step 2:  Local Search
         Evaluate $f^*(\epsilon_k, \cdot)$ for any $x' \in \mathcal{L}_k$ until some $x' \in \mathcal{L}_k$
         satisfying $f^*(\epsilon_k, x') - f^*(\epsilon_k, x_k) < -\zeta\,\varphi(\epsilon_k)$ is obtained, or until all points
         in $\mathcal{L}_k$ are evaluated.
Step 3:  Parameter Update
         If there exists an $x' \in \mathcal{G}_k \cup \mathcal{L}_k$ satisfying $f^*(\epsilon_k, x') - f^*(\epsilon_k, x_k) < -\zeta\,\varphi(\epsilon_k)$,
                 set $x_{k+1} = x'$, $s_{k+1} = s_k$, $\Delta_{k+1} = \Delta_k$, and $\epsilon_{k+1} = \epsilon_k$, do not change
                 $N$, and go to Step 4;
         else,
                 replace $N$ by $N + 1$ and set $\epsilon_{k+1} = \rho(N)$.
                 If $\varphi(\epsilon_{k+1})^\alpha/\Delta_k < \Delta_k$,
                         set $s_{k+1} = s_k + t_k$, with $t_k \in \mathbb{N}_+$ large enough
                         such that $\varphi(\epsilon_{k+1})^\alpha/\Delta_{k+1} \geq \Delta_{k+1}$ (with $\Delta_{k+1} = 1/r^{s_{k+1}}$),
                         and set $\Delta_{k+1} = 1/r^{s_{k+1}}$;
                 else,
                    set $s_{k+1} = s_k$ and $\Delta_{k+1} = \Delta_k$.
                Set $x_{k+1} = x_k$, and go to Step 4.
Step 4:   Replace $k$ by $k + 1$, and go to Step 1.
```

Remark 4.7.

1. In Step 3, the mesh refinement is well defined because there always exists a $t_k \in \mathbb{N}_+$ for which $\varphi(\epsilon_{k+1})^\alpha \geq \Delta_{k+1}^2$, namely any $t_k \in \mathbb{N}_+$ that satisfies

$$t_k \geq \frac{2\,\log \Delta_k - \alpha\,\log \varphi(\epsilon_{k+1})}{2\,\log r}. \tag{9}$$

2. In Step 3, the test $\varphi(\epsilon_{k+1})^\alpha/\Delta_k < \Delta_k$, with $\alpha \in (0,\,1)$, ensures that $\varphi(\epsilon_k)/\Delta_k \to 0$, as $\Delta_k \to 0$, which is essential for proving convergence.
3. The algorithm parameter α can be used to control how fast Δ_k is decreased. The smaller $\alpha \in (0,\,1)$, the later Δ_k is decreased.

5. Convergence analysis.

5.1. Unconstrained minimization. We will now establish the convergence properties of the GPS Algorithm models 4.4 and 4.6 on unconstrained minimization problems, i.e., for $\mathbf{X} = \mathbb{R}^n$.

The following obvious result will be used to show that $\Delta_k \to 0$ as $k \to \infty$.

Proposition 5.1. *Any bounded subset of a mesh $\mathbb{M}_k$ contains only a finite number of mesh points.*

Proposition 5.2. *Suppose that Assumption* 3.4 *is satisfied and let $\{\Delta_k\}_{k=0}^\infty \subset \mathbb{Q}_+$ be the sequence of mesh size parameters constructed by GPS Algorithm model* 4.4 *or* 4.6. *Then,* $\lim_{k\to\infty} \Delta_k = 0$.

Proof. We first prove the proposition for the GPS Algorithm model 4.4. By (6a), $\Delta_k = 1/r^{s_k}$, where $r \in \mathbb{N}$ with $r > 1$ and $\underline{s}_k \subset \mathbb{N}$ is a nondecreasing sequence. For the sake of contradiction, suppose that there exists a $\Delta_{k^*} \in \mathbb{Q}_+$ such that $\Delta_k \geq \Delta_{k^*}$ for all $k \in \mathbb{N}$. Then there exists a corresponding $s_{k^*} = \max_{k\in\mathbb{N}} s_k$, and the finest possible mesh is $\mathbb{M}_{k^*} \triangleq \{x_0 + (1/r^{s_{k^*}})\,D\,m \,|\, m \in \mathbb{N}^{2n}\}$.

Next, since by Assumption 3.4 there exists a compact set $\mathbf{C}$, such that $\mathbf{L}_{f^*(\epsilon_0,x_0)}\big(f^*(\epsilon,\cdot)\big) \subset \mathbf{C}$ for all $\epsilon \in \mathbb{R}_+^q$, with $\epsilon \leq \epsilon_0 = \rho(1)$, it follows from Proposition 5.1 that $\mathbb{M}_{k^*} \cap \mathbf{L}_{f^*(\epsilon_0,x_0)}\big(f^*(\epsilon_k,\cdot)\big)$ contains only a finite number of mesh points for any $\epsilon_k \in \mathbb{R}_+^q$, with $\epsilon_k \leq \rho(1)$. Thus, at least one point in $\mathbb{M}_{k^*}$ must belong to the sequence $\{x_k\}_{k=0}^\infty$ infinitely many times. Furthermore, because $\{s_k\}_{k=0}^\infty \subset \mathbb{N}$ is nondecreasing with s_{k^*} being its maximal element, it follows that $\epsilon_k = \epsilon_{k^*} = \rho(\Delta_{k^*}/\Delta_0)$ for all iterations $k \geq k^*$. Hence the sequence $\{f^*(\epsilon_k, x_k)\}_{k=k^*}^\infty$ cannot satisfy $f^*(\epsilon_k, x_{k+1}) - f^*(\epsilon_k, x_k) < -\zeta\,\varphi(\epsilon_k)$ for all $k \geq k^*$, which contradicts the constructions in Algorithm 4.4.

We now prove the proposition for the GPS Algorithm model 4.6. Suppose $\lim_{k\to\infty} \Delta_k \neq 0$. Then, there exists only a finite number of iterations in which there exists no $x' \in \mathcal{G}_k \cup \mathcal{L}_k$ that satisfies $f^*(\epsilon_k, x') - f^*(\epsilon_k, x_k) < -\zeta\,\varphi(\epsilon_k)$, because otherwise N is replaced by $N + 1$ an infinite number of times in Step 3, from which follows that $\varphi(\rho(N))^\alpha \to 0$, as $N \to \infty$, and hence $\Delta_k \to 0$, as $k \to \infty$. Thus, there exists an $N^* \in \mathbb{N}$ and a corresponding $k^* \in \mathbb{N}$ such that $N \leq N^*$, $\Delta_k = \Delta_{k^*}$, and $\epsilon_k = \epsilon_{k^*} = \rho(N^*)$ for all $k \geq k^*$, and the finest possible mesh is $\mathbb{M}_{k^*} \triangleq \big\{x_0 + \Delta_{k^*}\,D\,m \mid m \in \mathbb{N}^{2\,n}\big\}$.

By Assumption 3.4, there exists a compact set $\mathbf{C}$, such that $\mathbf{L}_{f^*(\epsilon_0,x_0)}\big(f^*(\epsilon,\cdot)\big) \subset \mathbf{C}$, for all $\epsilon \in \mathbb{R}_+^q$, with $\epsilon \leq \epsilon_0 = \rho(1)$. Hence, it follows from Proposition 5.1 that

$\mathbb{M}_{k^*} \cap \mathbf{L}_{f^*(\epsilon_0, x_0)}\big(f^*(\epsilon_k, \cdot)\big)$ contains only a finite number of mesh points for all $\epsilon_k \le \epsilon_0$. Thus, at least one point in $\mathbb{M}_{k^*}$ must belong to the sequence $\{x_k\}_{k=0}^{\infty}$ infinitely many times. Hence, the sequence $\{f^*(\epsilon_k, x_k)\}_{k=k^*}^{\infty}$ cannot satisfy $f^*(\epsilon_k, x_{k+1}) - f^*(\epsilon_k, x_k) < -\zeta\, \varphi(\epsilon_k)$ for all $k \ge k^*$, which contradicts the constructions in Algorithm 4.6. □

Having shown that $\lim_{k\to\infty} \Delta_k = 0$, we can use the notion of a refining subsequence as introduced by Audet and Dennis [3].

Definition 5.3 (refining subsequence). *Consider a sequence $\{x_k\}_{k=0}^{\infty}$ constructed by GPS Algorithm model* 4.4 *or* 4.6*. We will say that the subsequence $\{x_k\}_{k\in\mathbf{K}}$ is the* refining subsequence*, if $\Delta_{k+1} < \Delta_k$ for all $k \in \mathbf{K}$, and $\Delta_{k+1} = \Delta_k$ for all $k \notin \mathbf{K}$.*

When the cost function $f(\cdot)$ is only locally Lipschitz continuous, we, as well as Audet and Dennis [3], only get a weak characterization of limit points of the refining subsequence, as we will now see.

We recall the definition of Clarke's generalized directional derivative [8].

Definition 5.4 (Clarke's generalized directional derivative). *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz continuous at the point $x^* \in \mathbb{R}^n$. Then,* Clarke's generalized directional derivative *of $f(\cdot)$ at x^* in the direction $h \in \mathbb{R}^n$ is defined by*

$$f^\circ(x^*;\, h) \triangleq \limsup_{\substack{x\to x^*\\ t\downarrow 0}} \frac{f(x + t\,h) - f(x)}{t}. \tag{10}$$

Theorem 5.5. *Suppose that Assumptions* 3.1 *and* 3.4 *are satisfied, let D be as in Definition* 4.2*, and let $x^* \in \mathbb{R}^n$ be an accumulation point of the refining subsequence $\{x_k\}_{k\in\mathbf{K}}$ constructed by GPS Algorithm model* 4.4 *or* 4.6*. Then, for all $d \in D$,*

$$f^\circ(x^*;\, d) \ge 0. \tag{11}$$

Proof. The proof is identical for both algorithms. Let $\{x_k\}_{k\in\mathbf{K}}$ be the refining subsequence and, without loss of generality, suppose that $x_k \to^{\mathbf{K}} x^*$. By Assumption 3.4, there exists a compact set $\mathbf{C}$ such that $\mathbf{L}_{f^*(\epsilon_0,x_0)}\big(f^*(\epsilon,\cdot)\big) \subset \mathbf{C}$ for all $\epsilon \in \mathbb{R}^q_+$, with $\epsilon \le \epsilon_0 = \rho(1)$. Therefore, by Assumption 3.1, there exists an $\epsilon_{\mathbf{L}} \in \mathbb{R}^q_+$ and a scalar $K_{\mathbf{L}} \in (0,\infty)$ such that, for all $x \in \mathbf{C}$ and for all $\epsilon \in \mathbb{R}^q_+$, with $\epsilon \le \epsilon_{\mathbf{L}}$, we have $|f^*(\epsilon, x) - f(x)| \le K_{\mathbf{L}}\, \varphi(\epsilon)$. Because $f(\cdot)$ is locally Lipschitz continuous, its directional derivative $f^\circ(\cdot;\cdot)$ exists. The precision control schemes of Algorithms 4.4 and 4.6 both imply that $\epsilon_k \to^{\mathbf{K}} 0$, and furthermore that $f^*(\epsilon_k, x_k + \Delta_k\, d) - f^*(\epsilon_k, x_k) \ge -\zeta\, \varphi(\epsilon_k)$ for all $d \in D$ and for all $k \in \mathbf{K}$. Hence, for any $d \in D$,

$$\begin{aligned} f^\circ(x^*;\, d) &\triangleq \limsup_{\substack{x\to x^*\\ t\downarrow 0}} \frac{f(x + t\,d) - f(x)}{t} \\ &\ge \limsup_{k\in\mathbf{K}} \frac{f(x_k + \Delta_k\, d) - f(x_k)}{\Delta_k} \\ &\ge \limsup_{k\in\mathbf{K}} \frac{f^*(\epsilon_k, x_k + \Delta_k\, d) - f^*(\epsilon_k, x_k) - 2\,K_{\mathbf{L}}\, \varphi(\epsilon_k)}{\Delta_k} \\ &\ge \limsup_{k\in\mathbf{K}} \frac{f^*(\epsilon_k, x_k + \Delta_k\, d) - f^*(\epsilon_k, x_k)}{\Delta_k} - \limsup_{k\in\mathbf{K}} 2\,K_{\mathbf{L}}\, \frac{\varphi(\epsilon_k)}{\Delta_k} \\ &\ge -\limsup_{k\in\mathbf{K}} \zeta\, \frac{\varphi(\epsilon_k)}{\Delta_k} - \limsup_{k\in\mathbf{K}} 2\,K_{\mathbf{L}}\, \frac{\varphi(\epsilon_k)}{\Delta_k}. \qquad (12) \end{aligned}$$

The second line follows from the first because the choice of x and t is restricted to $\{x_k\}_{k\in\mathbf{K}}$ and $\{\Delta_k\}_{k\in\mathbf{K}}$. Since by Proposition 5.2 $\Delta_k \to^{\mathbf{K}} 0$, it follows from the constructions in GPS Algorithm models 4.4 and 4.6 that $\varphi(\epsilon_k)/\Delta_k \to^{\mathbf{K}} 0$. □

We now state that pattern search algorithms with adaptive precision cost function evaluations converge to stationary points.

THEOREM 5.6 (convergence to a stationary point). *Suppose that Assumptions* 3.1 *and* 3.4 *are satisfied and, in addition, that* $f(\cdot)$ *is once continuously differentiable. Let* $x^* \in \mathbb{R}^n$ *be an accumulation point of the refining subsequence* $\{x_k\}_{k \in \mathbf{K}}$*, constructed by GPS Algorithm model* 4.4 *or* 4.6*. Then,*

$$\nabla f(x^*) = 0. \tag{13}$$

Proof. Because $f(\cdot)$ is once continuously differentiable, we have $f^\circ(x^*; h) = df(x^*; h) = \langle \nabla f(x^*), h \rangle$ for all $h \in \mathbb{R}^n$. It follows from Theorem 5.5 that $0 \le \langle \nabla f(x^*), d \rangle$ for all $d \in D$, with D as in Definition 4.2. We can express any $h \in \mathbb{R}^n$ as

$$h = \sum_{i=1}^{2n} \alpha_i \, d_i, \quad d_i \in D, \quad \alpha_i \ge 0 \quad \forall\, i \in \{1, \ldots, 2n\}. \tag{14a}$$

Hence, $0 \le \langle \nabla f(x^*), h \rangle$. Similarly, we can express the vector $-h$, as follows:

$$-h = \sum_{i=1}^{2n} \beta_i \, d_i, \quad d_i \in D, \quad \beta_i \ge 0 \quad \forall\, i \in \{1, \ldots, 2n\}. \tag{14b}$$

Hence, $0 \ge \langle \nabla f(x^*), h \rangle$, which implies that $0 = \langle \nabla f(x^*), h \rangle$ and, since h is arbitrary, that $\nabla f(x^*) = 0$. □

5.2. Constrained minimization. We will now extend our convergence proofs to the box-constrained problem (1). First, we introduce the notion of a tangent cone and a normal cone, which are defined as follows.

DEFINITION 5.7 (tangent and normal cone).

1. *Let* $\mathbf{X} \subset \mathbb{R}^n$ *be defined as in* (1b)*. Then, we define the* tangent cone *to* $\mathbf{X}$ *at a point* $x^* \in \mathbf{X}$ *by*

$$\mathbf{T_X}(x^*) \triangleq \{\mu\,(x - x^*) \mid \mu \ge 0,\, x \in \mathbf{X}\}. \tag{15a}$$

2. *Let* $\mathbf{T_X}(x^*)$ *be as above. Then, we define the* normal cone *to* $\mathbf{X}$ *at* $x^* \in \mathbf{X}$ *by*

$$\mathbf{N_X}(x^*) \triangleq \{v \in \mathbb{R}^n \mid \forall\, t \in \mathbf{T_X}(x^*),\, \langle v, t \rangle \le 0\}. \tag{15b}$$

Next, for $x^* \in \mathbf{X}$ we define the subset of column vectors of the search direction matrix $D \triangleq [-e_1, +e_1, \ldots, -e_n, +e_n]$ that is required to generate the tangent cone $\mathbf{T_X}(x^*)$. This will facilitate the extension of Theorem 5.5 to box-constrained problems.

DEFINITION 5.8. *Let* $\mathbf{X} \subset \mathbb{R}^n$ *be as in* (1b) *and let* $\mathbf{T_X}(\cdot)$ *be as in Definition* 5.7*. For* $x^* \in \mathbf{X}$*, we define* $\mathcal{H}(x^*) \subset \{-e_1, +e_1, \ldots, -e_n, +e_n\}$ *such that* $\mathbf{T_X}(x^*) = \{\sum_{i=1}^{\operatorname{card} \mathcal{H}(x^*)} \alpha^i \, h_i \mid h_i \in \mathcal{H}(x^*),\, \alpha^i \ge 0,\, i \in \{1, \ldots, \operatorname{card} \mathcal{H}(x^*)\}\}$.

THEOREM 5.9. *Suppose that Assumptions* 3.1 *and* 3.4 *are satisfied. Let* $x^* \in \mathbf{X}$ *be an accumulation point of the refining subsequence* $\{x_k\}_{k \in \mathbf{K}}$ *constructed by GPS Algorithm model* 4.4 *or* 4.6 *in solving problem* (1) *and let* $\mathcal{H}(x^*)$ *be as defined in Definition* 5.8*. Then,*

$$f^\circ(x^*; d) \ge 0 \qquad \forall\, d \in \mathcal{H}(x^*). \tag{16}$$

Proof. If x^* is in the interior of $\mathbf{X}$, then the result reduces to Theorem 5.5.

Hence, suppose that $x^* \in \partial\mathbf{X}$ and let $\mathcal{H}(x^*)$ be as in Definition 5.8. Because x^* is an accumulation point of $\{x_k\}_{k\in\mathbf{K}}$, there exists an infinite subset $\mathbf{K}' \subset \mathbf{K}$, such that $x_k \to^{\mathbf{K}'} x^*$, and $\{x_k + \Delta_k\, d\}_{k\in\mathbf{K}',\, d\in\mathcal{H}(x^*)} \subset \mathbf{X}$. The precision control schemes of Algorithms 4.4 and 4.6 both imply that $\epsilon_k \to^{\mathbf{K}'} 0$, and furthermore that $f^*(\epsilon_k, x_k + \Delta_k\, d) - f^*(\epsilon_k, x_k) \geq -\zeta\, \varphi(\epsilon_k)$ for all $d \in \mathcal{H}(x^*)$ and for all $k \in \mathbf{K}'$. Thus, for any $d \in \mathcal{H}(x^*)$, (12) still holds, from which we conclude that (16) holds. □

We can now state that the GPS Algorithm models 4.4 and 4.6 generate sequences of iterates which contain accumulation points that are feasible stationary points of problem (1).

Theorem 5.10 (convergence to a feasible stationary point). *Suppose that Assumptions* 3.1 *and* 3.4 *are satisfied and, in addition, that* $f(\cdot)$ *is once continuously differentiable. Let* $x^* \in \mathbf{X}$ *be an accumulation point of the refining subsequence* $\{x_k\}_{k\in\mathbf{K}}$ *constructed by GPS Algorithm model* 4.4 *or* 4.6 *in solving problem* (1) *and let* $\mathcal{H}(x^*)$ *be as defined in Definition* 5.8. *Then,*

$$\langle \nabla f(x^*),\, t\rangle \geq 0 \qquad \forall\, t \in \mathbf{T}_{\mathbf{X}}(x^*), \tag{17a}$$

and

$$-\nabla f(x^*) \in \mathbf{N}_{\mathbf{X}}(x^*). \tag{17b}$$

Proof. Because $f(\cdot)$ is once continuously differentiable, it follows that $f^\circ(x^*; h) = \langle\nabla f(x^*), h\rangle$ for all $h \in \mathbb{R}^n$. Because $\langle\nabla f(x^*), \cdot\rangle$ is linear and because every $t \in \mathbf{T}_{\mathbf{X}}(x^*)$ can be expressed as a nonnegative linear combination of elements in $\mathcal{H}(x^*)$, we conclude in view of (16) that (17a) and (17b) hold. □

6. Numerical experiments.

In all numerical experiments, in the GPS Algorithm models 4.4 and 4.6 we set the mesh size divider $r = 2$ and the initial mesh size exponent $s_0 = 0$. If no (sufficient) decrease in cost has been obtained, then we divide the mesh size parameter Δ_k by a factor of two. Hence, if $\mathbf{K}$ denotes the set that contains the iteration indices of the refining subsequence, as defined in Definition 5.3, then in GPS Algorithm models 4.4 and 4.6 $t_k = 1$ for $k \in \mathbf{K}$ and $t_k = 0$ for $k \notin \mathbf{K}$.

All optimization runs were carried out using a 2.2 GHz AMD processor running Linux with the $2.4.18 - 3$ kernel.

6.1. Optimization problem with cost function defined on the solutions of a DAE system.

In this numerical experiment, we minimize the annual energy consumption for lighting, cooling, and heating of an office building in Houston, TX. We simulated three characteristic rooms that are representative of the energy consumption of a large office building. The components of the design parameter $x \in \mathbb{R}^n$ are the window sizes for the south and north facing windows, the depth of an overhang placed above the south facing window, and two control setpoints that activate shading devices outside the north and south facing windows, hence $n = 5$.

6.1.1. Exact cost function.

The cost function is once continuously differentiable and defined as

$$f(x) \triangleq F\big(z(x, 1)\big), \tag{18}$$

where $z(x,1)$ is the solution of a semiexplicit nonlinear DAE system with index one [5] of the form

$$\frac{dz(x,t)}{dt} = h\big(x, z(x,t), \mu\big), \qquad t \in [0,\ 1], \tag{19a}$$

$$z(x,0) = z_0(x), \tag{19b}$$

$$\gamma\big(x, z(x,t), \mu\big) = 0, \tag{19c}$$

where $h\colon \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^l \to \mathbb{R}^m$, $z_0\colon \mathbb{R}^n \to \mathbb{R}^m$, and $\gamma\colon \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^l \to \mathbb{R}^l$ are once Lipschitz continuously differentiable in all arguments. For all $x \in \mathbb{R}^n$ and for all $z(\cdot,\cdot) \in \mathbb{R}^m$, (19c) has a unique solution $\mu^*(x,z) \in \mathbb{R}^l$ and the matrix with partial derivatives $\partial\gamma(x, z(x,t), \mu^*(x,z))/\partial\mu \in \mathbb{R}^{l\times l}$ is nonsingular. Thus, by using the implicit function theorem and standard theory of ODEs [23] one can show that there exists a unique once continuously differentiable function $z(\cdot,1)$ and hence $f(\cdot)$ is once continuously differentiable. In this experiment, $m = 104$ and $l = 4$, and we defined the function $F(\cdot)$ in (18) as

$$F\big(z(x,1)\big) \triangleq \frac{z^1(x,1)}{\eta_h} + \frac{z^2(x,1)}{\eta_c} + 3\, z^3(x,1), \tag{20}$$

where $z^1(x,1)$ and $z^2(x,1)$ are the annual heating and cooling loads of the rooms, $z^3(x,1)$ is the electricity consumption for lighting the rooms, and $\eta_h = 0.44$ and $\eta_c = 0.77$ are plant efficiencies that relate the annual room load to the primary energy consumption for heating and cooling generation, including electricity consumption for fans and pumps [15]. The electricity consumption is multiplied by a factor of three to convert site electricity to source fuel energy consumption.

6.1.2. Approximating cost functions. To compute approximations to the cost function $f(\cdot)$, we had to write a thermal building and daylighting simulation program, called BuildOpt [29, 30], because existing thermal building and daylighting simulation programs are built on models that do not satisfy the smoothness assumptions required to prove existence, uniqueness, and differentiability of $z(\cdot,1)$.

BuildOpt is a complex program that consists of two parts. The first part, which we will call the *simulation model generator*, parses a text input file with the detailed description of the building geometry, the building materials, and the expected occupancy behavior—which, for our problem, was 1,700 lines long—and then generates a simulation model for the particular building, i.e., the functions $h(\cdot,\cdot,\cdot)$, $z_0(\cdot)$, and $\gamma(\cdot,\cdot,\cdot)$ of the DAE system (19). These functions are representations of various detailed models for the heat and daylighting transfer processes and for the building control systems. For example, the heat conduction in walls and ceilings are modeled using the Galerkin finite element method [11], and the transmittances of solar radiation and daylight through the windows are modeled using state-of-the-art optical calculations similar to those in commercial programs [12, 34]. There is also a detailed daylighting model that computes the available daylight at various locations in the building for different building and window configurations, and there are models for various control systems, such as the room lighting system and the heating and cooling system. The second part of BuildOpt, to which our simulation model generator was linked, is the commercial solver DASPK [6, 7].

The total size of BuildOpt is 38,000 lines of C/C++ and Fortran code, of which 30,000 lines (1.2 MB) of C/C++ code represent the simulation model generator and 8,000 lines (0.3 MB) of Fortran code represent the commercial solver DASPK.

Table 1

Normalized computation times required to solve the building energy optimization problem with Algorithm 4.6. *For each α, the last column shows the smallest Δ_k used in the search.*

	$\zeta = 0$	$\zeta = 10^{-8}$	$\zeta = 10^{-6}$	$\zeta = 10^{-4}$	$\zeta = 10^{-2}$	Δ_{k^*}
$\alpha = 1/7$	0.27	0.27	0.27	0.28	0.55	1/2
$\alpha = 1/6$	0.33	0.33	0.33	0.31	0.61	1/4
$\alpha = 1/4$	0.35	0.35	0.35	0.31	0.74	1/4
$\alpha = 1/3$	0.55	0.55	0.55	0.60	1.21	1/8

We constructed the models in such a way that the functions $h(\cdot,\cdot,\cdot)$, $z_0(\cdot)$, and $\gamma(\cdot,\cdot,\cdot)$ are once Lipschitz continuously differentiable in all arguments, which has not been done before for thermal building and daylighting simulation programs. This required various smoothing techniques to replace conditional statements, which were in fact required to achieve convergence of the DASPK solver when the solver tolerance was tight.

In [31], BuildOpt was validated using the ANSI/ASHRAE Standard test procedure 140-2001 [2] for the thermal models and benchmark tests [18, 13] produced in the Task 21 of the International Energy Agency (IEA) Solar Heating & Cooling Program for the daylighting models. The validation results of BuildOpt show good agreement with the results of the other validated programs.

The computation time for one cost function evaluation was 24 sec for a solver tolerance of $\epsilon = 10^{-1}$, 2 min 22 sec for $\epsilon = 10^{-2}$, 5 min 42 sec for $\epsilon = 10^{-3}$, 16 min 30 sec for $\epsilon = 10^{-4}$, and 33 min 23 sec for $\epsilon = 10^{-5}$.

6.1.3. Optimization algorithm. We solved the optimization problem with adaptive precision cost function evaluations using the Hooke–Jeeves algorithm of the GenOpt$^{(R)}$ 2.0.0 optimization program [28] with the precision controlled as in GPS Algorithm model 4.6. For comparison, we also solved the problem using the Hooke–Jeeves optimization algorithm with fixed precision cost function evaluations and $\zeta = 0$. In the optimization with fixed precision cost function evaluations, we set $\epsilon_k = 10^{-5}$ for all $k \in \mathbb{N}$ and we allowed the mesh size to be decreased four times before the optimization stopped.

6.1.4. Precision control. Present day DAE solvers, such as DASPK, typically control the local error at each time step and do not even attempt to control the global error directly. We assumed that the global error of the approximate solutions $z^*(\epsilon, x, 1)$ is one order of magnitude greater than the local error. Hence, we set $\varphi(\epsilon) = 10\,\epsilon$. (Alternatively, we could have absorbed the factor 10 in the constant $K_{\mathbf{S}}$ in (2a).)

We defined $\rho\colon \mathbb{N} \to \mathbb{R}_+$ as $\rho(N) = 10^{-N}$ and increased precision four times. Thus, $\epsilon_k = \rho(1) = 10^{-1}$ for the first iterations, and $\epsilon_k = \rho(5) = 10^{-5}$ for the last iterations, which is equal to the precision used in the optimization with fixed precision cost function evaluations.

6.1.5. Numerical results. In Table 1, we show the values that we selected for the algorithm parameters $\alpha \in (0, 1)$ and $\zeta \geq 0$, the corresponding normalized computation times, and in the last column the smallest mesh size parameter Δ_{k^*}. A computation time of 1 corresponds to 5.5 days of computing, which was the time required to solve the optimization problem with the Hooke–Jeeves algorithm with fixed precision cost function evaluations and $\zeta = 0$.

Note that in Algorithm 4.6, the parameter $\alpha \in (0, 1)$ is only used to adjust the mesh size parameter Δ_k so that $\varphi(\epsilon_k)^\alpha \geq \Delta_k^2$. Since $\varphi(\cdot)$ depends only on N, it is possible to compute for each $N \in \mathbb{N}$ the corresponding mesh size parameter. Such a computation shows that the sequence of mesh size parameters Δ_k, and hence the sequence of iterates x_k, are identical for all $\alpha \leq 1/7$, with $\alpha > 0$, and fixed ζ. Thus, a further reduction of α does not reduce the computation time.

For $\alpha \leq 1/4$, with $\zeta \in \{0, 10^{-8}, 10^{-6}, 10^{-4}\}$, our precision control algorithm reduces the computation time by a factor of three to four. For $\alpha = 1/7$ and $\zeta \leq 10^{-4}$, our precision control subprocedure reduced the computation time from about five days to one day, making the optimization fast enough to be applicable in building design processes. For our optimization problem, $\alpha = 1/3$ and $\zeta \geq 10^{-2}$ turn out to be too big, and imposing a sufficient decrease condition by setting $\zeta > 0$ does not reduce the computation time. All optimization runs converged to $x^* = (1, 1, 1, 0.19, 0.048)^T$ and reduced the energy consumption for lighting, cooling, and heating by 4.6 % or 9.4 kWh/(m^2 a).[1] The 4.6 % reduction is small but not representative for average savings, because in the literature [1, 32, 33] savings of 5% to 30% in energy consumption for lighting, cooling, and heating due to optimized building design have been reported. A reduction of 15%, which is more representative for average savings than the 4.6% that we achieved in our experiment, would correspond to a reduction in energy consumption for lighting, cooling, and heating of 30 kWh/(m^2 a). For an average current energy cost of \$0.10 per kWh, this corresponds to annual savings of \$3 per square meter floor area, or to annual savings of \$30,000 for a large, 10,000 m^2 office building. As large buildings are often designed using energy simulations, and hence a computer simulation model exists for those buildings, the additional effort to do an optimization is only a few man hours. Thus, the return-of-investment is achieved within the first year of the building operation time.

We will now describe how the optimization runs with fixed and adaptive precision cost function evaluations, with $\zeta = 10^{-4}$ and $\alpha = 1/6$, converged to a minimum. Let the normalized distance of the kth iterate $x_k \in \mathbb{R}^n$ to the minimizer $x^* \in \arg\min_{x \in \mathbf{X}} f(x)$, produced by the optimization algorithm, be defined as

$$d(x_k) \triangleq \frac{\|x_k - x^*\|}{\|x_0 - x^*\|}, \tag{21}$$

where $x_0 \in \mathbb{R}^n$ is the initial iterate. Figure 1 shows the cost function value and the distance to the minimizer as a function of the computation time. Below the axis we show when precision was increased. The different precision values are indicated by ϵ_m, $m \in \{0, 1, 2, 3, 4\}$, where $\epsilon_m = 10^{-(m+1)}$. The abscissa in Figure 1 shows the normalized CPU time rather than the number of cost function evaluations, because evaluating $f^*(10^{-1}, \cdot)$ was eighty times faster than evaluating $f^*(10^{-5}, \cdot)$ in our experiments (see also [29, 30]). In the left graph, we can see that even for such coarse a precision as $\epsilon = 10^{-1}$, the approximating cost function $f^*(10^{-1}, \cdot)$ allowed a substantial decrease in cost during the first 0.2% of the computation time.

6.2. Optimization problem with cost function defined on the solutions of a nonlinear system of equations. We will now present the computational performance of our precision control algorithms in minimizing a cost function that is defined on the approximate solutions of a nonlinear system of equations with 373 unknowns.

[1] The unit kWh/(m^2 a) is kilowatt hours per square meter floor area per year.

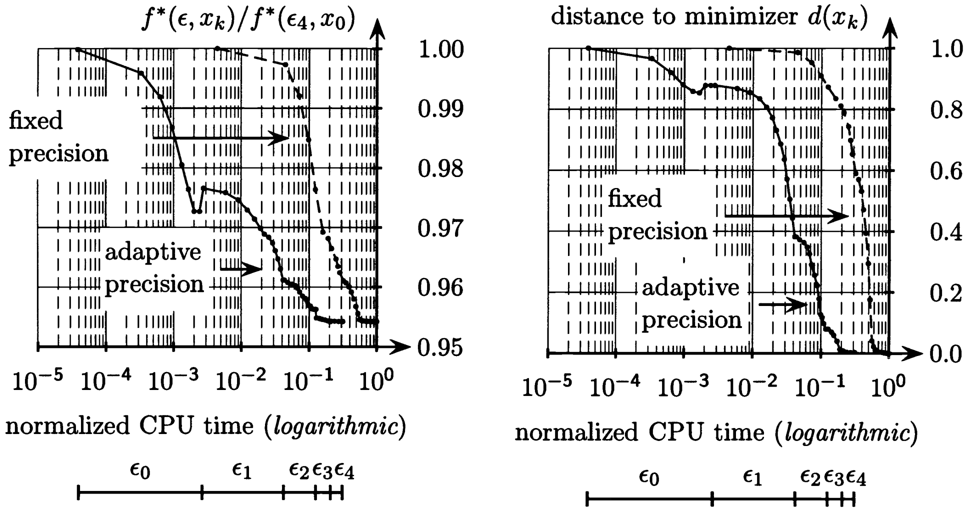


FIG. 1. *Normalized cost function value (left graph) and distance to the minimizer (right graph) as a function of the normalized CPU time in logarithmic scale. Below the graphs we show the intervals for which the precision parameter ϵ has been kept constant. For the adaptive precision optimization, we used $\zeta = 10^{-4}$ and $\alpha = 1/6$. For better display of the early iterations, the time axis is in logarithmic scale.*

The objective is to fit four parameters of a detailed air-to-water cooling coil computer simulation model in such a way that the difference between simulated and measured coil air outlet temperature is minimized for a prescribed number of measurement points. The measurement data are the air and water inlet temperature, the air humidity ratio, the air and water mass flow, and the valve position of the throttle valve in the water circuit. There are 401 measurement data, equally spaced in time.

The simulation model [35] consists of a coupled system of nonlinear equations that is solved for 373 variables using Newton iterations. The model is static and was simulated in SPARK 1.0.3 [19]. For the range of measurement data, all model equations are once continuously differentiable, and the Jacobian matrix is nonsingular in a neighborhood of the solution. Therefore, it follows from the implicit function theorem [23] that the exact solution, and hence the cost function, is once continuously differentiable.

The design parameters are the air and water side heat transfer coefficients and two parameters that define the valve characteristics. We controlled two precision parameters: the precision parameter for the Newton solver and the number of data points that were used in the data fit.

6.2.1. Exact cost function. We defined the exact cost function as follows. Let $\tau \triangleq [0,\ 1]$ denote the normalized time interval over which the measurement took place. Let $T_m \colon [0,\ 1] \to \mathbb{R}$ be the linear interpolation of the measured coil air outlet temperatures. For $x \in \mathbb{R}^n$ and $t \in \tau$, let $T_s(x,t) \in \mathbb{R}$ denote the exact solution of the system of equations that defines the coil air outlet temperature, obtained by using linearly interpolated measurement data. Then, for

$$e(x,t) \triangleq (T_m(t) - T_s(x,t))^2, \tag{22}$$

we defined the exact cost function

$$f(x) \triangleq \int_0^1 e(x,t)\,dt. \tag{23}$$

6.2.2. Approximating cost functions. The integral (23) cannot be evaluated because $T_s(x,t)$ can only be numerically approximated by an approximate solution $T_s^*(\epsilon^1, x, t) \in \mathbb{R}$ with precision parameter $\epsilon^1 \in \mathbb{R}_+$, and the integral can only be approximated by a quadrature formula. Thus, in computing the approximating cost functions, we have two sets of approximations.

We approximated $f(\cdot)$ as follows. For some $\epsilon_0 \in \mathbb{R}_+^2$, with $\epsilon_0^2 \leq 1/2$, let $\epsilon^1 \in (0, \epsilon_0^1]$ denote the precision parameter of the Newton solver, and let $\epsilon^2 \in (0, \epsilon_0^2]$ denote the time interval for the quadrature formula. For $t \in \tau$, we approximated (22) by

$$e^*(\epsilon^1, x, t) \triangleq (T_m(t) - T_s^*(\epsilon^1, x, t))^2, \tag{24a}$$

using Newton iterations. The Newton solver in the SPARK program is set up in such a way that for any compact set $\mathbf{S} \subset \mathbf{X}$, there exists an $\epsilon_{\mathbf{S}} \in \mathbb{R}_+$ and a $K'_{\mathbf{S}} \in (0, \infty)$ such that for all $x \in \mathbf{S}$, for all $t \in [0,\, 1]$, and for all $\epsilon \in \mathbb{R}_+$, with $\epsilon \leq \epsilon_{\mathbf{S}}$,

$$|e^*(\epsilon, x, t) - e(x,t)| \leq K'_{\mathbf{S}}\, \epsilon. \tag{24b}$$

We approximated the integral (23) by

$$f^*(\epsilon, x) \triangleq \sum_{i=0}^{N(\epsilon^2)-1} \frac{e^*(\epsilon^1, x, i/N(\epsilon^2)) + e^*(\epsilon^1, x, (i+1)/N(\epsilon^2))}{2\, N(\epsilon^2)}, \tag{25}$$

where $N(\epsilon^2) \triangleq \lfloor 1/\epsilon^2 \rfloor$.

It can be shown that for any compact set $\mathbf{S} \subset \mathbb{R}^n$, there exist a $K_{\mathbf{S}} \in (0, \infty)$ and an $\epsilon_{\mathbf{S}} \in \mathbb{R}_+^2$ such that

$$|f^*(\epsilon, x) - f(x)| \leq K_{\mathbf{S}}\, \|\epsilon\|, \tag{26}$$

for all $x \in \mathbf{S}$ and for all $\epsilon \in \mathbb{R}_+^2$, with $\epsilon \leq \epsilon_{\mathbf{S}}$. Therefore, $\varphi(\cdot)$ in Assumption 3.1 is $\varphi(\epsilon) = \|\epsilon\|$.

6.2.3. Optimization algorithm. Numerical experiments showed that $f(\cdot)$ has several local minima, and some local minima have a cost function value that is three times larger than the best found solution. Therefore, we used the multistart Hooke–Jeeves optimization algorithm from the GenOpt$^{(\mathrm{R})}$ 2.0.0 optimization program [28] with four randomly selected initial iterates. We controlled the precision of the approximating cost functions using the precision control algorithm from GPS Algorithm model 4.4. For comparison, we also solved the problem with the multistart Hooke–Jeeves algorithm with fixed precision $\epsilon_k = (10^{-10},\, 1/400)^T$ for all $k \in \mathbb{N}$, with $\zeta = 0$ and three step reductions.

6.2.4. Precision control. To control $\epsilon_k \in \mathbb{R}_+^2$, with $\epsilon_k \leq \epsilon_0$, as a function of the mesh size factor $\Delta_k \in \mathbb{Q}_+$, we defined $\rho\colon \mathbb{R}_+ \to \mathbb{R}_+^2$ as

$$\rho^i(\Delta) \triangleq \epsilon_{\min}^i \left(\frac{\Delta}{\Delta_{min}} \right)^{\alpha^i}, \qquad \alpha^i > 1, \qquad i \in \{1,\, 2\}, \tag{27}$$

Table 2

Initial precisions ϵ_0 used for approximating the cost functions, corresponding α, normalized computation time, and best obtained local minima for all optimizations.

ϵ_0^1	ϵ_0^2	α^1	α^2	CPU time	$f^*(\epsilon, x^*)$
10^{-10}	1/400	0	0	1	0.0236
10^{-1}	0.025	9.97	1.11	0.13	0.0242
10^{-1}	0.05	9.97	1.44	0.15	0.0225
10^{-1}	0.1	9.97	1.77	0.72	0.0216
10^{-1}	0.5	9.97	2.55	0.34	0.0217
10^{-1}	1	9.97	2.88	0.51	0.0221
10^{-2}	0.025	8.86	1.11	0.12	0.0242
10^{-2}	0.05	8.86	1.44	0.15	0.0225
10^{-2}	0.1	8.86	1.77	0.69	0.0216
10^{-2}	0.5	8.86	2.55	0.18	0.0217
10^{-2}	1	8.86	2.88	0.54	0.0221
10^{-3}	0.025	7.75	1.11	0.13	0.0242
10^{-3}	0.05	7.75	1.44	0.15	0.0225
10^{-3}	0.1	7.75	1.77	0.69	0.0216
10^{-3}	0.5	7.75	2.55	0.18	0.0217
10^{-3}	1	7.75	2.88	0.53	0.0221
average for adaptive precision				0.35	

where $\Delta_{min} \triangleq \min_{k\in\mathbb{N}}\{\Delta_k\} = 1/8$ is the smallest mesh size parameter, and $\epsilon_{min} = (10^{-10}, 1/400)^T$ is the precision parameter for the last iterations. Since $\alpha^i > 1$ for $i \in \{1, 2\}$, we have $\varphi(\rho(\Delta))/\Delta \to 0$, as $\Delta \to 0$.

To determine $\alpha > 1$, we selected different initial precisions $\epsilon_0 \in \mathbb{R}^2_+$ and then computed α by solving (27) with $\Delta = \Delta_0 = 1$ and $\rho^i(1) = \epsilon_0^i$ for $i \in \{1, 2\}$. In particular, we set

$$\alpha^i = \frac{\log\left(\epsilon_0^i/\epsilon_{min}^i\right)}{\log\left(\Delta_0/\Delta_{min}\right)} = \frac{\log\left(\epsilon_0^i/\epsilon_{min}^i\right)}{\log 8}, \qquad i \in \{1, 2\}. \tag{28}$$

Thus, for $\alpha^i > 1$, we need $\epsilon_0^i > 8\,\epsilon_{min}^i$ for $i \in \{1, 2\}$.

6.2.5. Numerical results. Table 2 shows the different settings for ϵ_0, the corresponding α, the normalized computation time, and the cost function values for the best obtained local minima. A normalized computation time of one corresponds to 45 minutes of computation time, which was the time required to solve the optimization problem with fixed precision cost function evaluations.

All optimization runs obtained a similar reduction in cost. On average, our precision control scheme reduced the computation time by a factor of three.

We will now describe how the optimizations with fixed and adaptive precision cost function evaluations, with $\zeta = 0$ and $\alpha = (9.97,\, 1.11)^T$, converged to a local minimum. Let the normalized distance to the best local minimizer of the optimization that used fixed precision cost function evaluations be defined as in (21). Figure 2 shows, for the optimizations with adaptive and fixed precision cost function evaluations, the cost function value and the normalized distance to the minimizer as a function of the computation time. For the initial iterate and the precision control parameter α used in the optimizations shown in Figure 2, both algorithms converged to the same local minimum. Below the axis we show when precision was increased (the different precisions are indicated by ϵ_m, $m \in \{0,\, 1,\, 2,\, 3\}$). For this example, the precision control algorithm set $\epsilon_0 = (10^{-1}, 0.025)^T$, $\epsilon_1 = (10^{-4}, 0.012)^T$, $\epsilon_2 = (10^{-7}, 0.0054)^T$, and $\epsilon_3 = (10^{-10}, 0.0025)^T$. After 3% of the computation time that was required to

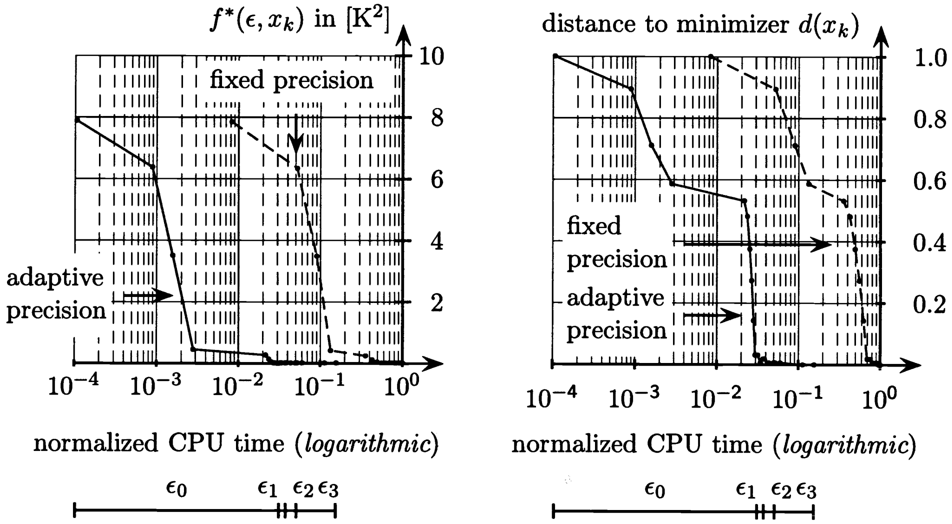


FIG. 2. *Cost function value (left graph) and distance to the minimizer (right graph) as a function of the normalized computation time in logarithmic scale. Below the graphs we show the intervals for which the precision parameter ϵ has been kept constant. For the adaptive precision optimization, we used $\zeta = 0$ and $\alpha = (9.97, 1.11)^T$. For better display of the early iterations, the time axis is in logarithmic scale.*

solve the fixed precision optimization problem, the cost function values and the iterates of the adaptive precision optimization problem were already close to the minimum.

7. Conclusion. We have extended the family of GPS algorithms to a form that converges to a stationary point of a smooth cost function that cannot be evaluated exactly, but that can be approximated by a family of possibly discontinuous functions $\{f^*(\epsilon, \cdot)\}_{\epsilon \in \mathbb{R}^q_+}$. An important feature of our algorithms is that they use low-cost, coarse precision approximations to the cost function when far from a solution, with the precision progressively increased as a solution is approached. We have shown by numerical experiments that our precision control algorithms lead to considerable time savings over using high precision approximations to the cost function in all iterations.

Acknowledgment. We thank the referees for their comments that simplified the presentation of this work.

REFERENCES

[1] M. S. Al-Homoud, *Optimum thermal design of office buildings*, Intern. J. Energy Res., 21 (1997), pp. 941–957.

[2] ASHRAE, *ANSI/ASHRAE Standard* 140-2001, Standard method of test for the evaluation of building energy analysis computer programs, 2001.

[3] C. Audet and J. E. Dennis, Jr., *Analysis of generalized pattern searches*, SIAM J. Optim., 13 (2003), pp. 889–903.

[4] A. J. Booker, J. E. Dennis, Jr., P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset, *A rigorous framework for optimization of expensive functions by surrogates*, Struct. Optim., 17 (1999), pp. 1–13.

[5] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, North–Holland, Amsterdam, 1989.

[6] P. N. Brown, A. C. Hindmarsh, and L. R. Petzold, *Using Krylov methods in the solution of large-scale differential-algebraic systems*, SIAM J. Sci. Comput., 15 (1994), pp. 1467–1488.

[7] P. N. Brown, A. C. Hindmarsh, and L. R. Petzold, *Consistent initial condition calculation for differential-algebraic systems*, SIAM J. Sci. Comput., 19 (1998), pp. 1495–1512.

[8] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Classics Appl. Math. 5, SIAM, Philadelphia, 1990.

[9] J. E. Dennis, Jr., and V. Torczon, *Direct search methods on parallel machines*, SIAM J. Optim., 1 (1991), pp. 448–474.

[10] J. E. Dennis, Jr. and V. Torczon, *Managing approximation models in optimization*, in Multidisciplinary Design Optimization: State of the Art, ICASE/NASA Langley Workshop on Multidisciplinary Optimization, N. M. Alexandrov and M. Y. Hussaini, eds., SIAM, Philadelphia, 1997, pp. 330–347.

[11] L. C. Evans, *Partial Differential Equations*, AMS, Providence, RI, 1998.

[12] E. U. Finlayson, D. K. Arasteh, C. Huizenga, M. D. Rubin, and M. S. Reilly, *WINDOW* 4.0: *Documentation of Calculation Procedures*, Tech. report LBL-33943, Lawrence Berkeley National Laboratory, Berkeley, CA, 1993.

[13] M. Fontoynont, P. Laforgue, R. Mitanchey, M. Aizlewood, J. Butt, W. Carroll, R. Hitchcock, H. Erhorn, J. De Boer, M. Dirksmöller, L. Michel, B. Paule, J. L. Scartezzini, M. Bodart, and G. Roy, *IEA SHC Task* 21*/ECBCS Annex* 29*, Daylight in Buildings, Subtask C1: Validation of Daylighting Simulation Programmes*, Tech. report T21/C1-/FRA/99-11, International Energy Agency, 1999.

[14] R. Hooke and T. A. Jeeves, *"Direct search" solution of numerical and statistical problems*, J. Assoc. Comput. Machinery, 8 (1961), pp. 212–229.

[15] J. Huang and E. Franconi, *Commercial Heating and Cooling Loads Component Analysis*, Tech. report LBL-37208, Lawrence Berkeley National Laboratory, EETD, 1999.

[16] C. T. Kelley, *Iterative Methods for Optimization*, Frontiers Appl. Math. 18, SIAM, Philadelphia, 1999.

[17] T. G. Kolda, R. M. Lewis, and V. Torczon, *Optimization by direct search: New perspectives on some classical and modern methods*, SIAM Rev., 45 (2003), pp. 385–482.

[18] P. Laforgue, *IEA SHC Task* 21*/ECBCS Annex* 29*, Daylight in Buildings, Subtask C1: Draft Report of Genelux Simulations and Other Software Results*, Tech. report T21/C1/97-10, International Energy Agency, 1997.

[19] L. Berkeley National Laboratory and A. Sowell Associates Inc., *SPARK, Reference Manual*, Berkeley, CA, 2003.

[20] R. M. Lewis and V. Torczon, *Pattern search algorithms for bound constrained minimization*, SIAM J. Optim., 9 (1999), pp. 1082–1099.

[21] R. M. Lewis and V. Torczon, *Pattern search methods for linearly constrained minimization*, SIAM J. Optim., 10 (2000), pp. 917–941.

[22] E. Polak, *Computational Methods in Optimization; A Unified Approach*, Math. in Sci. and Engrg 77, Academic Press, New York, 1971.

[23] E. Polak, *Optimization, Algorithms and Consistent Approximations*, Appl. Math. Sci. 124, Springer, New York, 1997.

[24] E. Polak and M. Wetter, *Generalized Pattern Search Algorithms with Adaptive Precision Function Evaluations*, Tech. report LBNL-52629, Lawrence Berkeley National Laboratory, Berkeley, CA, 2003.

[25] D. B. Serafini, *A Framework for Managing Models in Nonlinear Optimization of Computationally Expensive Functions*, Ph.D thesis, Rice University, Houston, TX, 1998.

[26] V. Torczon, *On the convergence of pattern search algorithms*, SIAM J. Optim., 7 (1997), pp. 1–25.

[27] V. Torczon and M. W. Trosset, *Using approximations to accelerate engineering design optimization*, in Proceedings of the 7th Annual AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA Paper 98-4800, St. Louis, MO, 1998.

[28] M. Wetter, *GenOpt, Generic Optimization Program, User Manual, Version* 2.0.0, Tech. report LBNL-54199, Lawrence Berkeley National Laboratory, Berkeley, CA, 2004.

[29] M. Wetter, *Simulation-Based Building Energy Optimization*, Ph.D thesis, University of California at Berkeley, Berkeley, CA, 2004.

[30] M. Wetter, *BuildOpt – A new building energy simulation program that is built on smooth models*, Building and Environment, 40 (2005), pp. 1085–1092.

[31] M. Wetter, E. Polak, and V. P. Carey, *BuildOpt* 1.0.1 *Validation*, Tech. report LBNL-54658, Lawrence Berkeley National Laboratory, Berkeley, CA, 2004.

[32] M. Wetter and J. Wright, *Comparison of a generalized pattern search and a genetic algorithm optimization method*, in Proceedings of the 8th Annual IBPSA Conference, G. Augenbroe and J. Hensen, eds., vol. III, Eindhoven, NL, 2003, pp. 1401–1408.

[33] M. Wetter and J. Wright, *A comparison of deterministic and probabilistic optimization algorithms for nonsmooth simulation-based optimization*, Building and Environment, 39 (2004), pp. 989–999.

[34] F. Winkelmann, *Modeling windows in EnergyPlus*, in Proceedings of the 7th Annual IBPSA Conference, Vol. 1, R. Lamberts, C. O. R. Negrão, and J. Hensen, eds., Rio de Janeiro, Brazil, 2001, pp. 457–464.

[35] P. Xu and P. Haves, *Library of Component Reference Models for Fault Detection (AHU and Chiller)*, Tech. report LBNL for the California Energy Commission, Lawrence Berkeley National Laboratory, Berkeley, CA, 2001.

SIAM J. OPTIM.
Vol. 16, No. 3, pp. 670–696

AN IMPLICIT PROGRAMMING APPROACH FOR A CLASS OF STOCHASTIC MATHEMATICAL PROGRAMS WITH COMPLEMENTARITY CONSTRAINTS[*]

HUIFU XU[†]

Abstract. In this paper, we consider a class of stochastic mathematical programs in which the complementarity constraints are subject to random factors and the objective function is the mathematical expectation of a smooth function which depends on both upper and lower level variables and random factors. We investigate the existence, uniqueness, and differentiability of the lower level equilibrium defined by the complementarity constraints using a nonsmooth version of implicit function theorem. We also study the differentiability and convexity of the objective function which implicitly depends upon the lower level equilibrium. We propose numerical methods to deal with difficulties due to the continuous distribution of the random variables and intrinsic nonsmoothness of lower level equilibrium solutions due to the complementarity constraints in order that the treated programs can be readily solved by available numerical methods for deterministic mathematical programs with complementarity constraints.

Key words. stochastic mathematical programs with complementarity constraints, lower level equilibrium, discretization, implicit smoothing

AMS subject classifications. 90C15, 90C30, 90C31, 90C33

DOI. 10.1137/040608544

1. Introduction.

Mathematical programs with equilibrium constraints (MPEC) are a class of optimization problems with two sets of variables: upper level variables and lower level variables and an equilibrium constraint defined by a parametric variational inequality or a complementarity system with lower variables being its prime variables and upper level variables being its parameters.

Over the past few years, MPEC has developed as a new area in optimization; see [23, 25] for an overview. One of the driving forces of the rapid development is that MPEC has found useful applications in many areas such as economics, management, and engineering. A particularly interesting example for MPEC is a Stackelberg–Nash leader-follower model for competition in an oligopoly market where a number of firms compete to supply homogeneous goods into a market in a noncooperative manner [24, 32]. Suppose that a distinct strategic firm (called leader), may anticipate the reaction of the remaining nonstrategic firms (called followers) to his decision and use this knowledge to select his optimal supply by minimizing the objective function. The followers' reaction to the leader's decision can be described by a Nash equilibrium which can be mathematically formulated as a variational inequality (VI). In structural optimization, the objective is often to optimize the performance of a structure, or its construction cost or weight by selecting design parameters, such as the shape of structure, or the choice of material under the constraints of the behavior of the structure, where the values of the state variables such as displacements, stresses, and contact forces are described by an equilibrium of minimal energy. The problem can be modeled as MPEC similarly [23, 25].

[*] Received by the editors May 18, 2004; accepted for publication (in revised form) June 2, 2005; published electronically January 6, 2006. This research is supported by the United Kingdom Engineering and Physical Sciences Research Council grant GR/S90850/01.

http://www.siam.org/journals/siopt/16-3/60854.html

[†] School of Mathematics, University of Southampton, Southampton SO17 1BJ, UK (h.xu@maths.soton.ac.uk).

In MPEC models, the underlying data are deterministic. However, in some important practical instances, there may be some stochastic (uncertain) factors involved in MPEC models. For instance, in a Stackelberg leader-follower equilibrium model, the leader's decision may be subject to some uncertainty in market demand. This is particularly so when the decision is made now for future output. Ignoring such an uncertainty may result in a decision being made on the basis of a particular market realization which occurs at a very low probability. De Wolf and Smeers [4] first considered this kind of stochastic leader-follower problem and applied it to model the European gas market. Xu [36] considered a more general model and investigated it with an MPEC approach.

In mechanical optimization, a structural equilibrium may be subject to the random properties of materials and randomly varying conditions such as weather and external forces [2]. The distribution of these random factors may be obtained from experience or through observation. It might be undesirable to base the optimal choice of design parameters on the expected values of the random data.

Patriksson and Wynter [26] first considered a general class of stochastic mathematical programs with equilibrium constraints (SMPEC) as follows:

$$\text{SMPEC} \quad \begin{array}{ll} \min & E(x) := \mathbb{E}[f(x, y(x, \xi(\omega)))] \\ \text{s.t.} & x \in \mathcal{X}, \end{array}$$

where $\xi : \Omega \to \mathbb{R}^l$ denotes a vector of random variables defined on a sample space Ω, $y(x, \xi(\omega))$ denotes a measurable selection from $\mathcal{S}(x, \xi(\omega))$, the set of solutions for the lower level VI problem parameterized by the upper level variable x and random vector $\xi(\omega)$; $\mathbb{E}$ denotes the expected value. They investigated the existence of an optimal solution and the directional differentiability of the objective function.

In this paper we consider a less complicated SMPEC model as follows:

$$(1) \qquad \text{SMPCC} \quad \begin{array}{ll} \min & \mathbb{E}\,[f(x, y, \xi(\omega))] \\ \text{s.t.} & x \in \mathcal{X}, \\ & 0 \le y \perp F(x, y, \xi(\omega)) \ge 0, \end{array}$$

where, by a slight abuse of notation, $f : \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}$ denotes a continuously differentiable function, $F : \mathbb{R}^m \times \mathbb{R}^n_+ \times \mathbb{R}^l \to \mathbb{R}^m$ denotes a continuously differentiable vector valued function, $\xi : \Omega \to \mathbb{R}^l$ denotes a vector of random variables defined on sample space Ω, $\mathbb{E}$ denotes the expected value, and $\mathcal{X}$ denotes a closed subset of $\mathbb{R}^m$.

In this model, we *implicitly assume* that the lower level vector of variables y uniquely solves a stochastic complementarity problem for every x and the realization of $\xi(\omega)$. The uniqueness can be guaranteed by the uniform strong monotonicity of F in y. Therefore in this model y is essentially a function of x and $\xi(\omega)$, not an independent decision vector. This is significantly different from an SMPEC model recently considered by Shapiro [31] where y is regarded as a second decision vector. The optimal upper level variable x is chosen to minimize the expected value of the objective function since the random factors are not realized at the time a decision is made. Model (1) is first investigated by Lin, Chen, and Fukushima [20] with a focus on the case when $\xi(\omega)$ is a random variable with a finite discrete distribution. It is shown that such a program can be transformed into a standard deterministic MPCC. Subsequently, a smoothing method is proposed for solving the transferred program. Lin, Chen, and Fukushima [20] also considered a variation of the model where the complementarity constraint may not necessarily have a solution for every realization of $\xi(\omega)$ and, consequently, a recourse is considered. In a revised version of the paper [21] (with a different title), Lin, Chen, and Fukushima proposed a Monte Carlo method for solving this type of recourse SMPEC model. The work is extended by Lin and Fukushima [22].

In this paper, we focus on the case when $\xi(\omega)$ is a vector of random variables with a known continuous distribution. We find that the case is more challenging in that the resulting SMPCC is no longer equivalent to a standard deterministic MPCC. To be more specific, let $\rho(t)$ denote the joint density function of $\xi(\omega)$ and $\mathcal{T}$ denote the support set of $\rho(t)$. Then (1) can be rewritten as

$$(2) \qquad \begin{array}{ll} \min & E(x) := \int_{\mathcal{T}} f(x,y,t)\rho(t)dt \\ \text{s.t.} & x \in \mathcal{X}, \\ & 0 \le y \perp F(x,y,t) \ge 0,\ t \in \mathcal{T}. \end{array}$$

Note that here t is a vector when $l > 1$ and hence the integration is multiple in general. As a result of this reformulation, we have transformed the stochastic program (1) into a deterministic program. Of course, there is a fundamental difference between a standard deterministic mathematical program with complementarity constraints and (2) since here the complementarity constraint contains a vector of parameters t and the objective function involves an integration with respect to t.

We need to investigate the properties of lower level equilibrium solution $y(x,t)$ defined by the complementarity problem in the constraint of (2) before proposing numerical methods to solve the problem. By using a nonlinear complementarity problem (NCP) function, we reformulate the complementarity constraint as an underdetermined system of nonsmooth equations and then investigate the dependence of the lower level prime variable y on the upper level vector of variables x and parametric vector t using a nonsmooth version of the implicit function theorem. We discuss Lipschitz continuity, and piecewise smoothness of $y(x,t)$ on space $\mathcal{X} \times \mathcal{T}$. The discussion is extended to upper level expected value function $E(x)$.

With the nice properties of lower level equilibrium solution and upper level objective function, we propose some numerical methods for solving (2). The methods are focused on addressing two fundamental issues in the problem. One is that since $\mathcal{T}$ is a set of positive Lebesgue measures, $y(x,t)$ is an infinite dimensional variable. This is significantly different from the case when $\mathcal{T}$ is a finite set and (2) can be easily reformulated as a standard deterministic mathematical program with a complementarity constraint. We deal with this issue by discretizing the support set $\mathcal{T}$ and replacing the integration in the objective function with a numerical integration. This kind of deterministic discretization approach is not necessarily efficient when $l > 1$ and/or $\mathcal{T}$ is large, but it is rather stable and suitable for $l = 1$ and/or a small $\mathcal{T}$. The other issue is the nonsmoothness in the constraint caused by the complementarity structure. This is similar to the deterministic MPCC case. We deal with this problem with a popular implicit NCP smoothing method as in the deterministic MPEC case.

During the revision of this paper, a new work on SMPEC by Shaprio has come up. In [31], Shapiro considered a slightly different model from (1) by choosing $y(x,\xi(\omega))$ in such a way that $f(x,y,\xi(\omega))$ is minimized for given x and $\xi(\omega)$, and in doing so he described his model as a two stage stochastic decision making problem. Moreover, he proposed a sample average approximation method to solve the problem and presented a probabilistic estimate of sample size for an ϵ-global optimizer of the original SMPEC to be a δ-optimizer of a sample average approximation program. The sample average approximation approach provides an effective alternative to the deterministic discretization approach that we will discuss in this paper in either case when: (a) $l \ge 2$, (b) the support set $\mathcal{T}$ is large, (c) the distribution of $\xi(\omega)$ is unknown.

The rest of this paper is organized as follows. In section 2, we investigate properties of lower level equilibrium solution using a nonsmooth version of implicit function theorem under the assumption that $F(x, y, t)$ is uniformly strongly monotone with respect to y. We show global Lipschitzness and piecewise smoothness for the lower level equilibrium solution $y(x, t)$. We then move on to discuss properties of upper level expected value function $E(x)$ and show that $E(x)$ is differentiable under some moderate conditions. We also use a stochastic Stackelberg–Nash–Cournot equilibrium problem as an example to discuss the differentiability and convexity of $E(x)$. In section 3, we propose a deterministic discretization approach to approximating (2) and obtain an error bound for an approximate global minimum. Note that our discussion is based on the case when $\xi(\omega)$ is a random variable ($l = 1$), but the results can be easily extended to $l > 1$ case. In section 4, we discuss an implicit smoothing approach for solving (2) and obtain error bounds for a global optimal solution of the smoothed program. Finally, in section 5, we investigate the limiting behavior of the Clarke stationary points of both discretized and smoothed programs.

2. Reformulation and characterization.

It is well known that a complementarity problem can be transformed into a system of nonsmooth equations and consequently a deterministic mathematical program with complementarity constraint can be transformed into a program with nonsmooth equality constraints. In this section, we will use the same idea to deal with the complementarity constraints in SMPCC.

2.1. Reformulation of the complementarity constraints.

Let $\phi : \mathbb{R}^2 \to \mathbb{R}$ be an NCP function [35], that is, it satisfies at least the following two properties:

$$\phi(a, b) = 0 \Longleftrightarrow a, b \geq 0 \quad \text{and} \quad ab = 0.$$

Then the complementarity constraints in (2) can be reformulated as

$$\Phi(x, y, t) := \begin{pmatrix} \phi(y_1, F_1(x, y, t)) \\ \vdots \\ \phi(y_n, F_n(x, y, t)) \end{pmatrix} = 0. \tag{3}$$

The reformulation is well known; see for instance [18, 16]. There are many NCP functions available in literature; see [35] for a review. Here we only consider the most popular two NCP functions.

One is the “min” function which is defined as

$$\phi(a, b) = \min(a, b).$$

The function is globally Lipschitz continuous and is continuously differentiable everywhere except at the line $a = b$.

The other is the Fischer–Burmeister function [9] which is defined as

$$\phi(a, b) = a + b - \sqrt{a^2 + b^2}.$$

The function is also globally Lipschitz continuous and is continuously differentiable everywhere except at $(0, 0)$.

With an NCP function, program (2) can be reformulated as

$$\begin{array}{ll} \min & \int_{\mathcal{T}} f(x, y, t)\rho(t)dt \\ \text{s.t.} & x \in \mathcal{X}, \\ & \Phi(x, y, t) = 0. \end{array} \tag{4}$$

Unfortunately, due to the presence of the integral with respect to t in the objective function, there is no available algorithm that can be directly applied to solve program (4). Our purpose here is to properly treat (4) so that it can be solved by existing algorithms for deterministic MPCC. From here on, we will focus on (4) rather than (2).

2.2. Properties of lower level equilibrium. For given x and t, the lower level equilibrium y is defined as a solution of (3). We are interested in the existence, uniqueness of such a solution and its dependence on x and t. For this purpose, we need to make some basic preparations.

Let $F : \mathbb{R}^m \times \mathbb{R}^n_+ \times \mathbb{R}^l \to \mathbb{R}^n$. $F(x, y, t)$ is said to be *uniformly strongly monotone* with respect to y if there exists a constant $\alpha > 0$ such that

$$(F(x, y', t) - F(x, y'', t))^T (y' - y'') \geq \alpha \|y' - y''\|^2 \forall y', y'' \in \mathbb{R}^n_+,\ x \in \mathcal{X},\ t \in \mathcal{T}. \tag{5}$$

Here and later the superscript T denotes the transpose of a vector and matrix.

Let $H : \mathbb{R}^j \to \mathbb{R}^l$ be a locally Lipschitz continuous function. The Clarke generalized Jacobian [3] of H at $x \in \mathbb{R}^j$ is defined as

$$\partial H(x) := \operatorname{conv} \left\{ \lim_{\substack{y \in D_H \\ y \to x}} \nabla H(y) \right\},$$

where D_H denotes the set of points near x at which H is Frechét differentiable, $\nabla H(y)$ denotes the usual Jacobian of H which is a $l \times j$ matrix, "conv" denotes the convex hull of a set. When $l = 1$ or $j = 1$, ∂H reduces to the Clarke subdifferential.

Let $D_a = diag(d_1^a, \ldots, d_n^a) \in \mathbb{R}^{n \times n}$ denote the diagonal matrix with the (i, i)th entry being d_i^a, for $i = 1, \ldots, n$. Let $D_b = diag(d_1^b, \ldots, d_n^b) \in \mathbb{R}^{n \times n}$ denote the diagonal matrix with the (i, i)th entry being d_i^b, for $i = 1, \ldots, n$. Let I denote the identity matrix in $\mathbb{R}^{n \times n}$. The function Φ defined by (3) is locally Lipschitz continuous and the Clarke generalized Jacobian of Φ with respect to y can be expressed as

$$\partial_y \Phi(x, y, t) = \left\{ (D_a, D_b) \begin{pmatrix} I \\ \nabla_y F(x, y, t) \end{pmatrix} : (d_i^a, d_i^b) \in \partial \phi(y_i, F_i(x, y, t)), i = 1, \ldots, n \right\}. \tag{6}$$

Moreover,

$$\partial_y \Phi(x, y, t) \subset \partial_y \Phi_1(x, y, t) \times \cdots \times \partial_y \Phi_n(x, y, t),$$

where

$$\partial_y \Phi_i(x, y, t) = \{ d_i^a e_i + d_i^b \nabla_y F(x, y, t) :\ (d_i^a, d_i^b) \in \partial \phi(y_i, F_i(x, y, t)) \},\ i = 1, \ldots, n.$$

The following proposition shows that under some proper conditions, the Clarke generalized Jacobian $\partial_y \Phi(x, y, t)$ is uniformly nonsingular.

PROPOSITION 2.1. *Suppose that $F(x, y, t)$ is uniformly strongly monotone with respect to y, and $\phi(a, b)$ is either the min-function or the Fischer–Burmeister function. Then there exists a constant $C > 0$ such that for all $x \in \mathcal{X}$, $y \geq 0$ and $t \in \mathcal{T}$*

$$\|(D_a + D_b \nabla_y F(x, y, t))^{-1}\| \leq C\ \forall (d_i^a, d_i^b) \in \partial \phi(y_i, F_i(x, y, t)),\ i = 1, \ldots, n.$$

Here and later on $\| \cdot \|$ denotes the 2-norm of a matrix and a vector.

We will not provide a proof since the result follows straightforwardly from [17, Proposition 3.2] where a similar conclusion is proved with ϕ being the Fischer–Burmeister function and F a P-function (of y). The case when ϕ is a min-function can be dealt with similarly.

Since y is implicitly dependent of x and t through the nonsmooth system of equations (3), the classical implicit function theorem cannot be used to study (3). We need the following generalized implicit function theorem which is established in [36] and is essentially due to Theorem 7.1.1 and the subsequent corollary of [3].

LEMMA 2.2 [36, Lemma 3.2]. *Consider an underdetermined system of nonsmooth equations*

$$H(y,z)=0,$$

where $H:\mathbb{R}^m\times\mathbb{R}^n\to\mathbb{R}^m$ *is locally Lipschitz. Let* $(\bar{y},\bar{z})\in\mathbb{R}^m\times\mathbb{R}^n$ *be such that* $H(\bar{y},\bar{z})=0$. *Suppose that* $\partial_y H(\bar{y},\bar{z})$ *is nonsingular. Then*

(i) *there exist neighborhoods* Z *of* $\bar{z}$, Y *of* $\bar{y}$, *and a locally Lipschitz function* $y:Z\to Y$ *such that* $y(\bar{z})=\bar{y}$ *and, for every* $z\in Z$, $y=y(z)$ *is the unique solution of the problem* $H(y,z)=0$, $y\in Y$*;*

(ii) *for* $z\in Z$,

$$\partial y(z)\subset\{-R^{-1}U:(R,U)\in\partial H(y(z),z),R\in\mathbb{R}^{m\times m},U\in\mathbb{R}^{m\times n}\}. \tag{7}$$

With Proposition 2.1 and Lemma 2.2, we are ready to give our main results on the lower level equilibrium solution.

THEOREM 2.3. *Let* $\Phi(x,y,t)$ *be defined as in* (3). *Suppose that* F *is uniformly strongly monotone in* y *and uniformly locally Lipschitz continuous in* x. *Then*

(i) *there exists a unique locally Lipschitz continuous function* $y(x,t)$ *such that*

$$\Phi(x,y(x,t),t)=0 \tag{8}$$

for every $x\in\mathcal{X}$ *and* $t\in\mathcal{T}$*;*

(ii) *for every* $t\in\mathcal{T}$, $y(\cdot,t)$ *is piecewise smooth in* $\mathcal{X}$*; moreover, if* $\mathcal{T}$ *is a set of positive Lebesgue measure, then* $y(\cdot,\cdot)$ *is piecewise smooth in* $\mathcal{X}\times\mathcal{T}$, *and for fixed* x, $y(x,\cdot)$ *is piecewise smooth in* $\mathcal{T}$*;*

(iii) *the Clarke generalized Jacobian of* $y(x,t)$ *with respect to* x *can be estimated as follows:*

$$\begin{aligned}\partial_x y(x,t)&\subset\{-R^{-1}U:(U,R,V)\in\partial\Phi(x,y(x,t),t),U\in\mathbb{R}^{n\times m},R\in\mathbb{R}^{n\times n},\\&\qquad V\in\mathbb{R}^{n\times l}\}\\&\subset\{-R^{-1}U:(U,R,V)\in\partial_C\Phi(x,y(x,t),t),U\in\mathbb{R}^{n\times m},R\in\mathbb{R}^{n\times n},\\&\qquad V\in\mathbb{R}^{n\times l}\},\end{aligned}$$

where $\partial_C\Phi=\partial\Phi_1\times\cdots\times\partial\Phi_n$*; moreover, if* F *is uniformly globally Lipschitz continuous in* x, *then* $y(x,t)$ *is also uniformly globally Lipschitz continuous in* x*;*

(iv) *the Clarke generalized Jacobian of* $y(x,t)$ *with respect to* t *can be estimated as follows:*

$$\begin{aligned}\partial_t y(x,t)&\subset\{-R^{-1}V:(U,R,V)\in\partial\Phi(x,y(x,t),t),U\in\mathbb{R}^{n\times m},R\in\mathbb{R}^{n\times n},\\&\qquad V\in\mathbb{R}^{n\times l}\}\\&\subset\{-R^{-1}V:(U,R,V)\in\partial_C\Phi(x,y(x,t),t),U\in\mathbb{R}^{n\times m},R\in\mathbb{R}^{n\times n},\\&\qquad V\in\mathbb{R}^{n\times l}\};\end{aligned}$$

if F is uniformly globally Lipschitz continuous in t, then $y(x,t)$ is also uniformly globally Lipschitz continuous in t.

The results are expected. In particular, similar results to parts (i) and (ii) are established by Facchinei and Pang in the context of sensitivity and stability analysis in [7]. For completeness, we attach a proof which utilizes the nonsmooth implicit function theorem in the appendix.

In practice, $y(x,t)$ represents an equilibrium at scenario t of the uncertainty. The piecewise smoothness of a component $y_i(x,t)$ at a point (x,t) implies that the value of ith lower level decision variable at the equilibrium may change at different rates at the point. In what follows, we investigate the piecewise structure of $y(x,t)$ and its differentiability.

Let

$$\mathcal{D} = \{(x,t) : x \in \mathcal{X}, t \in \mathcal{T}, y_i(x,t) + F_i(x, y(x,t), t) > 0, i = 1, \dots, n\}.$$

Obviously $y(x,t)$ is continuously differentiable on $\mathcal{D}$, and

$$\nabla_x y(x,t) = -\nabla_y \Phi(x, y(x,t), t)^{-1} \nabla_x \Phi(x, y(x,t), t)$$

and

$$\nabla_t y(x,t) = -\nabla_y \Phi(x, y(x,t), t)^{-1} \nabla_t \Phi(x, y(x,t), t) \ \forall x \in \mathcal{D}.$$

In general the structure of set $\mathcal{D}$ is complex even when x is a single variable.

2.3. Properties of the objective function. Let $y(x,t)$ be the solution of (3). We consider the objective function of the SMPCC

$$E(x) := \int_{\mathcal{T}} f(x, y(x,t), t)\rho(t)dt.$$

For simplicity of discussion, we make a blanket assumption that $E(\cdot)$ takes finite value on $\mathcal{X}$. We also assume throughout this subsection that $\mathcal{T}$ is a set of positive Lebesgue measure. We are interested in the properties of $E(x)$ such as Lipschitz continuity, differentiability, and convexity which are related to the development of numerical methods and the uniqueness of optimal solution. Note that in the general context of SMPEC, Patriksson and Wynter [26] investigated Lipschitz continuity and directional differentiability of the objective function. Our approach and results here are more specifically utilizing Clarke subdifferential.

THEOREM 2.4. *Let $\Phi(x,y,t)$ be defined as in* (3). *Suppose that F is uniformly strongly monotone in y and uniformly globally Lipschitz continuous in x. Suppose also that f is globally Lipschitz continuous with respect to (x,y), that is, for every $t \in \mathcal{T}$, there exists $L(t) > 0$ such that*

$$|f(x', y', t) - f(x'', y'', t)| \le L(t)(\|x' - x''\| + \|y' - y''\|) \ \forall x', x'' \in \mathcal{X}, \text{ and } y', y'' \in \mathbb{R}^n_+.$$

Suppose also that

$$\int_{\mathcal{T}} L(t)\rho(t)dt < \infty. \tag{9}$$

Then $E(x)$ is globally Lipschitz continuous and piecewise smooth. Moreover,

$$\partial E(x) \subset \int_{\mathcal{T}} [\nabla_x f(x, y(x,t), t) + \nabla_y f(x, y(x,t), t)\partial_x y(x,t)]\rho(t)dt. \tag{10}$$

Proof. Let $x', x'' \in \mathcal{X}$. Then

$$\begin{aligned}|E(x') - E(x'')| &\le \int_{\mathcal{T}} |f(x', y(x', t), t) - f(x'', y(x'', t), t)|\rho(t)dt \\ &\le \int_{\mathcal{T}} L(t)(\|x' - x''\| + \|y(x', t) - y(x'', t)\|)\rho(t)dt.\end{aligned}$$

By Part (iii) of Theorem 2.3, $y(x, t)$ is uniformly Lipschitz in x. Thus, there exists a constant $C > 0$ such that

$$\|y(x', t) - y(x'', t)\| \le C\|x' - x''\|.$$

Consequently,

$$|E(x') - E(x'')| \le \left[(1 + C)\int_{\mathcal{T}} L(t)\rho(t)dt\right]\|x' - x''\|.$$

The global Lipschitz continuity of $E(x)$ follows from this and (9). Given the Lipschitz continuity, we can obtain (10) by applying [3, Theorem 2.7.2] to $E(x)$. The piecewise smoothness of $E(x)$ is obvious given the piecewise smoothness of $y(x, t)$ and the smoothness of $f(x, y, t)$. □

The above theorem shows the global Lipschitzness and subdifferentiability of $E(x)$. In what follows we investigate its differentiability. Let $y(x, t)$ be the solution of (3). For $x \in \mathcal{X}$, let

$$\mathcal{T}_i(x) := \{t \in \mathcal{T} : y_i(x, t) \ge 0, F_i(x, y(x, t), t) \ge 0, y_i(x, t) + F_i(x, y(x, t), t) = 0\} \tag{11}$$

for $i = 1, \dots, n$. This is a set of points in $\mathcal{T}$ where the ith complementarity constraint degenerates for fixed x.

LEMMA 2.5. *$\mathcal{T}_i(x)$ is Lebesgue measurable.*

Proof. By Theorem 2.3, for each fixed x, $y_i(x, \cdot)$ is continuous. Thus $\mathcal{T}_i(x)$ is Lebesgue measurable. □

In general, the Lebesgue measure of $\mathcal{T}_i(x)$ in space $\mathcal{T}$ may not be zero.

ASSUMPTION 2.6. *For $i = 1, \dots, n$, the Lebesgue measure of $\mathcal{T}_i(x)$ relative to that of $\mathcal{T}$ is zero.*

In subsection 2.4, we will show that Assumption 2.6 holds in a practical instance.

PROPOSITION 2.7. *Suppose that for any x, t at which $y(\cdot, t)$ is continuously differentiable at x, the following holds:*

$$y(x', t) - y(x, t) - \nabla_x y(x, t)(x' - x) = o(\|x - x'\|). \tag{12}$$

Under Assumption 2.6,

(i) *$E(x)$ is differentiable and*

$$\nabla E(x) = \int_{\mathcal{T} \setminus \mathcal{T}(x)} [\nabla_x f(x, y(x, t), t) + \nabla_y f(x, y(x, t), t)\nabla_x y(x, t)]\rho(t)dt, \tag{13}$$

where

$$\mathcal{T}(x) = \bigcup_{i=1}^{n} \mathcal{T}_i(x);$$

(ii) *if, in addition, ∇f is uniformly Lipschitz continuous in x and y, then $E(\cdot)$ is continuously differentiable on $\mathcal{X}$.*

See a proof in the appendix.

The proposition above shows that the only possibility that $E(\cdot)$ is not differentiable at x is when the Lebesgue measure of $\mathcal{T}(x)$ is not zero.

Finally, we discuss the convexity of $E(x)$. We assume that for every $t \in T$, $f(x, y, t)$ is convex in x, y. Since the density function $\rho(t)$ is nonnegative, the integral of f with respect t gives a convex function of x, y. Unfortunately, these conditions are not adequate to ensure the convexity of $E(x)$ because $E(x)$ involves the integration of the $y(x, t)$. It is obvious that if each component function $y_i(x, t)$ is convex in x for every $t \in \mathcal{T}$, then $E(x)$ is convex. So a sufficient condition to ensure the convexity of $E(x)$ is when $y_i(x, t)$ becomes convex in x.

In general, $y_i(\cdot, t)$ is not necessarily convex. However, under some particular circumstances, we may obtain the convexity of $y_i(\cdot, t)$. We will discuss all these through an example in the next subsection.

2.4. An example. Consider a stochastic Stackelberg–Nash–Cournot equilibrium problem in an oligopoly market where $n + 1$ firms compete to supply homogeneous goods into a market in a noncooperative manner. A strategic firm, called the leader, needs to make a decision for its future output now. Assume that the leader has perfect knowledge of how other firms, called followers, react to his output and the future market distribution. Then the leader's decision problem can be formulated as

$$\max_{x \geq 0} \mathbb{E}\left[xp\left(x + \sum_{i=1}^{n} y_i(x, \xi(\omega)), \xi(\omega)\right)\right] - c_0(x), \tag{14}$$

where

$$y_i(x, \xi(\omega)) \in \arg\max_{y_i \geq 0} \left(y_i p\left(x + y_i + \sum_{k=1, k\neq i}^{n} y_k(x, \xi(\omega)), \xi(\omega)\right) - c_i(y_i)\right), i = 1, \ldots, n. \tag{15}$$

Here x denotes the leader's decision variable, y_i denotes the ith follower's decision variable, and $p(q, \xi(\omega))$ denotes the inverse market demand function which is subject to a random shock $\xi(\omega)$, that is, if the total supply to the market by all firms is q, then market price at scenario $\xi(\omega)$ is $p(q, \xi(\omega))$; $c_0(q)$ denotes the leader's cost function and $c_i(q)$, $i = 1, \ldots, n$, denotes follower i's cost function. We assume that both demand function $p(\cdot, \cdot)$ and cost functions $c_i(q)$, $i = 0, \ldots, n$, are sufficiently smooth.

In this problem, the followers are assumed to play a Nash–Cournot game after the leader's output is known and the market demand is realized and the leader needs to make a decision to maximize its expected profit before the realization of market demand. The problem was initially considered by De Wolf and Smeers [4] in the study of competition in the European gas market where the random variable $\xi(\omega)$ has only a finite discrete distribution. Recently Xu [36] extended the model to the case when the random variable $\xi(\omega)$ has a continuous distribution and reformulated (16) as a stochastic mathematical program with complementarity constraints. Assuming the leader knows the distribution of $\xi(\omega)$, Xu further reformulated the program as follows:

$$\begin{array}{ll} \max & E(x) := \int_0^u \left[xp\left(x + e^T y(x, t), t\right)\right] \rho(t)dt - c_0(x) \\ \text{s.t.} & x \geq 0, \\ & y(x, t) \text{ solves } 0 \leq y \perp F(x, y, t) \geq 0,\ t \in [0, u], \end{array} \tag{16}$$

where

$$F_i(x,y,t) = -p(x + y^T e, t) - p'_x(x + y^T e, t)y_i + c'_i(y_i),\ i = 1, \dots, n,$$

$e = (1, \dots, 1)^T$, and ρ is the density function of the random variable $\xi(\omega)$ with an interval support set $[0, u]$. Note that here and later on we use $a'(x)$ rather than $\nabla a(x)$ to denote the derivative of a real-valued function $a(x)$ with a single variable.

Obviously program (16) is an example of program (2). In what follows, we will investigate the differentiability and convexity of $E(x)$. For simplicity of discussion, we assume that the demand function is linear, that is,

$$p(q,t) = \alpha - \beta q + \gamma t,\ \text{for } t \in [0, u],$$

where $\alpha, \beta, \gamma > 0$.

PROPOSITION 2.8. *$E(x)$ is differentiable for $x > 0$.*

Proof. We use Proposition 2.7 to prove the result. Since $p(q,t) = \alpha - \beta q + \gamma t$,

$$F_i(x,y,t) = -\alpha + \beta(x + y^T e) - \gamma t + \beta y_i + c'_i(y_i),$$

and

$$\frac{dF_i(x,y,t)}{dy_j} = \begin{cases} 2\beta + c''_i(y_i), & j = i, \\ \beta, & j \neq i. \end{cases}$$

Since $\beta > 0$ and $c''_i(q) \geq 0$, it is easy to verify that $\nabla_y F(x,y,t)$ is uniformly positive definite. Therefore by Theorem 2.3, the complementarity problem

$$0 \leq y \perp F(x,y,t) \geq 0$$

has a unique solution $y(x,t)$ for every $x \geq 0$ and $t \in [0, u]$. Note that $y(x,t)$ is followers' Nash–Cournot equilibrium at demand scenario $p(\cdot, t)$. At the Nash equilibrium, the aggregate supply by followers is $y(x,t)^T e$. By Theorem 2.3, $y(x,t)^T e$ is a piecewise smooth function of x and t. Moreover, by [36, Proposition 3.4]

$$\partial_x y(x,t)e \in (-1, 0)\ \forall t \in [0, u]. \tag{17}$$

In what follows, we investigate the monotonicity of $y_i(\cdot, t)$, $i = 1, \dots, n$, for fixed $t \in [0, u]$. By the complementarity condition, we have

$$\min\left(y_i(x,t), -\alpha + \beta\left(x + y(x,t)^T e\right) - \gamma t + \beta y_i(x,t) + c'_i(y_i(x,t))\right) = 0, \text{ for } i = 1, \dots, n.$$

If $y_i(x,t) > 0$, then

$$-\alpha + \beta(x + y(x,t)^T e) - \gamma t + \beta y_i(x,t) + c'_i(y_i(x,t)) = 0. \tag{18}$$

Consequently, we have

$$\partial_x y_i(x,t) \subset -(\beta + c''_i(y_i(x,t)))^{-1}\beta(1 + \partial_x y(x,t)e).$$

Since $\beta > 0$, $c''_i(y_i(x,t)) \geq 0$, by (17), the relation above implies that every element of $\partial_x y_i(x,t)$ is negative. This shows $y_i(\cdot, t)$ is strictly decreasing at a point x where $y_i(x,t) > 0$. Furthermore, from this and the continuity of $y_i(\cdot, t)$, we can easily show that if there exists $x_i(t)$ at which $y_i(x_i(t), t) = 0$, then $y_i(x,t) = 0$ for all $x > x_i(t)$.

Let $x_i(t)$ denote the smallest x at which $y_i(x,t) = 0$ (being $+\infty$ if it does not exist). Then $y_i(\cdot,t)$ is strictly decreasing on $[0, x_i(t)]$ and $y_i(x,t) = 0$ for $x \geq x_i(t)$. The economic interpretation of $x_i(t)$ is that in certain demand scenarios, when the leader's supply reaches $x_i(t)$, follower i drops out of the market since it is no longer making a profit.

We now verify (12) for $y(x,t)$. We consider $y_i(x,t)$. Obviously, (12) is satisfied for $x > x_i(t)$. Let $x \in (0, x_i(t))$ and suppose that $y(\cdot,t)$ is differentiable at x. Then by differentiating both sides of (18) with respect to x, we obtain

$$(y_i)'_x(x,t) = -(2\beta + c''_i(y_i(x,t)))^{-1}\beta \left(1 + \sum_{\substack{j \pm 1 \\ y_j > 0}} (y_j)'_x e \right).$$

Since $c_i(\cdot)$ is assumed to be sufficiently smooth, we can show from the relation above that $y_i(x,t)$ is also twice continuously differentiable. This shows that $y(\cdot,t)$ is twice continuously differentiable at the considered point; consequently, (12) holds.

Now we prove that Assumption 2.6 holds. First let $t \in [0,u]$ be fixed. For $x > x_i(t)$, we have $y_i(x,t) = 0$. Thus

$$\partial_x F_i(x, y(x,t), t) = \beta\left(1 + \partial_x y(x,t)e\right).$$

By (17), every element of $\partial_x F_i(x, y(x,t), t)$ is positive. This shows that $F_i(x, y(x,t), t) > 0$ for $x > x_i(t)$. Therefore $x_i(t)$ is the only point satisfying the following:

$$y_i(x,t) + F_i(x, y(x,t), t) = 0. \tag{19}$$

This shows that for each t there exist at most n degenerate points.

In what follows, we investigate the behavior of $x_i(t)$ as t varies. For this purpose, we need to consider $(y_i)'_t(x,t)$. Consider

$$y_i(x,t) - \alpha + \beta(x + y(x,t)^T e) - \gamma t + \beta y_i(x,t) + c'_i(y_i(x,t)) = 0. \tag{20}$$

By differentiating both sides with respect to t, we obtain that

$$\partial_t y_i(x,t)(1 + c''_i(y_i(x,t)) + \beta) = -\beta \partial_t y(x,t)e + \gamma. \tag{21}$$

Since by part (iii) of [36, Proposition 3.4],

$$\partial_t y(x,t)e \subset \left(0, \frac{\gamma}{\beta}\right],$$

and $c''_i(y_i(x,t)) \geq 0$, we know from (21) that $y_i(x,\cdot)$ is strictly increasing at a considered t where $y_i(x,t) = x_i(t)$. This shows that $x_i(t)$ is strictly increasing as t increases, which means for any x, there exists at most one $t \in [0,u]$ such that $x_i(t) = x$. This show that $\mathcal{T}_i(x)$ defined by (11) contains at most n points. Hence Assumption 2.6 holds. By Proposition 2.7, $E(x)$ is differentiable. □

Note that this result significantly strengthens the previous result on lower level equilibrium $y(x,t)$ in [36].

We now investigate the concavity of $E(x)$. For this purpose, we look at the convexity of $y_i(\cdot,t)$. Suppose that $y_i(\cdot,t)$, $i = 1, \ldots, n$, is differentiable at x where $y_i(x,t) > 0$. Differentiating (20) with respect to x (ignoring the first term $y_i(x,t)$), we obtain

$$(\beta + c''_i(y_i(x,t))(y_i)'_x(x,t) = -\beta(1 + y'_x e). \tag{22}$$

Since $y_i(\cdot,t)$ is strictly decreasing on $(0, x_i(t))$, for each i and fixed t, there exists at most one point at which (19) is satisfied. Let $x_i(t)$ denote such a point (being $+\infty$

if it does not exist). We are interested in the details of the structure of $y_i(\cdot, t)$. For simplicity of discussion, we further assume that the marginal cost of follower i, c_i, is affine. Then $F_i(x, y, t)$ is affine in x and y, and $y_i(x, t)$ is piecewise affine.

Let

$$\mathcal{X}(t) = \{x_i(t), i = 1, \ldots, n\}.$$

For $x \in \mathcal{X} \backslash \mathcal{X}(t)$, the strict complementarity is satisfied for each i, $i = 1, \ldots, n$, which means $y_i(\cdot, t)$, $i = 1, \ldots, n$, is continuously differentiable at x. Therefore the only possibility that $y_i(\cdot, t)$ becomes nonsmooth (where it switches from one smooth piece to another) is at $x_j(t) \in \mathcal{X}(t)$, where $x_j(t) < x_i(t)$. From (22) we see that

$$\lim_{x \uparrow x_j(t)} (y_i)'_x(x, t) > \lim_{x \downarrow x_j(t)} (y_i)'_x(x, t), \tag{23}$$

which shows that $y_i(\cdot, t)$ is not differentiable at $x_j(t)$. Moreover, (23) indicates a decrease of the derivative of $y_i(\cdot, t)$ at the point $x_j(t)$. The market interpretation is that at the point where follower j drops out of the market, the unit increase on the leader's supply will more significantly reduce the remaining follower's optimal supply. Obviously, (23) indicates the local concavity of $y_i(\cdot, t)$ at point $x_j(t)$. We can summarize the main properties of $y_i(x, t)$ as follows:

- $y_i(x, t)$ is continuous and piecewise affine;
- for $x < x_i(t)$, $y_i(\cdot, t)$ is concave, and at $x_i(t)$, $y_i(\cdot, t)$ is locally convex;
- if the followers are identical, then $y_i(\cdot, t)$ is convex;
- if $X(t) = \{+\infty\}$, that is, no follower drops out as the leader increases supply up to its capacity, then $y_i(\cdot, t)$ is convex;
- $y_i(x, t)$ is not differentiable at $x_j(t) < x_i(t)$, $j = 1, \ldots, n$ and $x_i(t)$, but it is differentiable elsewhere.

Note that, at this stage we are not ready to assert whether or not $E(x)$ is concave. Observe first that $E(x)$ is a function of $Q(x, t)$, where $Q(x, t) = \sum_{i=1}^n y_i(x, t)$. If we can show that $Q(x, t)$ is convex, then it is not difficult to see that $E(x)$ is concave under usual assumptions [36]. For this purpose we look at the convexity of $Q(\cdot, t)$. Let $\mathcal{I}(x, t) = \{i : y_i(x, t) > 0\}$. Since $c_i'' = 0$, we have from (22) that

$$Q'_x(x, t) = -\frac{|\mathcal{I}(x, t)|}{1 + |\mathcal{I}(x, t)|},$$

where $|\mathcal{I}(x, t)|$ denotes the cardinality of set $|\mathcal{I}(x, t)|$. Obviously, as the value of x changes from $x_i(t) - \delta$ to $x_i(t) + \delta$, where δ is sufficiently small, $|\mathcal{I}(x, t)|$ decreases and $Q'_x(\cdot, t)$ increases. This shows the convexity of $Q(\cdot, t)$. Note that Sherali [33] obtained a similar conclusion in a deterministic Stackelberg model. Based on the discussion above, we have the following.

Proposition 2.9. *If $p(q, t)$ is affine and $c_i(q)$, $i = 1, \ldots, n$, is also affine, then $E(x)$ is concave for $x \geq 0$.*

3. Discretization methods. In this section, we discuss numerical methods for solving programs (2) through (4). The main obstacle that prevents direct application of many recently developed numerical methods for deterministic MPEC to (2) is the presence of an integral in the objective function which requires lower level variables to be solved from constraint before the objective function can be evaluated. In general, it is difficult to obtain an explicit expression of $y(x, t)$ even when F is an affine function. Our idea here is to discretize the integral and replace it with a numerical integration.

The resulting discretized program can then be solved by available numerical methods for deterministic MPECs.

To simplify the discussion, we focus on the case when $\xi(\omega)$ is a random variable. It is not difficult to see that the methods and results established in this section can be easily extended to the case when $\xi(\omega)$ consists of several random variables.

First, we deal with possible unboundedness of the support set $\mathcal{T}$. The following result is a special case of Berge's well-known stability theorem and is needed in several places in later discussion.

LEMMA 3.1. *Consider a general constrained minimization problem*

$$\begin{array}{ll} \min & p(x) \\ \text{s.t.} & x \in \mathcal{C}, \end{array}$$

where $p : \mathbb{R}^n \to \mathbb{R}$ *is continuous and* $\mathcal{C}$ *is a subset of* $\mathbb{R}^n$*, and a perturbed program*

$$\begin{array}{ll} \min & \tilde{p}(x) \\ \text{s.t.} & x \in \mathcal{C}, \end{array}$$

where $\tilde{p} : \mathbb{R}^n \to \mathbb{R}$ *is continuous and*

$$|\tilde{p}(x) - p(x)| \leq \delta \ \forall x \in \mathcal{C}.$$

Suppose that x^* *is a global minimizer of* $p(x)$ *over* $\mathcal{C}$*, and* $\tilde{x}^*$ *is a global minimizer of* $\tilde{p}(x)$ *over* $\mathcal{C}$*. Then*

$$|p(x^*) - \tilde{p}(\tilde{x}^*)| \leq \delta.$$

PROPOSITION 3.2. *Suppose that the support set* $\mathcal{T}$ *is unbounded. Let*

$$\mathcal{T}_N := \{t \in \mathcal{T} : \|t\|_\infty \leq N\},$$

where N *is a positive number and* $\|\cdot\|_\infty$ *denotes the infinity norm. Let* x_N *be a global minimizer of the following program:*

$$\begin{array}{ll} \min & E_N(x) := \int_{\mathcal{T}_N} f(x,y,t)\rho(t)dt \\ \text{s.t.} & x \in \mathcal{X}, \\ & \Phi(x,y,t) = 0. \end{array} \tag{24}$$

Then for every $\delta > 0$*, there exists* $N_0 > 0$ *such that for all* $N > N_0$*,*

$$|E(x) - E_N(x)| \leq \delta \ \forall x \in \mathcal{X},$$

and

$$|E(x^*) - E_N(x_N)| \leq \delta,$$

where x^* *denotes a global minimizer of program* (4).

Proof. The first inequality is obvious. The second inequality follows from the first one and Lemma 3.1. □

The proposition shows that we can approximate program (4) with (24). To simplify the discussion, we assume, from here on, that the support set $\mathcal{T}$ is bounded. Since $\xi(\omega)$ is a random variable, $\mathcal{T}$ is a bounded interval. We normalize it to $[0, u]$. Let $\mathcal{T}_K$ denote a set of grid points of $\mathcal{T}$ where

$$\mathcal{T}_K = \left\{ t : t_0 = 0, t_l = t_{l-1} + \frac{u}{K}, \text{ for } l = 1, \ldots, K \right\}.$$

Note that we can discretize the program (2) directly by considering

$$\begin{aligned} &\min \quad E_K(x) := \frac{u}{K}\sum_{l=1}^{K} f(x, y(x, t_l), t_l)\rho(t_l) \\ &\text{s.t.} \quad x \in \mathcal{X}, \\ &\qquad\;\; y(x, t_l) \text{ solves } 0 \le y \perp F(x, y, t_l) \ge 0,\ l = 1, \dots, K. \end{aligned} \tag{25}$$

Proposition 3.3. *Let $E_K(x)$ be defined as in* (25). *Suppose that $f(x, y(x, t), t)$ is uniformly locally Lipschitz with respect to t, that is, for every $t \in \mathcal{T}$, there exists a constant $A(t) > 0$ such that*

$$|f(x, y(x, t''), t'') - f(x, y(x, t'), t')| \le A(t)|t'' - t'| \tag{26}$$

for t', t'' near t, where $A : \mathcal{T} \to \mathbb{R}^+$ is continuous and

$$\int_0^u A(t)\rho(t)dt < \infty. \tag{27}$$

Suppose also that the density function $\rho(t)$ is differentiable on $\mathcal{T}$ and

$$\int_0^u |f(x, y(x, t), t)\rho'(t)|dt < \infty. \tag{28}$$

Then

(i) *there exists a constant $\tilde{C}$ such that*

$$|E_K(x) - E(x)| \le \frac{\tilde{C}u}{K}\ \forall x \in \mathcal{X}; \tag{29}$$

(ii)

$$|E_K(x_K) - E(x^*)| \le \frac{\tilde{C}u}{K},$$

where x_K denotes a global minimizer of $E_K(\cdot)$ and x^* denotes a global minimizer of $E(\cdot)$.

Proof. Part (i). Let

$$\Delta_l(x, t) := f(x, y(x, t_l), t_l)\rho(t_l) - f(x, y(x, t), t)\rho(t),\ \text{for } t \in (t_{l-1}, t_l).$$

By definition,

$$E_K(x) - E(x) = \frac{u}{K}\sum_{l=1}^{K}\int_{t_{l-1}}^{t_l} \Delta_l(x, t)dt.$$

Since

$$|\Delta_l(x, t)| \le \frac{u}{K}\left(\rho(t_l)\sup_{t\in[t_{l-1},t_l]} A(t) + |f(x, y(x, t), t)| \sup_{t\in[t_{l-1},t_l]} |\rho'(t)|\right),\ \text{for } t \in (t_{l-1}, t_l),$$

by (27) and (28), there exists a constant $\delta > 0$ such that for K sufficiently large

$$\begin{aligned} |E_K(x) - E(x)| &\le \frac{u}{K}\sum_{l=1}^{K}\int_{t_{l-1}}^{t_l}\left(\rho(t_l)\sup_{t\in[t_{l-1},t_l]} A(t) + |f(x, y(x, t), t)| \sup_{t\in[t_{l-1},t_l]} |\rho'(t)|\right) dt \\ &\le \left(\int_0^u A(t)\rho(t)dt + \int_0^u |f(x, y(x, t), t)||\rho'(t)|dt + \delta\right)\frac{u}{K}. \end{aligned}$$

The conclusion follows by letting $\tilde{C} = \left(\int_0^u A(t)\rho(t)dt + \int_0^u |f(x, y(x,t), t)||\rho'(t)|dt + \delta\right)$.

Part (ii) follows from Part (i) and Lemma 3.1. □

Remark 1. We make a few comments on the assumptions made in Proposition 3.3. First, the condition on the continuous differentiability of $\rho(t)$ can be relaxed to piecewise smoothness. Second, (26) and (27) holds under the following conditions:

(a) $f(x, y, t)$ is globally Lipschitz continuous in x and y and (9) holds,
(b) $f(x, y, t)$ is uniformly locally Lipschitz continuous in t with rank $A_1(t)$ where $\int_0^u A_1(t)\rho(t) < \infty$,
(c) F is uniformly monotone in y and it is uniformly globally Lipschitz continuous in t.

Note that both (a) and (c) are assumed in Theorem 2.4.

The advantage of (25) is that we can rewrite it as

$$\begin{array}{lll} & \min & E_K(x) := \dfrac{u}{K}\displaystyle\sum_{l=1}^{K} f(x, y_l, t_l)\rho(t_l) \\ (30) & \text{s.t.} & x \in \mathcal{X}, \\ & & 0 \le y_l \perp F(x, y_l, t_l) \ge 0,\ l = 1, \ldots, K, \end{array}$$

where we can treat x and $y_1, \ldots, y_K$ equally in the sense that there is no need to solve y_l from constraints in advance. Note that (30) may be viewed as a discrete stochastic mathematical program with complementarity constraints if we regard $\rho(t_l), l = 1, \ldots, K$ as probability distribution. See [26, 20] for some research on discrete stochastic mathematical program with complementarity constraints.

It is easy to see that (30) is a deterministic mathematical program with complementarity constraint, therefore a number of numerical methods proposed in [10, 11, 6, 12, 13, 15, 16, 19, 34] can be applied to this program.

The disadvantage is that to obtain a better approximation, K may be large and, consequently, a large number of variables are introduced in (30). The discretized scheme is useful only when the support set of the density function is small and/or the random variable is relatively evenly distributed over the support set.

An alternative approach to solving (2) is the sample average approximation (SAA) method. SAA is well known in stochastic programming and is effective when a problem involves several random variables. The basic idea is to generate a sample $\xi^1, \ldots, \xi^N$ with independent identical distribution as ξ and to solve the following SAA program:

$$\begin{array}{lll} (31) & \min\limits_{x \in \mathcal{X}} & \frac{1}{N}\sum_{i=1}^{N} \left[f(x, y^i, \xi^i)\right] \\ & \text{s.t.} & 0 \le y^i \perp F(x, y^i, \xi^i) \ge 0,\ i = 1, \ldots, N, \end{array}$$

to obtain an approximate solution of the original problem (1). In comparison with (30), the SAA scheme generates less evenly spread grid points which usually concentrate in areas where the density function take relatively larger values; see [31] for details.

4. An implicit smoothing method. In this section, we deal with the nonsmoothness of lower equilibrium solution $y(x, t)$. It is well known that such nonsmoothness arises from the nature of complementarity. The issue has been extensively discussed in deterministic MPCC and many methods have been proposed to deal with it. It is beyond the scope of this paper to give a comprehensive review on this topic. Here we just mention two types of methods.

One is called the smoothing NCP function method. The idea of this kind of method is to find a smooth approximation of an NCP function and replace the NCP reformulation with such a smoothed approximate NCP reformulation; see recent work in [6, 16] and references therein.

The other kind of method is called the regularization method which reformulates complementarity constraints as a system of inequalities. Such a system is often ill-posed because some constraint qualifications may not hold at any feasible point. A small perturbation at the right-hand side of the system may effectively overcome this problem; see [34] for details.

Both methods will generate a smooth approximation of the solution of a complementarity problem. Here we will use the smoothing NCP function methods.

Recall that the smoothing of an NCP function $\phi(a,b)$ is a function $\psi(a,b,c)$ satisfying the following:

(A1) $\psi(a,b,0)=\phi(a,b)$;

(A2) $\psi(a,b,c)$ is Lipschitz continuous and is continuously differentiable everywhere except at $c=0$;

(A3) (Strong Jacobian Consistency [1]) for $(a,b)\in\mathbb{R}^2$,

$$\partial_{a,b}\psi(a,b,0)=\partial\phi(a,b).$$

A smoothing function of $\min(a,b)$ is

$$\psi(a,b,c)=-\frac{1}{2}\left(\sqrt{(a-b)^2+c^2}-(a+b)\right)$$

and a smoothing function of the Fischer–Burmeister function is

$$\psi(a,b,c)=a+b-\sqrt{a^2+b^2+c^2};$$

see, for instance, [18].

Let $H:\mathbb{R}^n\to\mathbb{R}^m$ be a locally Lipschitz function. The ϵ-generalized Jacobian is defined as

$$\partial^\epsilon H(x)=\operatorname{conv}\bigcup_{y\in B(x,\epsilon)}\partial H(y),$$

where $B(x,\epsilon)$ denotes the unit ball in $\mathbb{R}^n$ centered at x with radius ϵ. The notion was introduced in [37] as a generalization of ϵ-subdifferential [30] for the purpose of the approximation of the Clarke generalized Jacobian in solving nonsmooth equations.

LEMMA 4.1. *Let $\psi(a,b,c)$ be a smoothing of an NCP function $\phi(a,b)$ satisfying properties* A1–A3. *Then there exists continuous function $\epsilon:\mathbb{R}_+\to\mathbb{R}_+$ such that a*

$$\nabla_{a,b}\psi(a,b,c)\in\partial^{\epsilon(c)}\phi(a,b) \tag{32}$$

for c close to 0, *where*

$$\lim_{c\to 0}\epsilon(c)=0.$$

Proof. The conclusion follows from the upper semicontinuity of the Clarke generalized Jacobian and the strong Jacobian consistency of ψ. ∎

Let ψ be either the smoothing of the min-function or the smoothing of the Fischer–Burmeister function. Let

$$\Psi(x,y,t,\mu) := \begin{pmatrix} \psi(y_1, F_1(x,y,t),\mu) \\ \vdots \\ \psi(y_n, F_n(x,y,t),\mu) \end{pmatrix}. \tag{33}$$

Then

$$\Psi(x,y,t,0) = \begin{pmatrix} \psi(y_1, F_1(x,y,t),0) \\ \vdots \\ \psi(y_n, F_n(x,y,t),0) \end{pmatrix} = \begin{pmatrix} \phi(y_1, F_1(x,y,t)) \\ \vdots \\ \phi(y_n, F_n(x,y,t)) \end{pmatrix}.$$

We consider the following program which is a smoothing of (4):

$$\begin{array}{ll} \min & \int_{\mathcal{T}} f(x,y,t)\rho(t)dt \\ \text{s.t.} & x \in \mathcal{X}, \\ & \Psi(x,y,t,\mu) = 0. \end{array} \tag{34}$$

We regard the approach as implicit smoothing in the sense that by replacing Φ with Ψ, we achieve the smoothing of the implicit function $y(x,t)$. Note that Lin, Chen, and Fukushima [20] considered a similar approach for a class of discrete stochastic mathematical programs with complementarity constraint. Here we rely more heavily on the implicit function approach in dealing with (34).

Recall that a vector-valued function $g : \mathbb{R}^n \to \mathbb{R}^m$ is called *calm* at point $\bar{x}$ if there exists a $\kappa > 0$ such that

$$\|g(x) - g(\bar{x})\| \leq \kappa \|x - \bar{x}\|$$

for all x near $\bar{x}$; see page 351 in [28].

PROPOSITION 4.2. *Suppose that $\mathcal{T}$ is a set of positive Lebesgue measures. Suppose also that F is uniformly strongly monotone with respect to y and it is uniformly locally Lipschitz continuous with respect to x. Suppose that ϕ is either min-function or the Fischer–Burmeister function. Then*

(i) *$\partial_y \Psi(x,y,t,\mu)$ is uniformly nonsingular and there exists $\mu_0 > 0$ such that the system of equations*

$$\Psi(x,y,t,\mu) = 0$$

defines a unique implicit function $\tilde{y}(x,t,\mu)$ which satisfies

$$\Psi(x,\tilde{y}(x,t,\mu),t,\mu) = 0, \text{ for } x \in \mathcal{X}, t \in \mathcal{T},\ |\mu| \in (0,\mu_0];$$

(ii) *$\tilde{y}(x,t,\mu)$ is continuously differentiable with respect to (x,t,μ) on $\mathcal{X} \times \mathcal{T} \times [-\mu_0,\mu_0]\backslash\{0\}$; it is locally Lipschitz continuous with respect to x and t if F is so;*

(iii) *$\tilde{y}(x,t,\mu)$ is uniformly calm in μ at 0, that is, there exists $\hat{C} > 0$ such that*

$$\|\tilde{y}(x,t,\mu) - \tilde{y}(x,t,0)\| \leq \hat{C}|\mu|, \ \textit{for}\ |\mu| \in [0,\mu_0]; \tag{35}$$

(iv) *there exists a real valued function $\epsilon : \mathbb{R}_+ \to \mathbb{R}_+$ such that*

$$\nabla_x \tilde{y}(x,t,\mu) \in \partial_x^{\epsilon(\mu)} y(x,t),\ |\mu| \in [0,\mu_0], \tag{36}$$

where $\lim_{\mu\to 0} \epsilon(\mu) = 0$.

From part (iv), we see that any accumulation matrix of $\nabla_x \tilde{y}(x,t,\mu)$ as $\mu \to 0$ is contained in the Clarke generalized Jacobian $\partial y(x,t)$.

Proof of Proposition 4.2. Part (i). The uniform nonsingularity of $\partial_y \Psi(x,y,t,\mu)$ follows essentially from [17, Proposition 3.2]. The existence and uniqueness of $\tilde{y}(x,t,\mu)$ for some $\mu_0 > 0$ follows from part (i) of Theorem 2.3.

Part (ii). The continuous differentiability of $\tilde{y}(x,t,\mu)$ follows from Part (i) and the classical implicit function theorem. We can prove the local Lipschitz continuity by applying Lemma 2.2.

Since $\partial_y \Psi(x,y,t,\mu)$ is uniformly nonsingular, $\partial_{(x,t,\mu)}\Psi(x,y,t,\mu)$ is bounded in a closed neighborhood of $(x,t,\mu) \in \mathcal{X} \times \mathcal{T} \times [-\mu_0,\mu_0]\backslash\{0\}$. It is evident by part (ii) Lemma 2.2 that $\partial\tilde{y}(x,t,\mu)$ is bounded, hence $\tilde{y}(x,t,\mu)$ is locally Lipschitz continuous.

Part (iii). Since $\nabla\tilde{y}(x,t,\mu)$ is continuous for $\mu \neq 0$,

$$\tilde{y}(x,t,\mu) - \tilde{y}(x,t,0) = \int_0^1 \tilde{y}'_\mu(x,t,\mu\nu)\mu d\nu.$$

Thus

$$\begin{aligned}\|\tilde{y}(x,t,\mu) - \tilde{y}(x,t,0)\| &\le |\mu| \int_0^1 \|\tilde{y}'_\mu(x,t,\mu\nu)\| d\nu \\ &= |\mu| \int_0^1 \|\nabla_y \Psi(x,\tilde{y}(x,t,\mu),t,\mu\nu)^{-1}\Psi'_\mu(x,\tilde{y}(x,t,\mu\nu),t,\mu)\| d\nu \\ &\le \hat{C}|\mu|,\end{aligned}$$

where $\hat{C}$ is a constant.

Part (iv). By definition,

$$\Psi(x,\tilde{y}(x,t,\mu),t,\mu) = 0.$$

By the classical implicit function theorem,

$$\nabla_x\tilde{y}(x,t,\mu) = -\nabla_y\Psi(x,\tilde{y}(x,t,\mu),t,\mu)^{-1}\nabla_x\Psi(x,\tilde{y}(x,t,\mu),t,\mu).$$

Consider $\nabla\Psi(x,\tilde{y}(x,t,\mu),t,\mu)$. Since $\tilde{y}(x,t,\cdot)$ is uniformly calm at $\mu = 0$ as we proved in part (iii), by Lemma 4.1, we know that there exists $\epsilon_1(\mu) > 0$ such that

$$\nabla\Psi(x,\tilde{y}(x,t,\mu),t,\mu) \in \partial^{\epsilon_1(\mu)}\Psi(x,y,t).$$

By the definition of ϵ-generalized Jacobian and part (ii) of Lemma 2.2, there exists $\epsilon : \mathbb{R} \to \mathbb{R}_+$, $\epsilon(\mu) \to 0$ as $\mu \to 0$, such that (36) holds. This completes the proof. □

COROLLARY 4.3. *Suppose that $\mathcal{T}$ is a set of positive Lebesgue measure. Suppose also that F is uniformly strongly monotone with respect to y and is uniformly Lipschitz continuous with respect to x. Let*

$$\tilde{E}(x,\mu) = \int_{\mathcal{T}} f(x,\tilde{y}(x,t,\mu),t)\rho(t)dt. \tag{37}$$

Then there exists $\mu_0 > 0$ such that

(i) *$\tilde{E}(x,\mu)$ is uniformly calm with respect to μ at 0, that is, there exists $\hat{C} > 0$ such that*

$$\|\tilde{E}(x,\mu) - E(x)\| \le \hat{C}|\mu|, \ \textit{for } |\mu| \in [0,\mu_0]; \tag{38}$$

(ii) *there exists a real valued function* $\epsilon : \mathbb{R} \to \mathbb{R}_+$ *such that*

$$\nabla_x \tilde{E}(x,\mu) \in \partial^{\epsilon(\mu)} E(x),\ |\mu| \in [0,\mu_0], \tag{39}$$

where $\lim_{\mu\to 0} \epsilon(\mu) = 0$.

Proof. Part (i). By the assumption on f and part (iii) of Proposition 4.2,

$$\begin{aligned}|f(x,\tilde{y}(x,t,\mu),t) - f(x,\tilde{y}(x,t,0),t)| &\le L(t)\|\tilde{y}(x,t,\mu) - \tilde{y}(x,t,0)\|\\ &\le \hat{C}L(t)|\mu|.\end{aligned}$$

Hence

$$|\tilde{E}(x,\mu) - E(x)| \le \hat{C}|\mu| \int_{\mathcal{T}} L(t)\rho(t)dt.$$

Part (ii) follows from Part (iv) of Proposition 4.2. □

Corollary 4.3 shows that $\tilde{E}(x,\mu)$ approximates $E(x)$ uniformly. It also implies that any accumulation vector of $\nabla_x \tilde{E}(x,\mu)$ as $\mu \to 0$ is an element of $\partial E(x)$. Therefore $\nabla_x \tilde{E}(x,\mu)$ can be used to calculate an element of the Clarke subdifferential of $E(x)$.

THEOREM 4.4. *Let* $\{\mu_k\}$ *be a strictly decreasing sequence such that* $\mu_k \downarrow 0$ *as* $k \to \infty$. *Let* $\{(x_k, \tilde{y}(x_k,\cdot,\mu_k))\}$ *be a sequence of solutions of* (34). *Under the conditions of Proposition* 4.2*,*

(i) *any accumulation point of* $\{(x_k, \tilde{y}(x_k,\cdot,\mu_k))\}$ *is a solution of* (4)*;*

(ii) *there exists a constant* $C > 0$ *such that*

$$|\tilde{E}(x_k,\mu_k) - E^*| \le C\mu_k,$$

where E^* *denotes the minimum of* (2)*;*

(iii) *if* x *is an accumulation point of* $\{x_k\}$ *and* M *is an accumulation matrix of* $\{\nabla_x \tilde{y}(x_k,t,\mu_k)\}$ *and* ξ *is an accumulation vector of* $\{\nabla_x \tilde{E}(x_k,u_k)\}$*, then* $M \in \partial_x y(x,t)$ *and* $\xi \in \partial E(x)$.

Proof. Parts (i) and (ii) follow from part (i) of Corollary 4.3 and Lemma 3.1. Part (iii) follows from part (iv) of Proposition 4.2 and part (ii) of Corollary 4.3. □

Theorem 4.4 ensures a smooth approximation of (2) by (34). There exist at least two ways to solve the latter. One is to consider

$$\begin{array}{ll}\min & \tilde{E}(x,\mu)\\ \text{s.t.} & x \in \mathcal{X}\end{array} \tag{40}$$

and solve it with a smooth nonlinear programming method which depends only on the function and gradient values of $\tilde{E}(x,\mu)$. In this way, we only treat x as a variable. The other is to discretize the smoothed program (34). In what follows, we consider the latter.

We consider the discretized smoothed program

$$\begin{array}{ll}\min & \tilde{E}_K(x,\mu) := \dfrac{u}{K}\displaystyle\sum_{l=0}^{K} f(x,\tilde{y}(x,t_l,\mu),t_l)\rho(t_l)\\ \text{s.t.} & x \in \mathcal{X},\\ & \tilde{y}(x,t_l,\mu) \text{ solves } \Psi(x,y,t_l,\mu) = 0,\ l = 1,\ldots,K.\end{array} \tag{41}$$

PROPOSITION 4.5. *Let* $\tilde{E}_K(x,\mu)$ *be defined as in* (41)*. Suppose that* $f(x,\tilde{y}(x,t,\mu),t)$ *is uniformly locally Lipschitz with respect to* t*, that is, for every* $t \in \mathcal{T}$*, there exists a positive constant* $\tilde{A}(t)$ *(depends on* t*) such that*

$$|f(x,\tilde{y}(x,t'',\mu),t'') - f(x,\tilde{y}(x,t',\mu),t')| \le \tilde{A}(t)|t'' - t'| \tag{42}$$

for all t', t'' near t, where $\tilde{A} : \mathcal{T} \to \mathbb{R}^+$ is continuous. Moreover, there exists $\mu_0 > 0$, such that for $\mu \in [0, \mu_0]$,

$$\int_0^u \tilde{A}(t)\rho(t)dt < \infty. \tag{43}$$

Suppose also that the density function $\rho(t)$ is differentiable on $\mathcal{T}$ and

$$\int_0^u |f(x, \tilde{y}(x, t, \mu), t)\rho'(t)|dt < \infty, \tag{44}$$

where $\rho'(t)$ denotes the derivative of $\rho(t)$. Then

(i) *there exists a constant $\tilde{C}$ such that*

$$|\tilde{E}_K(x, \mu) - \tilde{E}(x, \mu)| \le \frac{\tilde{C}u}{K} \ \forall x \in \mathcal{X}; \tag{45}$$

(ii)

$$|\tilde{E}_K(x_K^\mu, \mu) - \tilde{E}(x_\mu, \mu)| \le \frac{\tilde{C}u}{K}, \tag{46}$$

where x_K^μ denotes a global minimizer of $\tilde{E}_K(x, \mu)$ and x_μ denotes a global minimizer of $\tilde{E}(x, \mu)$.

We omit the proof as it is similar to that of Proposition 3.3.

It might be helpful to make a few comments about conditions (42)–(44). It is not difficult to verify that (42)–(44) hold under the conditions (a)–(c) in Remark 1 and (28). Indeed, under the condition (c), both $\|\nabla_y \Psi(x, \tilde{y}(x, t, \mu), t, \mu)^{-1}\|$ and $\|\nabla_t \Psi(x, \tilde{y}(x, t, \mu), t, \mu)\|$ are uniformly bounded for μ sufficiently small. Since

$$\nabla_t \tilde{y}(x, t, \mu) = -\nabla_y \Psi(x, \tilde{y}(x, t, \mu), t, \mu)^{-1} \nabla_t \Psi(x, \tilde{y}(x, t, \mu), t, \mu),$$

then $\nabla_t \tilde{y}(x, t, \mu)$ is uniformly bounded which implies that $\tilde{y}(x, \cdot, \mu)$ is uniformly globally Lipschitz continuous in set $\mathcal{T}$. Combining this with conditions (a) and (c) in the remark, we can prove (42) and (43). Finally, (44) follows the uniform calmness of $\tilde{y}(x, t, \cdot)$ at $\mu = 0$ and (28).

Based on Proposition 4.5, we can solve the smoothed program (34) by solving the discretized program (41). Since the latter is a typical deterministic smooth mathematical program with complementarity constraint, it can be solved by a number of existing algorithms such as those proposed by Jiang and Ralph [16]. Note that if we choose μ to be a proportion of T/K, we can easily obtain an estimation of $|\tilde{E}_K(x_K^\mu, \mu) - E^*|$ using Theorem 4.4 and Proposition 4.5.

Note also that in order to reduce the error bound in (46), we need to increase the number of grid points K, which means increasing the number of lower level variables and equality constraints in (41). This may increase problem size and reduce the computational efficiency. In contrast, the first way may avoid the increase of problem size although it also requires discretization of $\mathcal{T}$ to compute numerically the function and gradient values of $\tilde{E}(x, \mu)$.

5. Optimality conditions. In the preceding sections, we outlined three ways to solve (2): (a) solving discretized program (25) and increase K if necessary; (b) solving smoothed program (40); (c) solving smoothed discretized program (41). The error bounds obtained in Proposition 3.3, Theorem 4.4, and Proposition 4.5 are based on *global* minimizers of the relevant programs although these results would also apply to local minimizers after localizing the set $\mathcal{C}$ in Lemma 3.1. In practice, finding a global minimizer might be difficult and in some cases we might just find a stationary point. Consequently, we want to know whether or not an accumulation point of the sequence of stationary points is a stationary point of program (4). For this purpose, we need to investigate the optimality conditions of program (4), the discretized program (25), the smoothed program (40), and smoothed discretized program (41) and their relationship.

Program (4). The generalized Karush–Kuhn–Tucker (KKT) condition [14] of the program (4) is

$$(47)\quad 0 \in \int_{\mathcal{T}} [\nabla_x f(x, y(x,t), t)^T + \partial_x y(x,t)^T \nabla_y f(x, y(x,t), t)^T]\rho(t)dt + \mathcal{N}_{\mathcal{X}}(x),$$

where $\mathcal{N}_{\mathcal{X}}(x)$ denotes the normal cone of $\mathcal{X}$ at x, that is,

$$\mathcal{N}_{\mathcal{X}}(x) = \{d : d^T(x' - x) \le 0 \ \forall x' \in \mathcal{X}\}.$$

A point x^* satisfying the KKT condition is known as a Clarke stationary point. Using the estimation of $\partial_x y(x,t)$ in Part (iii) of Theorem 2.3, we obtain

$$(48)\quad 0 \in \int_{\mathcal{T}} [\nabla_x f(x, y(x,t), t)^T + \Im_{\Phi}(x,t)^T \nabla_y f(x, y(x,t), t)^T]\rho(t)dt + \mathcal{N}_{\mathcal{X}}(x)$$

where

$$\Im_{\Phi}(x,t) = \{-R^{-1}U : (U, R, V) \in \partial\Phi(x, y(x,t), t), U \in \mathbb{R}^{n\times m}, R \in \mathbb{R}^{n\times n}, V \in \mathbb{R}^{n\times l}\}.$$

Discretized program. Consider the discretized program (30) which is equivalent to

$$(49)\quad \begin{array}{ll} \min & E_K(x) := \dfrac{u}{K}\displaystyle\sum_{l=1}^{K} f(x, y_l, t_l)\rho(t_l) \\ \text{s.t.} & x \in \mathcal{X}, \\ & \Phi(x, y_l, t_l) = 0,\ l = 1, \dots, K. \end{array}$$

Note that (49) can be viewed as a discretized program of (4). The generalized KKT condition of (49) is

$$\left\{ \begin{array}{l} 0 \in \dfrac{u}{K}\displaystyle\sum_{l=1}^{K} \nabla_x f(x, y_l, t_l)^T \rho(t_l) + \sum_{l=1}^{K} \partial_x \Phi(x, y_l, t_l)^T \lambda_l + \mathcal{N}_{\mathcal{X}}(x), \\ 0 \in \dfrac{u}{K}\nabla_{y_l} f(x, y_l, t_l)^T \rho(t_l) + \partial_{y_l}\Phi(x, y_l, t_l)^T \lambda_l, l = 1, \dots, K, \end{array}\right.$$

which can be equivalently written as

$$0 \in \frac{u}{K}\sum_{l=1}^{K} \left[\nabla_x f(x, y_l, t_l)^T - \partial_x \Phi(x, y_l, t_l)\partial_{y_l}\Phi(x, y_l, t_l)^{-T}\nabla_{y_l} f(x, y_l, t_l)^T\right]\rho(t_l) + \mathcal{N}_{\mathcal{X}}(x).$$

Solving $y_l, l = 1, \dots, K$, from $\Phi(x, y_l, t_l) = 0$ and writing $\partial_{y_l}\Phi(x, y_l, t_l)$ and $\nabla_{y_l} f(x, y_l, t_l)$ as $\partial_y \Phi(x, y(x, t_l), t_l)$ and $\nabla_y f(x, y(x, t_l), t_l)$, we can rewrite the KKT condition as

$$\begin{aligned} 0 \in \frac{u}{K}\sum_{l=1}^{K} &[\nabla_x f(x, y(x, t_l), t_l)^T \\ &-\partial_x \Phi(x, y(x, t_l), t_l)^T \partial_y \Phi(x, y(x, t_l), t_l)^{-T} \nabla_y f(x, y(x, t_l), t_l)^T]\rho(t_l) + \mathcal{N}_{\mathcal{X}}(x). \end{aligned} \tag{50}$$

Naturally, we would like to link (50) to the following condition:

$$\begin{aligned} 0 \in \int_0^u &[\nabla_x f(x, y(x, t), t)^T \\ &-\partial_x \Phi(x, y(x, t), t)^T \partial_y \Phi(x, y(x, t), t)^{-T} \nabla_y f(x, y(x, t), t)^T]\rho(t)dt + \mathcal{N}_{\mathcal{X}}(x) \end{aligned} \tag{51}$$

and view (51) as a limit of (50). It is not difficult to prove that when $\mathcal{T}(x)$ is a finite set and

$$\int_0^u \left| d(\nabla_x \Phi(x, y(x, t), t)^T \nabla_y \Phi(x, y(x, t), t)^{-T} \nabla_y f(x, y(x, t), t)^T]\rho(t)/dt \right| dt < \infty,$$

any accumulation point of sequence $\{x_K\}$, where x_K satisfies (50), is a KKT point satisfying (51). It seems, however, difficult to extend the conclusion to the general case. Observe also that the KKT condition (48) is sharper than that of (51), which means even if an accumulation point of sequence $\{x_K\}$ satisfies (51), it is not necessarily a KKT point of (48).

Smoothed program. Consider the smoothed program (40). Let x_μ be a KKT point of the program. We are interested in the convergence of sequence $\{x_\mu\}$ as $\mu \to 0$.

PROPOSITION 5.1. *Suppose that x_μ is a KKT point of* (34) *and x^* is an accumulation point of sequence $\{x_\mu\}$ as $\mu \to 0$. Then x^* is a KKT point of* (4).

Proof. By definition

$$0 \in \nabla_x \tilde{E}(x_\mu, \mu)^T + \mathcal{N}_{\mathcal{X}}(x_\mu).$$

By upper semicontinuity of the normal cone,

$$\overline{\lim_{\mu \to 0}}\ \mathcal{N}_{\mathcal{X}}(x_\mu) \subset \mathcal{N}_{\mathcal{X}}(x^*),$$

where $\overline{\lim}$ denotes the outer limit.

Note that if we treat μ as a variable, then $\tilde{E}(\cdot, \cdot)$ is continuously differentiable at any point (x, μ), where $\mu > 0$ and is locally Lipschitz continuous near point $(x, 0)$. For a set valued mapping $\mathcal{A} : \mathbb{R}^m \times \mathbb{R} \to 2^{\mathbb{R}^{(m+1)\times m}}$, we use $\Pi_x \mathcal{A}(x, \epsilon)$ to denote the set of all $m \times m$ matrices U such that, for some vector $V \in \mathbb{R}^m$, the $(m+1) \times m$ matrix $[U^T, V]^T$ belongs to $\mathcal{A}(x, \epsilon)$. Using this notation, we have

$$\nabla_x \tilde{E}(x_\mu, \mu) = \Pi_x \nabla \tilde{E}(x_\mu, \mu).$$

By the definition of the Clarke generalized Jacobian

$$\overline{\lim_{\mu \to 0}}\ \nabla \tilde{E}(x_\mu, \mu) \subset \partial \tilde{E}(x^*, 0).$$

Hence

$$\overline{\lim_{\mu \to 0}}\ \nabla_x \tilde{E}(x_\mu, \mu) = \overline{\lim_{\mu \to 0}}\ \Pi_x \nabla \tilde{E}(x_\mu, \mu) \subset \Pi_x \partial \tilde{E}(x^*, 0) = \partial \tilde{E}(x^*).$$

The last equality is due to the Jacobian consistency. This shows

$$0 \in \partial E(x^*)^T + \mathcal{N}_{\mathcal{X}}(x^*).$$

The proof is complete. □

Discretized smoothed program. Finally, we consider the discretized smoothed program (41) with $\mathcal{X} \subset \mathbb{R}^n$

$$\begin{array}{lll} & \min & E_K(x,\mu) := \dfrac{u}{K}\displaystyle\sum_{l=0}^{K} f(x, y_l, t_l)\rho(t_l) \\ (52) & \text{s.t.} & x \in \mathcal{X}, \\ & & \Psi(x, y_l, t_l, \mu) = 0,\ l = 1, \dots, K. \end{array}$$

The KKT condition of this program is

$$(53) \quad \begin{cases} 0 & \in \dfrac{u}{K}\displaystyle\sum_{l=1}^{K} \nabla_x f(x, y_l, t_l)^T \rho(t_l) + \sum_{l=1}^{K} \nabla_x \Phi(x, y_l, t_l, \mu)^T \lambda_l + \mathcal{N}_{\mathcal{X}}(x), \\ 0 & = \dfrac{u}{K} \nabla_{y_l} f(x, y_l, t_l)^T \rho(t_l) + \nabla_{y_l} \Phi(x, y_l, t_l, \mu)^T \lambda_l, l = 1, \dots, K, \end{cases}$$

equivalently,

$$0 \in \frac{u}{K} \sum_{l=1}^{K} \left[\nabla_x f(x, y_l, t_l)^T - \nabla_x \Phi(x, y_l, t_l, \mu)^T \nabla_{y_l} \Phi(x, y_l, t_l, \mu)^{-T} \nabla_{y_l} f(x, y_l, t_l, \mu)^T \right] \\ \rho(t_l) + \mathcal{N}_{\mathcal{X}}(x)$$

Since y_l can be solved from $\Phi(x, y_l, t_l, \mu) = 0$, we can express y_l as $\tilde{y}(x, t_l, \mu)$. Thus we have

$$\begin{aligned} 0 \in \frac{u}{K} \sum_{l=1}^{K} & [\nabla_x f(x, \tilde{y}(x, t_l, \mu), t_l)^T \\ & - \nabla_x \Phi(x, \tilde{y}(x, t_l, \mu), t_l, \mu)^T \nabla_y \Phi(x, \tilde{y}(x, t_l, \mu), t_l, \mu)^{-T} \nabla_y f(x, \tilde{y}(x, t_l, \mu), t_l, \mu)^T] \rho(t_l) \\ & + \mathcal{N}_{\mathcal{X}}(x). \end{aligned}$$

Driving K to ∞, we obtain

$$\begin{aligned} 0 \in \int_0^u & [\nabla_x f(x, \tilde{y}(x, t, \mu), t)^T \\ & - \nabla_x \Phi(x, \tilde{y}(x, t, \mu), t, \mu)^T \nabla_y \Phi(x, \tilde{y}(x, t, \mu), t, \mu)^{-T} \nabla_y f(x, \tilde{y}(x, t, \mu), t, \mu)^T] \rho(t) dt \\ & + \mathcal{N}_{\mathcal{X}}(x). \end{aligned}$$

Driving μ to 0 and considering the strong Jacobian consistency of ψ, we obtain

$$0 \in \int_0^u \left[\nabla_x f(x, y(x, t), t)^T + \Im_\Phi(x, y(x, t), t)^T \nabla_y f(x, y(x, t), t)^T \right] \rho(t) dt + \mathcal{N}_{\mathcal{X}}(x).$$

From the discussion above, we can conclude that, from a KKT perspective, numerical methods based on the smoothed program (40) and the discretized smoothed program (41) may be more preferable.

Appendix.

Proof of Theorem 2.3. Part (i). Since F is uniformly strongly monotone in y, it is well known that the complementarity problem in (2) has a unique solution for all $t \in \mathcal{T}$ and $x \in \mathcal{X}$; see, for instance, [8, Corollary 3.2]. Thus, (3) has a unique solution for each $x \in \mathcal{X}$ and $t \in [0,T]$. Here we use Lemma 2.2. Under the assumption on F, the Clarke generalized Jacobian $\partial_y \Phi(x,y,t)$ is uniformly nonsingular. By Lemma 2.2, for $(\bar{x},\bar{y},\bar{u}) \in \mathcal{X} \times \mathbb{R}^n_+ \times \mathcal{T}$, there exists a Lipschitz continuous function $y(x,t)$ such that $y(\bar{x},\bar{t}) = \bar{y}$, and (8) holds for (x,t) in a neighborhood of $(\bar{x},\bar{t})$. The uniform monotonicity of F with respect to y allows the implicit function to be extended to the whole area $\mathcal{X} \times \mathcal{T}$.

Part (ii). Since Φ is piecewise smooth, by [29, Lemma 4.11], the implicit function $y(x,t)$ which is defined in part (ii) is piecewise smooth with respect to either x for fixed t or t for fixed x or both.

Part (iii). By part (ii) of Lemma 2.2,

$$\partial_x y(x,t) \subset \{-R^{-1}U : (U,R,V) \in \partial\Phi(x,y(x,t),t), U \in \mathbb{R}^{n\times m}, R \in \mathbb{R}^{n\times n}, V \in \mathbb{R}^{n\times l}\}.$$

This shows the first differential inclusion. The second inclusion is well known; see, for example, [3]. To show the uniform boundedness of $\partial_x y(x,t)$, we use the first differential inclusion. Thus it suffices to show the uniform boundedness of R^{-1} and U. Since F is uniformly strongly monotone, by Proposition 2.1, R^{-1} is uniformly bounded, and since F is uniformly Lipschitz continuous in x, U is uniformly bounded. The uniform global Lipschitz continuity of $y(x,t)$ in x follows subsequently.

Part (iv). We can show the differential inclusions as in Part (iii) by using Part (ii) of Lemma 2.2 with respect to y and t. To show the uniform boundedness of $\partial_t y(x,t)$, it suffices to show the uniform boundedness of R^{-1} and V. Since F is uniformly strongly monotone, by Proposition 2.1, R^{-1} is uniformly bounded, and since F is uniformly Lipschitz continuous in t, V is uniformly bounded. The uniform global Lipschitz continuity of $y(x,t)$ in t follows subsequently. □

Proof of Proposition 2.7. By Lemma 2.5, $\mathcal{T}(x)$ is Lebesgue measurable, and by Assumption 2.6, the Lebesgue measure of $\mathcal{T}(x)$ is zero.

Let $x' \in \mathcal{X}$ be any point close to x, let

$$\xi^T = \int_{\mathcal{T}\setminus\mathcal{T}(x)} [\nabla_x f(x,y(x,t),t) + \nabla_y f(x,y(x,t),t)\nabla_x y(x,t)]\rho(t)dt.$$

Let

$$R(x',x) = (E(x') - E(x) - \xi^T(x'-x))/\|x'-x\|.$$

Then

$$R(x',x) = R_1(x',x) + R_2(x',x) + R_3(x',x),$$

where

$$R_1(x',x) = \int_{\mathcal{T}\setminus\mathcal{T}(x)} \left[\frac{f(x',y(x',t),t) - f(x,y(x',t),t) - \nabla_x f(x,y(x,t),t)(x'-x)}{\|x'-x\|}\right]\rho(t)dt$$

and

$$R_2(x',x) = \int_{\mathcal{T}\setminus\mathcal{T}(x)} \left[\frac{f(x,y(x',t),t) - f(x,y(x,t),t) - \nabla_y f(x,y(x,t),t)\nabla_x y(x,t)(x'-x)}{\|x'-x\|}\right]\rho(t)dt$$

and

$$R_3(x',x) = \int_{\mathcal{T}(x)} \left[\frac{f(x', y(x',t),t) - f(x, y(x,t),t)}{\|x'-x\|} \right] \rho(t)dt.$$

Since f is continuously differentiable in x, it is obvious that $R_1(x', x) \to 0$ as $x' \to x$.

We now estimate R_3.

$$\begin{aligned} |R_3(x',x)| &\leq \int_{\mathcal{T}(x)} \frac{|f(x', y(x',t),t) - f(x, y(x,t),t)|}{\|x'-x\|} \rho(t)dt \\ &\leq \int_{\mathcal{T}(x)} \frac{L(t)(\|x'-x\| + \|y(x',t) - y(x,t)\|)}{\|x'-x\|} \rho(t)dt \\ &\leq (1+C) \int_{\mathcal{T}(x)} L(t)\rho(t)dt \\ &= 0. \end{aligned}$$

The last equality is due to the fact that the Lebseque measure of $\mathcal{T}(x)$ is zero.

Finally, we estimate $R_2(x', x)$. By (12) and twice continuous differentiability of f, we have

$$f(x, y(x',t),t) - f(x, y(x,t),t) - \nabla_y f(x, y(x,t),t)\nabla_x y(x,t)(x'-x) = o(\|x'-x\|)$$

which implies

$$R_2(x',x) \to 0, \text{ as } x' \to x.$$

This shows

$$R(x',x) \to 0, \text{ as } x' \to x,$$

and hence (13).

Now we show the continuity of $\nabla E(\cdot)$,

$$\begin{aligned} \nabla_x E(x') - \nabla_x E(x) &= \int_{\mathcal{T}\backslash\mathcal{T}(x')} [\nabla_x f(x', y(x',t),t) + \nabla_y f(x', y(x',t),t)\nabla_x y(x',t)]\rho(t)dt \\ &\quad - \int_{\mathcal{T}\backslash\mathcal{T}(x)} [\nabla_x f(x, y(x,t),t) + \nabla_y f(x, y(x,t),t)\nabla_x y(x,t)]\rho(t)dt \\ &= \int_{\mathcal{T}\backslash\mathcal{T}(x')\cup\mathcal{T}(x)} [\nabla_x f(x', y(x',t),t) + \nabla_y f(x', y(x',t),t)\nabla_x y(x',t) \\ &\quad -\nabla_x f(x, y(x,t),t) + \nabla_y f(x, y(x,t),t)\nabla_x y(x,t)]\rho(t)dt \\ &\quad + \int_{\mathcal{T}(x)\backslash\mathcal{T}(x')\cap\mathcal{T}(x)} [\nabla_x f(x', y(x',t),t) \\ &\quad +\nabla_y f(x', y(x',t),t)\nabla_x y(x',t)]\rho(t)dt \\ &\quad - \int_{\mathcal{T}(x')\backslash\mathcal{T}(x')\cap\mathcal{T}(x)} [\nabla_x f(x, y(x,t),t) \\ &\quad +\nabla_y f(x, y(x,t),t)\nabla_x y(x,t)]\rho(t)dt \end{aligned}$$

We show that the three terms at the right-hand side of the last equality tends to zero as $x' \to x$. Since $\mathcal{T}(x') \to \mathcal{T}(x)$ as $x' \to x$, the Lebesgue measure of $\mathcal{T}\backslash\mathcal{T}(x')\cup\mathcal{T}(x)$ tends to that of $\mathcal{T}$. Moreover, ∇f is uniformly continuous in x, y by assumption, $y(x', t)$ uniformly approximates $y(x, t)$ by part (iv) of Theorem 2.3, and $\nabla y(x', t)$ uniformly approximates $\nabla y(x, t)$ by (12). This shows the first term tends to zero. The proofs for the second and third terms are similar. This completes the proof. □

Acknowledgments. I would like to thank Daniel Ralph for helpful comments and suggestions particularly regarding the optimality conditions. I would also like to thank Houyuan Jiang, Gui-Hua Lin, and two anonymous referees for careful reading of the paper and helpful comments which lead to a significant improvement of the paper.

REFERENCES

[1] X. Chen, L. Qi, and D. Sun, *Global and superlinear convergence of the smoothing Newton's method and its application to general box constrained variational inequalities*, Math. Comp., 67 (1998), pp. 519–540.

[2] S. Christiansen, M. Patriksson, and L. Wynter, *Stochastic Bilevel Programming in Structral Optimization*, preprint, PRISM, Université de Versailles, Versailles, France, 1999.

[3] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Wiley, New York, 1983.

[4] D. De Wolf and Y. Smeers, *A Stochastic version of a Stackelberg–Nash–Cournot equilibrium model*, Management Sciences, 43 (1997), pp. 190–197.

[5] A. Evgrafov and M. Patriksson, *On the existence of solutions to stochastic mathematical programs with equilibrium constraints*, J. Optim. Theory Appl., 121 (2004), pp. 65–76.

[6] F. Facchinei, H. Jiang, and L. Qi, *A smoothing method for mathematical programs with equilibrium constraints*, Math. Program., 85 (1999), pp. 81–106.

[7] F. Facchinei and J. S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Springer-Verlag, New York, 2003.

[8] M. Ferris and J. S. Pang, *Engineering and economic applications of complementarity problems*, SIAM Rev., 39 (1997), pp. 669–713.

[9] A. Fischer, *A special Newton-type optimization method*, Optimization, 24 (1992), pp. 269–284.

[10] R. Fletcher and S. Leyffer, *Numerical experience with solving MPECs as NLPs*, University of Dundee Report NA210, Dundee, UK, 2002.

[11] R. Fletcher, S. Leyffer, D. Ralph, and S. Scholtes, *Local Convergence of SQP Methods for Mathematical Programs with Equilibrium Constraints*, University of Dundee Report NA209, Dundee, UK, 2002.

[12] M. Fukushima and J.-S. Pang, *Convergence of a smoothing continuation method for mathematical programs with complementarity constraints*, Ill-posed Variational Problems and Regularization Techniques (Trier 1998), Lecture Notes in Econom. Math. Systems, 477, Springer-Verlag, Berlin, 1999, pp. 99–110.

[13] M. Fukushima and P. Tseng, *An implementable active-set algorithm for computing a B-stationary point of a mathematical program with linear complementarity constraints*, SIAM J. Optim., 12 (2002), pp. 724–739.

[14] J.-B. Hiriart-Urruty, *Refinements of necessary optimality conditions in nondifferentiable programming*, I, Appl. Math. Optim., 5 (1979), pp. 63–82.

[15] X. Hu and D. Ralph, *A note on sensitivity of value functions of mathematical programs with complementarity constraints*, Math. Program., 93 (2002), pp. 265–279.

[16] H. Jiang and D. Ralph, *Smooth SQP methods for mathematical programs with nonlinear complementarity constraints*, SIAM J. Optim., 10 (2000), pp. 779–808.

[17] H. Jiang and L. Qi, *A new nonsmooth equations approach to nonlinear complementarity problems*, SIAM J. Control Optim., 35 (1997), pp. 178–193.

[18] C. Kanzow, *Some noninterior continuation methods for linear complementarity problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 851–868.

[19] S. Leyffer, *Mathematical Programs with Complementarity Constraints*, SIAG/OPT Views-and-News, 14 (2003), pp. 15–18.

[20] G.-H. Lin, X. Chen, and M. Fukushima, *Smoothing Implicit Programming Approaches for Stochastic Mathematical Programs with Linear Complementarity Constraints*, http://www.amp.i.kyoto-u.ac.jp/tecrep/index-e.html, 2003.

[21] G.-H. Lin, X. Chen, and M. Fukushima, *Combined smoothing implicit programming and penalty method for stochastic method for stochastic mathematical programs with equilibrium constraints*, Kyoto University, Japan, 2004, preprint.

[22] G.-H Lin and M. Fukushima, *A class of stochastic mathematical programs with complementarity constraints: reformulations and algorithms*, J. Ind. Manag. Optim., 1 (2005), pp. 99–122.

[23] Z. Q. Luo, J.-S. Pang, and D. Ralph, *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press, Cambridge, UK, 1996.

[24] F. H. Murphy, H. D. Sherali, and A. L. Soyster, *A mathematical programming approach for determining oligopolistic market equilibruium*, Math. Program., 24 (1982), pp. 92–106.

[25] J. Outrata, M. Kocvara, and J. Zowe, *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints*, Theory, Applications, and Numerical Constraints, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[26] M. Patriksson and L. Wynter, *Stochastic mathematical programs with equilibrium constraints*, Oper. Res. Letters, 25 (1999), pp. 159–167.

[27] M. Patriksson and L. Wynter, *Stochastic mathematical programs with equilibrium constraints*, Oper. Res. Lett., 25(1999), pp. 159–167.

[28] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, Springer-Verlag, Berlin, 1998.

[29] D. Ralph and H. Xu, *Implicit smoothing and its application to optimization with piecewise smooth equality constraints*, J. Optim. Theory Appl., 124 (2005), pp. 673–699.

[30] E. Polak, D. Q. Mayne, and Y. Wardi, *On the extension of constrained optimization algorithms from differentiable to nondifferentiable problems*, SIAM J. Control Optim., 21 (1983), pp. 179–203.

[31] A. Shapiro, *Stochastic mathematical programs with equilibrium constraints*, , School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 2004, preprint. To appear in J. Optim. Theory Appl.

[32] H. D. Sherali, A. L. Soyster, and F. H. Murphy, *Stackelberg–Nash–Cournot equilibria: Characterizations and computations*, Oper. Res., 31 (1983), pp. 253–276.

[33] H. D. Sherali, *A multiple leader Stackelberg model and analysis*, Oper. Res., 32 (1984), pp. 390–404.

[34] S. Scholtes, *Convergence properties of a regularization scheme for mathematical programs with complementarity constraints*, SIAM J. Optim., 11 (2001), pp. 918–936.

[35] D. Sun and L. Qi, *On NCP functions*, Comput. Optim. Appl., 13 (1999), pp. 201–220.

[36] H. Xu, *An MPCC approach for stochastic Stackelberg–Nash–Cournot equilibrium*, Optimization, 54 (2005), pp. 27–57.

[37] H. Xu and X. W. Chang, *Approximate Newton methods for nonsmooth equations*, J. Optim. Theory Appl., 93 (1997), pp. 373–394.

SIAM J. OPTIM.
Vol. 16, No. 3, pp. 697–725

INTERIOR GRADIENT AND PROXIMAL METHODS FOR CONVEX AND CONIC OPTIMIZATION[*]

ALFRED AUSLENDER[†] AND MARC TEBOULLE[‡]

Abstract. Interior gradient (subgradient) and proximal methods for convex constrained minimization have been much studied, in particular for optimization problems over the nonnegative octant. These methods are using non-Euclidean projections and proximal distance functions to exploit the geometry of the constraints. In this paper, we identify a simple mechanism that allows us to derive global convergence results of the produced iterates as well as improved global rates of convergence estimates for a wide class of such methods, and with more general convex constraints. Our results are illustrated with many applications and examples, including some new explicit and simple algorithms for conic optimization problems. In particular, we derive a class of interior gradient algorithms which exhibits an $O(k^{-2})$ global convergence rate estimate.

Key words. convex optimization, interior gradient/subgradient algorithms, proximal distances, conic optimization, convergence and efficiency

AMS subject classifications. 90C25, 90C30, 90C22

DOI. 10.1137/S1052623403427823

1. Introduction.

Consider the following convex minimization problem:

$$\text{(P)} \qquad f_* = \inf\{f(x) \mid x \in \overline{C}\},$$

where $\overline{C}$ denotes the closure of C, a nonempty convex open set in $\mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a proper, lower semicontinuous (lsc) convex function. In this paper we study two closely related iterative schemes for solving (P). The first one is proximal based. Given some proximity measure d, it consists of generating a sequence $\{x^k\}$ via the iteration

$$x^k \in \operatorname{argmin}\{\lambda_k f(x) + d(x, x^{k-1}) \mid x \in \overline{C}\}, \quad k = 1, 2, \dots \quad (\lambda_k > 0). \tag{1.1}$$

The second iterative scheme is subgradient based (or explicit proximal) and produces a sequence $\{x^k\}$ via

$$x^k \in \operatorname{argmin}\{\lambda_k \langle g^{k-1}, x\rangle + d(x, x^{k-1}) \mid x \in \overline{C}\}, \quad k = 1, 2, \dots, \tag{1.2}$$

where $\langle \cdot, \cdot \rangle$ is an inner product on $\mathbb{R}^n$ and g^{k-1} is a subgradient of the function f at the point x^{k-1}. With the choice $d(x, y) = 2^{-1}\|x - y\|^2$, one recovers the proximal algorithm (PA) (see, e.g., Martinet [31] and Rockafellar [40]) and the projected subgradient method (see, e.g., [41]), respectively. In that case, the sequence $\{x^k\}$ produced by either one of the above algorithms does not necessarily belong to C. In this paper, the proximal term $d(x, y)$ will play the role of a distance-like function satisfying certain desirable properties (see section 2), which will force the iterates of the produced sequence to stay in C, and thus automatically eliminate the constraints.

[*] Received by the editors May 13, 2003; accepted for publication (in revised form) June 4, 2005; published electronically January 6, 2006.
http://www.siam.org/journals/siopt/16-3/42782.html

[†] Département de Mathématiques, University Lyon I, Lyon, France (aauslen@cegetel.net).

[‡] School of Mathematical Sciences, Tel-Aviv University, Ramat-Aviv 69978, Israel (teboulle@math.tau.ac.il). This author's work was partially supported by the United States–Israel Binational Science Foundation, BSF grant 2002-2010.

1.1. Motivation and related works. The idea of replacing the quadratic proximal term by a proximal function $d(x, y)$ has been pursued in the literature in several works. In the context of proximal-based methods of the form (1.1), two popular choices for d include either a Bregman distance (see, e.g., [15, 16, 19, 29]) or a φ-divergence distance (see, e.g., [21, 43, 44]). More recent works have also proposed proximal methods based on second order homogeneous kernels; see, e.g., [4, 5, 10, 42, 45]. These works have concentrated on the ground set $\overline{C}$ being polyhedral and in particular when $\overline{C}$ is the nonnegative octant in $\mathbb{R}^n$. For semidefinite programming problems, two particular proximal distances were proposed by Doljansky and Teboulle [18]. Furthermore, applications of these algorithms to the dual of convex programs, leading to smooth Lagrangian multiplier methods as well as extension to variational inequalities over polyhedral constraints, have also been developed in many studies, e.g., [3, 27, 28, 36, 37]. More recent applications include continuous time models of proximal-based methods; see, e.g., [1, 2, 13] and references therein.

In the context of explicit proximal methods, namely subgradient projection-type algorithms of the form (1.2), a recent paper of Ben-Tal, Margalit, and Nemirovski [9] has shown that an algorithm based on the mirror descent algorithm of Nemirovski and Yudin introduced in [32] can be used to solve efficiently convex minimization problems over the unit simplex with millions of variables. In a more recent study, Beck and Teboulle [8] have shown that the mirror descent method can be viewed as a subgradient projection algorithm based on a Bregman distance and have proposed a specific variant for convex minimization over the unit simplex. Other interior gradient schemes can be found, for example, in [25, 26] and references therein. These two works study multiplicative interior gradient-type schemes for minimizing a continuously differentiable function over the nonnegative octant under various assumptions, the former being a scheme suggested by [21], and the latter being based on the φ-divergence distance. The revived interest in such gradient-type methods relies mainly on the following facts. They require only first order information (e.g., function and subgradient evaluation at each step), they often lead to simple iterative schemes for particular types of constraints (e.g., by picking the appropriate proximal distance), and they exhibit a nearly dimension independent computational complexity in terms of the problem's dimension; see, e.g., [9, 8]. One main disadvantage of gradient-based methods is that they often share a slow convergence rate for producing high accuracy solutions, typically an $O(k^{-1})$ global convergence rate estimate for function values, where k is the iteration counter; see, e.g., [32, 41]. In comparison, the theoretically more efficient polynomial interior point methods (IPM) can achieve high accuracy but require second order derivative information and an increase in computational effort that depends on and grows polynomially in the design dimension n of the problem. Thus, gradient-based methods can be a suitable practical alternative for solving large-scale applications where high accuracy is not needed, and where the IPM are not computationally tractable; see [9] and references therein.

1.2. Main contributions and summary of results. Motivated by all these works, this paper focuses on the following three main theoretical goals: (a) to uncover the main tools needed to analyze and design interior gradient and proximal methods, (b) to establish convergence and global efficiency estimates for the basic representative schemes of these methods, and (c) to devise some new methods with improved complexity. To achieve these goals, we first develop a general and simple principle, also capable of handling more general constraints than the one alluded to above. This is achieved by identifying the common mechanism underlying the analysis of interior proximal methods. This is developed in section 2, where we derive general results on the convergence of the sequence produced by proximal-type methods and establish global rates of convergence estimates in terms of function values. This development allows us to recover some of the well-known variants of such methods, and to derive and analyze new schemes. This is illustrated in section 3, which includes many examples and applications. In section 4, we continue along this line of analysis, to develop a simple and general framework for the interior subgradient/gradient methods, akin to the ones given in [9, 8]. The interior gradient methods we propose include both fixed and Armijo–Goldstein stepsize rules. Applications of these results to conic optimization and to convex minimization over the unit simplex are given. In particular we propose, for the first time to our knowledge, new explicit interior gradient methods for semidefinite and second order conic programs. Further motivated by the discussion and references outlined above on the potential usefulness of interior gradient methods, it is natural to ask if one can devise simple interior gradient algorithms with an improved computational complexity. Building on our previous results, and inspired by the work of Nesterov [34], we answer this question positively in the last section for one of the class of interior gradient algorithms discussed in section 4. The scheme we propose naturally extends the optimal classical gradient scheme given in [34], and it leads to a class of interior gradient algorithms for solving conic problems which exhibits the faster global convergence rate estimate $O(k^{-2})$.

1.3. Notation. We adopt the standard notation of convex analysis [39]. For a proper convex and lsc function $F : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, its effective domain is defined by $\operatorname{dom} F = \{x \mid F(x) < +\infty\}$, and for all $\epsilon \geq 0$ its ϵ-subdifferential at x is defined by $\partial_\epsilon F(x) = \{g \in \mathbb{R}^n \mid \forall z \in \mathbb{R}^n,\ F(z) + \epsilon \geq F(x) + \langle g, z - x\rangle\}$, which coincides with the usual subdifferential $\partial F \equiv \partial_0 F$ whenever $\epsilon = 0$. We set $\operatorname{dom} \partial F = \{x \in \mathbb{R}^n \mid \partial F(x) \neq \emptyset\}$. For any closed convex set $S \subset \mathbb{R}^n$, δ_S denotes the indicator function of S, $\operatorname{ri} S$ its relative interior, and $N_S(x) = \partial \delta_S(x) = \{\nu \in \mathbb{R}^n \mid \langle \nu, z - x\rangle \leq 0\ \forall z \in S\}$ the normal cone to S at $x \in S$. The set of n-vectors with nonnegative (positive) components is denoted by $\mathbb{R}^n_+$ ($\mathbb{R}^n_{++}$).

2. A general framework for interior proximal methods.

Let C be a nonempty convex open set in $\mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ a proper, lsc, and convex function. Consider the optimization problem

$$\text{(P)} \qquad f_* = \inf\{f(x) \mid x \in \overline{C}\},$$

where $\overline{C}$ denotes the closure of C. Unless otherwise specified, throughout this paper we make the following standing assumptions on (P):

(a) $\operatorname{dom} f \cap C \neq \emptyset$,

(b) $-\infty < f_*$.

We study the behavior of the following basic proximal iterative scheme to solve (P):

$$x^k \in \operatorname{argmin}\{\lambda_k f(x) + d(x, x^{k-1}) \mid x \in \overline{C}\}, \quad k = 1, 2, \dots \quad (\lambda_k > 0),$$

where d is some proximal distance. Our approach is motivated by and patterned after many of the studies mentioned in the introduction, and our objective is to develop a general framework to analyze the convergence of the resulting methods under various settings. Given the optimization problem (P), essentially the basic ingredients needed to achieve the aforementioned goals are

- to pick an appropriate proximal distance d which allows us to eliminate the constraints,

- given d, to find an induced proximal distance H, which will control the behavior of the resulting method.

We begin by defining an appropriate proximal distance d for problem (P).

Definition 2.1. *A function $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+ \cup \{+\infty\}$ is called a proximal distance with respect to an open nonempty convex set $C \subset \mathbb{R}^n$ if for each $y \in C$ it satisfies the following properties:*

(P_1) *$d(\cdot, y)$ is proper, lsc, convex, and C^1 on C;*

(P_2) *$\operatorname{dom} d(\cdot, y) \subset \overline{C}$ and $\operatorname{dom} \partial_1 d(\cdot, y) = C$, where $\partial_1 d(\cdot, y)$ denotes the subgradient map of the function $d(\cdot, y)$ with respect to the first variable;*

(P_3) *$d(\cdot, y)$ is level bounded on $\mathbb{R}^n$, i.e., $\lim_{\|u\| \to \infty} d(u, y) = +\infty$;*

(P_4) *$d(y, y) = 0$.*

We denote by $\mathcal{D}(C)$ the family of functions d satisfying Definition 2.1. Property (P_1) is needed to preserve convexity of $d(\cdot, y)$, (P_2) will force the iterate x^k to stay in C, and (P_3) is used to guarantee the existence of such an iterate. For each $y \in C$, let $\nabla_1 d(\cdot, y)$ denote the gradient map of the function $d(\cdot, y)$ with respect to the first variable. Note that by definition $d(\cdot, \cdot) \geq 0$, and from (P_4) the global minimum of $d(\cdot, y)$ is obtained at y, which shows that $\nabla_1 d(y, y) = 0$.

Proposition 2.1. *Let $d \in \mathcal{D}(C)$, and for all $y \in C$ consider the optimization problem*

$$P(y) \qquad f_*(y) = \inf\{f(u) + d(u, y) \mid u \in \mathbb{R}^n\}.$$

Then the optimal set $S(y)$ of $P(y)$ is nonempty and compact, and for each $\epsilon \geq 0$ there exist $u(y) \in C$, $g \in \partial_\epsilon f(u(y))$ such that

$$g + \nabla_1 d(u(y), y) = 0, \tag{2.1}$$

where $\partial_\epsilon f(u(y))$ denotes the ϵ-subdifferential of f at $u(y)$. For such a $u(y) \in C$ we have

$$f(u(y)) + d(u(y), y) \leq f_*(y) + \epsilon. \tag{2.2}$$

Proof. We set $t(u) = f(u) + d(u, y) + \delta_{\overline{C}}(u)$. Then by ($P_2$) we have $f_*(y) = \inf\{t(u) \mid u \in \mathbb{R}^n\}$. Furthermore, since f_* is finite, it follows by (P_3) that $t(\cdot)$ is level bounded. Therefore with $t(\cdot)$ being a proper, lsc convex function, it follows that $S(y)$ is nonempty and compact. From the optimality conditions, for each $u(y) \in S(y)$ we have $0 \in \partial t(u(y))$. Now, since $\operatorname{dom} f \cap C \neq \emptyset$ and C is open, we can apply [39, Theorem 23.8] so that

$$\partial t(u) = \partial f(u) + \nabla_1 d(u, y) + N_{\overline{C}}(u) \quad \forall u.$$

Since $\operatorname{dom} \partial_1 d(\cdot, y) = C$, it follows that $u(y) \in C$, and hence $N_{\overline{C}}(u(y)) = \{0\}$, and (2.1) holds for $\epsilon = 0$ with $g \in \partial f(u(y))$. For $\epsilon > 0$, (2.1) holds for such a pair $(u(y), g)$ since $\partial f(u(y)) \subset \partial_\epsilon f(u(y))$, and thus the first part of the proposition is proved. Finally, since for each $y \in C$ the function $d(\cdot, y)$ is convex, and since $g \in \partial_\epsilon f(u(y))$, we have

$$f(u) + d(u, y) \geq f(u(y)) + d(u(y), y) + \langle g + \nabla_1 d(u(y), y), u - u(y)\rangle - \epsilon$$

so that $f_*(y) = \inf\{f(u) + d(u, y) \mid u \in \overline{C}\} \geq f(u(y)) + d(u(y), y) - \epsilon$. □

Thanks to the above proposition, the following basic algorithm is well defined.

Interior proximal algorithm (IPA). Given $d \in \mathcal{D}(C)$, start with a point $x^0 \in C$, and for $k = 1, 2, \dots$ with $\lambda_k > 0$, $\epsilon_k \geq 0$, generate a sequence

$$\{x^k\} \in C \text{ with } g^k \in \partial_{\epsilon_k} f(x^k) \tag{2.3}$$

such that

$$\lambda_k g^k + \nabla_1 d(x^k, x^{k-1}) = 0. \tag{2.4}$$

The IPA can be viewed as an approximate interior proximal method when $\epsilon_k > 0$ $\forall k \in \mathbb{N}$ (the set of natural numbers), which becomes exact for the special case $\epsilon_k = 0$ $\forall k \in \mathbb{N}$.

The next step is to associate with each given $d \in \mathcal{D}(C)$ a corresponding proximal distance satisfying some desirable properties needed to analyze the IPA.

Definition 2.2. *Given $C \subset \mathbb{R}^n$, open and convex, and $d \in \mathcal{D}(C)$, a function $H : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+ \cup \{+\infty\}$ is called the induced proximal distance to d if H is finite valued on $C \times C$ and for each $a, b \in C$ satisfies*

$$H(a, a) = 0, \tag{2.5}$$

$$\langle c - b, \nabla_1 d(b, a) \rangle \leq H(c, a) - H(c, b) \quad \forall c \in C. \tag{2.6}$$

We write $(d, H) \in \mathcal{F}(C)$ to quantify the triple $[C, d, H]$ that satisfies the premises of Definition 2.2.

Likewise, we will write $(d, H) \in \mathcal{F}(\overline{C})$ for the triple $[\overline{C}, d, H]$ whenever there exists H which is finite valued on $\overline{C} \times C$, satisfies (2.5)–(2.6) for any $c \in \overline{C}$, and is such that $\forall c \in \overline{C}$ one has $H(c, \cdot)$ level bounded on C. Clearly, one has $\mathcal{F}(\overline{C}) \subset \mathcal{F}(C)$.

The motivation behind such a construction is not as mysterious as it might look at first sight. Indeed, for the moment, notice that the classical PA, which corresponds to the special case $C = \overline{C} = \mathbb{R}^n$, $d(x, y) = 2^{-1}\|x - y\|^2$ and the induced proximal distance H being exactly d, clearly satisfies (2.6), thanks to the well-known identity

$$\|z - x\|^2 = \|z - y\|^2 + \|y - x\|^2 + 2\langle z - y, y - x \rangle.$$

IPA with $d \equiv H$ will be called *self-proximal.* Several useful examples of more general self-proximal methods for various classes of constraint sets C will be given in the next section.

As we shall see below, the requested properties for the function H associated with d naturally emerge from the analysis of the classical PA as given in [24] and later extended for various specific classes of IPA in [16, 44, 5]. Building on these works, we can already easily obtain global rates of convergence estimates as well as convergence in limit points of the produced sequence by IPA. To derive the global convergence of the sequence $\{x^k\}$ to an optimal solution of (P), additional assumptions on the induced proximal distance H, akin to the properties of norms, will be required.

Before giving our convergence results, we recall the following well-known properties on nonnegative sequences, which will be useful to us throughout this work.

Lemma 2.1 (see [35]). *Let $\{v_k\}$, $\{\gamma_k\}$, and $\{\beta_k\}$ be nonnegative sequences of real numbers satisfying $v_{k+1} \leq (1+\gamma_k)v_k + \beta_k$ and such that $\sum_{k=1}^{\infty} \beta_k < \infty$, $\sum_{k=1}^{\infty} \gamma_k < \infty$. Then, the sequence $\{v_k\}$ converges.*

Lemma 2.2 (see [35]). *Let $\{\lambda_k\}$ be a sequence of positive numbers, $\{a_k\}$ a sequence of real numbers, and $b_n := \sigma_n^{-1} \sum_{k=1}^{n} \lambda_k a_k$, where $\sigma_n = \sum_{k=1}^{n} \lambda_k$. If $\sigma_n \to \infty$, one has*

(i) $\liminf a_n \le \liminf b_n \le \limsup b_n \le \limsup a_n$,

(ii) $\lim b_n = a$ *whenever* $\lim a_n = a$.

THEOREM 2.1. *Let* $(d, H) \in \mathcal{F}(C)$ *and let* $\{x^k\}$ *be the sequence generated by IPA. Set* $\sigma_n = \sum_{k=1}^n \lambda_k$. *Then the following hold:*

(i) $f(x^n) - f(x) \le \sigma_n^{-1} H(x, x^0) + \sigma_n^{-1} \sum_{k=1}^n \sigma_k \epsilon_k \ \forall x \in C$.

(ii) *If* $\lim_{n\to\infty} \sigma_n = +\infty$ *and* $\epsilon_k \to 0$, *then* $\liminf_{n\to\infty} f(x^n) = f_*$ *and the sequence* $\{f(x^k)\}$ *converges to* f_* *whenever* $\sum_{k=1}^\infty \epsilon_k < \infty$.

(iii) *Furthermore, suppose the optimal set* X_* *of problem* (P) *is nonempty, and consider the following cases:*

(a) X_* *is bounded,*

(b) $\sum_{k=1}^\infty \lambda_k \epsilon_k < \infty$ *and* $(d, H) \in \mathcal{F}(\overline{C})$.

Then, under either (a) *or* (b), *the sequence* $\{x^k\}$ *is bounded with all its limit points in* X_*.

Proof. (i) From (2.4), since $g^k \in \partial_{\epsilon_k} f(x^k)$ we have

$$\lambda_k(f(x^k) - f(x)) \le \langle x - x^k, \nabla_1 d(x^k, x^{k-1})\rangle + \lambda_k \epsilon_k \quad \forall x \in C. \tag{2.7}$$

Using (2.6) at the points $c = x$, $a = x^{k-1}$, $b = x^k$, the above inequality implies that

$$\lambda_k(f(x^k) - f(x)) \le H(x, x^{k-1}) - H(x, x^k) + \lambda_k \epsilon_k \quad \forall x \in C. \tag{2.8}$$

Summing over $k = 1, \dots, n$ we obtain

$$-\sigma_n f(x) + \sum_{k=1}^n \lambda_k f(x^k) \le H(x, x^0) - H(x, x^n) + \sum_{k=1}^n \lambda_k \epsilon_k. \tag{2.9}$$

Now setting $x = x^{k-1}$ in (2.8), we obtain

$$f(x^k) - f(x^{k-1}) \le \epsilon_k. \tag{2.10}$$

Multiplying the latter inequality by σ_{k-1} (with $\sigma_0 \equiv 0$) and summing over $k = 1, \dots, n$, we obtain, after some algebra,

$$\sigma_n f(x^n) - \sum_{k=1}^n \lambda_k f(x^k) \le \sum_{k=1}^n \sigma_{k-1} \epsilon_k.$$

Adding this inequality to (2.9) and recalling that $\lambda_k + \sigma_{k-1} = \sigma_k$, it follows that

$$f(x^n) - f(x) \le \sigma_n^{-1}[H(x, x^0) - H(x, x^n)] + \sigma_n^{-1} \sum_{k=1}^n \sigma_k \epsilon_k \quad \forall x \in C, \tag{2.11}$$

proving (i), since $H(\cdot, \cdot) \ge 0$.

(ii) If $\sigma_n \to +\infty$ and $\epsilon_k \to 0$, then dividing (2.9) by σ_n and invoking Lemma 2.2(i), we obtain from (2.9) that $\liminf_{n\to\infty} f(x^n) \le \inf\{f(x) \mid x \in C\}$, which together with $f(x^n) \ge \inf\{f(x) \mid x \in \overline{C}\}$ implies that $\liminf_{n\to\infty} f(x^n) = \inf\{f(x) \mid x \in \overline{C}\} = f_*$. From (2.10) we have

$$0 \le f(x^k) - f_* \le f(x^{k-1}) - f_* + \epsilon_k.$$

Then using Lemma 2.1 it follows that the sequence $\{f(x^k)\}$ converges to f_* whenever $\sum_{k=1}^\infty \epsilon_k < \infty$.

(iii) Case (a): If X_* is bounded, then f is level bounded over $\overline{C}$, and since the sequence $\{f(x^k)\}$ converges to f_*, it follows that the sequence $\{x^k\}$ is bounded. Since f is lsc, passing to the limit, and recalling that $\{x^k\} \subset C$, it follows that each limit point is an optimal solution.

Case (b): Here, we suppose that $\sum_{k=1}^{\infty} \lambda_k \epsilon_k < \infty$ and that $(d, H) \in \mathcal{F}(\overline{C})$. Then (2.8) holds for each $x \in \overline{C}$, and in particular for $x \in X_*$, so that

$$H(x, x^k) \leq H(x, x^{k-1}) + \lambda_k \epsilon_k \quad \forall x \in X_*. \tag{2.12}$$

Summing over $k = 1, \dots, n$, we obtain

$$H(x, x^n) \leq H(x, x^0) + \sum_{k=1}^{\infty} \lambda_k \epsilon_k.$$

But, since in this case $H(x, \cdot)$ is level bounded, the last inequality implies that the sequence $\{x^k\}$ is bounded, and thus as in Case (a) it follows that all its limit points are in X_*. □

An immediate byproduct of the above analysis yields the following global rate of convergence estimate for the exact version of IPA, i.e., with $\epsilon_k = 0 \; \forall k$.

COROLLARY 2.1. *Let $(d, H) \in \mathcal{F}(\overline{C})$, $X_* \neq \emptyset$, and $\{x^k\}$ be the sequence generated by IPA with $\epsilon_k = 0 \; \forall k$. Then, $f(x^n) - f_* = O(\sigma_n^{-1}) \; \forall x \in \overline{C}$.*

Proof. Under the given hypothesis, Theorem 2.1(i) holds for any $x \in \overline{C}$, and it follows that $f(x^n) - f_* \leq (\sigma_n)^{-1} H(x^*, x^0)$. □

To establish the global convergence of the sequence $\{x^k\}$ to an optimal solution of problem (P), we need to make further assumptions on the induced proximal distance H, mimicking the behavior of norms.

Let $(d, H) \in \mathcal{F}_+(\overline{C}) \subset \mathcal{F}(\overline{C})$ be such that the function H satisfies the following two additional properties:

(a_1) $\forall y \in \overline{C}$ and $\forall \{y_k\} \subset C$ bounded with $\lim_{k \to +\infty} H(y, y_k) = 0$, we have $\lim_{k \to +\infty} y_k = y$;

(a_2) $\forall y \in \overline{C}$ and $\forall \{y_k\} \subset C$ converging to y, we have $\lim_{k \to +\infty} H(y, y_k) = 0$.

With these additional hypotheses on H we immediately obtain that IPA globally converges to an optimal solution of (P).

THEOREM 2.2. *Let $(d, H) \in \mathcal{F}_+(\overline{C})$ and let $\{x^k\}$ be the sequence generated by IPA. Suppose that the optimal set X_* of* (P) *is nonempty, $\sigma_n = \sum_{k=1}^{n} \lambda_k \to \infty$, $\sum_{k=1}^{\infty} \lambda_k \epsilon_k < \infty$, and $\sum_{k=1}^{\infty} \epsilon_k < \infty$. Then the sequence $\{x^k\}$ converges to an optimal solution of* (P).

Proof. Let $x \in X_*$. Then, since $(d, H) \in \mathcal{F}_+(\overline{C})$, from (2.12) with $\sum_{k=1}^{n} \lambda_k \epsilon_k < +\infty$ and Lemma 2.1 we obtain that the sequence $\{H(x, x^k)\}$ converges to some $a(x) \in \mathbb{R} \; \forall x \in X_*$. Let x_∞ be the limit of a subsequence $\{x^{k_l}\}$. Obviously, from Theorem 2.1, $x_\infty \in X_*$. Then by assumption (a_2) $\lim_{l \to \infty} H(x_\infty, x^{k_l}) = 0$, so that $\lim_{k \to \infty} H(x_\infty, x^k) = 0$, and by assumption ($a_1$) it follows that the sequence $\{x^k\}$ converges to x_∞. □

Note that we have separated the two types of convergence results to emphasize

- the differences and roles played by each of the three classes $\mathcal{F}_+(\overline{C}) \subset \mathcal{F}(\overline{C}) \subset \mathcal{F}(C)$,
- that the largest, and less demanding, class $\mathcal{F}(C)$ already provides reasonable convergence properties for IPA, with minimal assumptions on the problem's data.

These aspects are illustrated by several application examples in the next section.

Relations (2.3), (2.4) defining IPA can sometimes be difficult to implement, since at each step we have to find by some algorithm in a finite number of steps an ϵ_k-solution for the minimization of the function $\lambda_k f(\cdot) + d(\cdot, x^{k-1})$. To overcome this difficulty, we consider here (among others) a variant of the approximate rule proposed in [20] for self-proximal Bregman methods.

Interior proximal algorithm with approximation rule (IPA1). Let $(d, H) \in \mathcal{F}(C)$, $\lambda_* > 0$, and for each $k = 1, 2, \ldots$, let $\lambda_k \geq \lambda_*$, $\eta_k > 0$, and $\epsilon_k > 0$ with $\sum_{k=1}^{\infty} \epsilon_k < \infty$, $\sum_{k=1}^{\infty} \eta_k < \infty$. Starting from a point $x^0 \in C$, for all $k \geq 1$ we generate the sequences $\{x^k\}_{k=1}^{\infty} \subset C$, $\{e^k\}_{k=1}^{\infty} \subset \mathbb{R}^n$ via

$$e^k = \lambda_k g^k + \nabla_1 d(x^k, x^{k-1}) \text{ with } g^k \in \partial f(x^k), \tag{2.13}$$

where the error sequence $\{e^k\}$ satisfies the conditions

$$\|e^k\| \leq \epsilon_k, \quad \|e^k\| \sup(\|x^k\|, \|x^{k-1}\|) \leq \eta_k. \tag{2.14}$$

Remark 2.1. From Proposition 2.1, a sequence $\{x^k\}$ given by relations (2.13), (2.14) always exists. Furthermore, if f is C^1 on C (C^2 on C with $d(\cdot, y) \in C^2$ on C for all $y \in C$), then any convergent gradient-type method (Newton-type method) will provide such an x^k in a finite number of steps.

Theorem 2.3. *Let $(d, H) \in \mathcal{F}(C)$, and let $\{x^k\}$ be a sequence generated by IPA*1. *Then we have the following:*

(i) *The sequence $\{f(x^k)\}$ converges to f_*.*

(ii) *Furthermore, suppose that the optimal set X_* is nonempty, and consider the following cases:*

  (a) *X_* is bounded;*

  (b) *$(d, H) \in \mathcal{F}(\overline{C})$;*

  (c) *$(d, H) \in \mathcal{F}_+(\overline{C})$.*

Then under (a) *or* (b), *the sequence $\{x^k\}$ is bounded with all limit points in X_*, while under* (c) *the sequence $\{x^k\}$ converges to an optimal solution.*

Proof. Since $g^k \in \partial f(x^k)$, using (2.6) and the Cauchy–Schwarz inequality we get for any $x \in C$

$$\begin{aligned} \lambda_k(f(x^k) - f(x)) &\leq \langle x - x^k, \nabla_1 d(x^k, x^{k-1}) \rangle + \langle e^k, x^k - x \rangle \\ &\leq H(x, x^{k-1}) - H(x, x^k) + \tilde{\epsilon_k}(x), \end{aligned} \tag{2.15}$$

with $\tilde{\epsilon_k}(x) := \|e^k\| \|x\| + \langle x^k, e^k \rangle$. Summing (2.15) over $k = 1, \ldots, n$ and dividing by $\sigma_n = \sum_{i=1}^n \lambda_k$ we obtain

$$-f(x) + \sum_{k=1}^{n} \frac{\lambda_k f(x^k)}{\sigma_n} \leq \sigma_n^{-1} \left[H(x, x^0) - H(x, x^n) + \sum_{k=1}^{n} \tilde{\epsilon_k}(x) \right]. \tag{2.16}$$

Now setting $x = x^{k-1}$ in (2.15) and $\alpha_k := |\tilde{\epsilon_k}(x^{k-1})| \lambda_*^{-1}$, we obtain $(f(x^k) - f(x^{k-1})) \leq \alpha_k$. But using (2.14), one has $\sum_{k=1}^{\infty} \tilde{\epsilon_k}(x) < \infty$ and $\sum_{k=1}^{\infty} \alpha_k < \infty$. Therefore by passing to the limit in (2.16) and invoking Lemma 2.2(i) it follows that $\liminf_{n \to \infty} f(x^n) - f(x) \leq 0$ for each $x \in C$ so that $\liminf_{n \to \infty} f(x^n) \leq \inf\{f(x) \mid x \in C\}$. From here, the proof can be completed with the same arguments as in the proofs of Theorems 2.1 and 2.2. □

Theorem 2.3(c) recovers and extends [20, Theorem 1, p. 120] for the case of convex minimization, which was proved there only for the Bregman self-proximal method.

3. Proximal distances (d, H): Examples. It turns out that in most situations, when constructing an IPA for solving the convex problem (P), the proximal distance H induced by d will be a Bregman proximal distance D_h generated by some convex kernel h. In the first part of this section we recall the special features of the Bregman proximal distance. In the second part we consider various types of constraint sets $\overline{C}$ for problem (P). We demonstrate through many examples for the pair (d, H) that many well-known proximal methods, as well as new ones, can be handled through our framework.

3.1. Bregman proximal distances. Let $h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper, lsc, and convex function with $\operatorname{dom} h \subset \overline{C}$ and $\operatorname{dom} \nabla h = C$, strictly convex and continuous on $\operatorname{dom} h$, C^1 on $\operatorname{int} \operatorname{dom} h = C$. Define

$$\begin{aligned} H(x,y) := D_h(x,y) &:= h(x) - [h(y) + \langle \nabla h(y), x - y\rangle] \quad \forall x \in \mathbb{R}^n,\ \forall y \in \operatorname{dom} \nabla h \\ &= +\infty \quad \text{otherwise}. \end{aligned} \tag{3.1}$$

The function D_h enjoys a remarkable three point identity [16, Lemma 3.1],

$$H(c,a) = H(c,b) + H(b,a) + \langle c - b, \nabla_1 H(b,a)\rangle \quad \forall a, b \in C,\ \forall c \in \operatorname{dom} h. \tag{3.2}$$

This identity plays a central role in the convergence analysis.

To handle the constraint cases C versus $\overline{C}$, we consider two types of kernels h. The first type consists of convex kernel functions h (often called a Bregman function with zone C; see, e.g., [15]) that satisfy the following conditions:

(B_1) $\operatorname{dom} h = \overline{C}$;
(B_2) (i) $\forall x \in \overline{C}$, $D_h(x, \cdot)$ is level bounded on $\operatorname{int}(\operatorname{dom} h)$;
(ii) $\forall y \in C$, $D_h(\cdot, y)$ is level bounded;
(B_3) $\forall y \in \operatorname{dom} h$, $\forall \{y_k\} \subset \operatorname{int}(\operatorname{dom} h)$ with $\lim_{k\to\infty} y_k = y$, one has $\lim_{k\to\infty} D_h(y, y_k) = 0$;
(B_4) if $\{y_k\}$ is a bounded sequence in $\operatorname{int}(\operatorname{dom} h)$ and $y \in \operatorname{dom} h$ such that $\lim_{k\to\infty} D_h(y, y_k) = 0$, then $y = \lim_{k\to\infty} y_k$.

Note that (B_4) is a direct consequence of the first three properties, a fact proved by Kiwiel in [29, Lemma 2.16].

Let $\mathcal{B}$ be the class of kernels h satisfying properties (B_1)–(B_4). More general Bregman proximal distances such as those introduced in [29] could also be candidates. For the sake of simplicity we consider here only the case $h \in \mathcal{B}$.

For the second type of kernels, we require the convex kernel h to satisfy two (weaker)[1] conditions:

(WB_1) $\operatorname{dom} h = C$;
(WB_2) (i) $\forall x \in C$, $D_h(x, \cdot)$ is level bounded on C;
(ii) $\forall y \in C$, $D_h(\cdot, y)$ is level bounded.

We denote by $\mathcal{WB}$ the set of such convex kernels h.

We give here some examples that underline the difference between the classes $\mathcal{B}$ and $\mathcal{WB}$.

Example 3.1. Let $C = \mathbb{R}^n_{++}$. Separable Bregman proximal distances are the most commonly used in the literature. Let $\theta : \mathbb{R} \to \mathbb{R} \cup +\infty$ be a proper convex and lsc function with $(0, +\infty) \subset \operatorname{dom} \theta \subset [0, +\infty)$ and such that $\theta \in C^2(0, +\infty)$, $\theta''(t) > 0$

[1] The terminology "weaker" is used here to indicate that "weaker type" of convergence results can be derived for this class. Indeed, with $h \in \mathcal{WB}$ one has $(d, D_h) \in \mathcal{F}(C)$ and only Theorem 2.1 (except (iii)(b)) can be applied.

$\forall t > 0$, and $\lim_{t\to 0^+} \theta'(t) = -\infty$. We denote this class by Θ_0 if $\theta(0) < +\infty$ and by Θ_+ whenever $\theta(0) = +\infty$ and θ is also assumed nonincreasing. Given θ in either class, define $h(x) = \sum_{j=1}^n \theta(x_j)$ so that D_h is separable. The first two examples are functions $\theta \in \Theta_0$, i.e., with $\operatorname{dom}\theta = [0, +\infty)$, and the last two are in Θ_+, i.e., with $\operatorname{dom}\theta = (0, +\infty)$:

- $\theta_1(t) = t \log t$ (Shannon entropy),
- $\theta_2(t) = (pt - t^p)/(1 - p)$ with $p \in (0, 1)$,
- $\theta_3(t) = -\log t$ (Burg entropy),
- $\theta_4(t) = t^{-1}$.

More examples can be found in, e.g., [29, 43]. Then, the corresponding proximal distances $D_{h_1}, D_{h_2} \in \mathcal{B}$, while $D_{h_3}, D_{h_4} \in \mathcal{WB}$.

3.2. Self-proximal methods. The three point identity (3.2) plays a fundamental role in the convergence of Bregman-based self-proximal methods, namely those for which we take d itself as a Bregman proximal distance, that is, $d(x, y) = H(x, y) = D_h(x, y)$, with D_h as defined in (3.1). Whenever $h \in \mathcal{B}$, or in $\mathcal{WB}$, properties (P_1), (P_2), and (P_3) hold for $d = D_h$.

Clearly, $D_h(a, a) = 0\ \forall a \in C$, so that (P_4) holds, and since H is always nonnegative it follows from (3.2) that (2.6) holds. Therefore for $h \in \mathcal{WB}$ one has $(d, H) = (D_h, D_h) \in \mathcal{F}(C)$, while if $h \in \mathcal{B}$, then $(d, H) = (D_h, D_h) \in \mathcal{F}_+(\overline{C})$.

When $C = \mathbb{R}^n$, with $h(\cdot) = \|\cdot\|^2/2 \in \mathcal{B}$, then $D_h(x, y) = \|x - y\|^2/2$, and with $(d, H) = (D_h, D_h) \in \mathcal{F}_+(\mathbb{R}^n)$, the IPA is exactly the classical proximal method and Theorems 2.1 and 2.2 cover the usual convergence results, e.g., [24, 30, 31].

We now list several interesting special cases for the pair (d, H) leading to self-proximal schemes for various types of constraints.

Nonnegative constraints. Let $C = \mathbb{R}^n_{++}$ and $\overline{C} = \mathbb{R}^n_+$. For the examples given in Example 3.1, the resulting self-proximal algorithms, namely with $d = H = D_{h_i}$, yield $(d, D_{h_i}) \in \mathcal{F}_+(\overline{C})$ for $i = 1, 2$ and $(d, D_{h_i}) \in \mathcal{F}(C)$ for $i = 3, 4$.

Semidefinite constraints. We denote by S^n the linear space of symmetric real matrices equipped with the trace inner product $\langle x, y\rangle := \operatorname{tr}(xy)$ and $\|x\| = \sqrt{\operatorname{tr}(x^2)}$ $\forall x, y \in S^n$, where $\operatorname{tr}(x)$ is the trace of the matrix x and $\det x$ its determinant. The cone of $n \times n$ symmetric positive semidefinite (positive definite) matrices is denoted by S^n_+ (S^n_{++}). Let $C = S^n_{++}$ and $\overline{C} = S^n_+$. Let $h_1 : S^n_+ \to \mathbb{R}$, $h_1(x) = \operatorname{tr}(x \log x)$ and $h_3 : S^n_{++} \to \mathbb{R}$, $h_3(x) = -\operatorname{tr}(\log x) = -\log\det(x)$ (which corresponds to θ_1 and θ_3, respectively, of Example 3.1). For any $y \in S^n_{++}$, let

$$\begin{aligned} d_1(x, y) &= \operatorname{tr}(x \log x - x \log y + y - x) \text{ with } \operatorname{dom} d_1(\cdot, y) = S^n_+,\\ d_3(x, y) &= \operatorname{tr}(-\log x + \log y + xy^{-1}) - n\\ &= -\log\det(xy^{-1}) + \operatorname{tr}(xy^{-1}) - n \text{ with } \operatorname{dom} d_3(\cdot, y) = S^n_{++}. \end{aligned}$$

The proximal distances d_1, d_3 are Bregman type corresponding to h_1, h_3, respectively, and were proposed by Doljansky and Teboulle in [18], who derived convergence results for the associated IPA. From the results of [18] it is easy to see that $d_i \in \mathcal{D}(C)$, $i = 1, 3$, and with $H(x, y) = d_i(x, y)$ it follows that $(d_1, H) \in \mathcal{F}(S^n_+)$ and $(d_3, H) \in \mathcal{F}(S^n_{++})$ so that we recover the convergence results of [18] through Theorem 2.1. However, as noticed in a counterexample [18, Example 4.1], property (B_3) does not hold even for d_1, and therefore $(d_i, H) \notin \mathcal{F}_+(\overline{C})$, $i = 1, 3$. Consequently, Theorem 2.2 does not apply, i.e., global convergence to an optimal solution cannot be guaranteed. Similar results can be easily extended to the more general case with $C = \{x \in \mathbb{R}^m \mid B(x) \in S^n_{++}\}$ assumed nonempty, with $B(x) = \sum_{i=1}^m x_i B_i - B_0$, where

$B_i \in S^n\ \forall i = 0, 1, \dots, m$, and the map $x \to \sum_{i=1}^m x_i B_i$ being onto, by considering the corresponding proximal distances,

$$D_1(x, y) = d_1(B(x), B(y)), \qquad D_3(x, y) = d_3(B(x), B(y)).$$

Convex programming. Let $f_i : \mathbb{R}^n \to \mathbb{R}$ be concave and C^1 on $\mathbb{R}^n$ for each $i \in [1, m]$. We suppose that Slater's condition holds, i.e., there exists some point $x_0 \in \mathbb{R}^n$ such that $f_i(x_0) > 0\ \forall i \in [1, m]$ and that the open convex set C is described by

$$C = \{x \in \mathbb{R}^n \mid f_i(x) > 0\ \forall i = 1, \dots, m\}$$

so that by Slater's assumption $C \neq \emptyset$ and $\overline{C} = \{x \in \mathbb{R}^n \mid f_i(x) \geq 0,\ i \in [1, m]\}$. Consider the class Θ_+ of functions defined in Example 3.1, and for each $\theta \in \Theta_+$ let

$$h(x) = \begin{cases} \sum_{i=1}^m \theta(f_i(x)) & \text{if } x \in C, \\ +\infty & \text{otherwise.} \end{cases} \tag{3.3}$$

Obviously h is a proper, lsc, and convex function. Now, consider the Bregman proximal distance associated with $h_\nu(x) := h(x) + \frac{\nu}{2}\|x\|^2$ with $\nu > 0$. Then, we take $d(x, y) = D_{h_\nu}(x, y)$, where D_{h_ν} is the Bregman distance associated with h_ν. Thanks to the condition $\nu > 0$, it follows that $h_\nu \in \mathcal{WB}$ and $(d, D_{h_\nu}) \in \mathcal{F}(C)$. An important and interesting case is obtained by choosing the Burg function, $\theta_3(t) = -\log t$. In this case we obtain the following:

$$d(x, y) = \sum_{i=1}^m -\log\frac{f_i(x)}{f_i(y)} + \frac{\langle \nabla f_i(y), x - y\rangle}{f_i(y)} + \frac{\nu}{2}\|x - y\|^2. \tag{3.4}$$

Note that in this case the function $d(\cdot, y)$ enjoys other interesting properties: for example, when the functions f_i are concave quadratic, then $d(\cdot, y)$ is self-concordant for each $y \in C$, a property which is very useful when minimizing the function with Newton-type methods [33]. When $\nu = 0$, i.e., with $d = D_h$, such proximal distance has been recently introduced by Alvarez, Bolte, and Brahic [1], in the context of dynamical systems to study interior gradient flows, but it requires a nondegeneracy condition, $\forall x \in C$: $\operatorname{span}\{\nabla f_i(x) \mid i = 1, \dots, m\} = \mathbb{R}^n$, which is satisfied mostly in the polyhedral case. Here, in the context of proximal methods, the addition of the regularized term in D_h precludes the use of such a condition.

Second order cone constraints. Let $C = L^n_{++} := \{x \in \mathbb{R}^n \mid x_n > (x_1^2 + \cdots + x_{n-1}^2)^{1/2}\}$ be the interior of the Lorentz cone, with closure denoted by L^n_+. Let J_n be a diagonal matrix with its first $(n-1)$ entries being -1 and the last being 1, and define $h : L^n_{++} \to \mathbb{R}$ by $h(x) = -\log(x^T J_n x)$. Then h is proper, lsc, and convex on $\operatorname{dom} h = L^n_{++}$. Let $h_\nu(x) = h(x) + \nu\|x\|^2/2$. Then thanks to $\nu > 0$, one has $h_\nu \in \mathcal{WB}$, and the Bregman proximal distance associated with h_ν is given by

$$D_{h_\nu}(x, y) = -\log\frac{x^T J_n x}{y^T J_n y} + \frac{2x^T J_n y}{y^T J_n y} - 2 + \frac{\nu}{2}\|x - y\|^2, \tag{3.5}$$

and we have $(D_{h_\nu}, D_{h_\nu}) \in \mathcal{F}(L^n_{++})$. As in the convex and semidefinite programming cases, one can easily handle the more general case with a nonempty $C = \{x \in \mathbb{R}^n \mid Ax - b \in L^m_{++}\}$, where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, by choosing $h(x) := -\log(Ax - b)^T J_m (Ax - b)$.

We will now show that, interestingly, even for IPA which are not self-proximal, the induced proximal distance H from the choice of d for various types of constraints will still be a Bregman proximal distance D_h with an appropriate convex kernel h in the class $\mathcal{B}$ or $\mathcal{WB}$.

3.3. Proximal functions based on φ-divergences.

φ-divergence kernels. Let $\varphi : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ be an lsc, convex, proper function such that $\operatorname{dom}\varphi \subset \mathbb{R}_+$ and $\operatorname{dom}\partial\varphi = \mathbb{R}_{++}$. We suppose in addition that φ is C^2, strictly convex, and nonnegative on $\mathbb{R}_{++}$ with $\varphi(1) = \varphi'(1) = 0$. We denote by Φ the class of such kernels and by Φ_1 the subclass of these kernels satisfying

$$\varphi''(1)\left(1 - \frac{1}{t}\right) \le \varphi'(t) \le \varphi''(1)\log t \quad \forall t > 0. \tag{3.6}$$

The other subclass of Φ of interest is denoted by Φ_2, where (3.6) is replaced by

$$\varphi''(1)\left(1 - \frac{1}{t}\right) \le \varphi'(t) \le \varphi''(1)(t-1) \quad \forall t > 0. \tag{3.7}$$

Examples of functions in Φ_1, Φ_2 are (see, e.g., [5, 44])

$$\begin{aligned}
&\varphi_1(t) = t\log t - t + 1, \quad \operatorname{dom}\varphi = [0, +\infty),\\
&\varphi_2(t) = -\log t + t - 1, \quad \operatorname{dom}\varphi = (0, +\infty),\\
&\varphi_3(t) = 2(\sqrt{t} - 1)^2, \quad \operatorname{dom}\varphi = [0, +\infty).
\end{aligned}$$

Corresponding to the classes Φ_r, with $r = 1, 2$, we define a φ-divergence proximal distance by

$$d_\varphi(x, y) = \sum_{i=1}^n y_i^r \varphi\left(\frac{x_i}{y_i}\right).$$

For any $\varphi \in \Phi$, since $\operatorname{argmin}\{\varphi(t) \mid t \in \mathbb{R}\} = \{1\}$, φ is coercive and thus it follows that $d_\varphi \in \mathcal{D}(C)$, with $C = \mathbb{R}^n_{++}$.

The use of φ-divergence proximal distances is particularly suitable for handling polyhedral constraints. Let $C = \{x \in \mathbb{R}^n \mid Ax < b\}$, where A is an (m, n) matrix of full rank m $(m \ge n)$. Particularly important cases include $C = \mathbb{R}^n_{++}$ or $C = \{x \in \mathbb{R}^n \mid a_i < x_i < b_i \ \forall i = 1, \dots, n\}$, with $a_i, b_i \in \mathbb{R}$. For the sake of simplicity we consider here only the case where $C = \mathbb{R}^n_{++}$. Indeed, as is already noted in several works (e.g., [3, 4, 44]), since these proximal distances are separable they can thus be extended without difficulty to the polyhedral case by redefining d in the form $d(x, y) := \sum_{i=1}^m d_i(b_i - \langle a_i, x\rangle, b_i - \langle a_i, y\rangle)$, where a_i are the rows of the matrix A and $d_i(u_i, v_i) = v_i^r \varphi(u_i v_i^{-1})$.

The class Φ_1. It turns out that the induced proximal distance H associated with d_φ is the Bregman proximal distance obtained from the kernel $h(x) = \sum_{j=1}^n x_j \log x_j$ (obtained from θ_1) and given by

$$D_h(x, y) := K(x, y) = \sum_{j=1}^n x_j \log \frac{x_j}{y_j} + y_j - x_j \quad \forall x \in \mathbb{R}^n_+, \ \forall y \in \mathbb{R}^n_{++}, \tag{3.8}$$

which is the Kullback–Liebler relative entropy. The fact that K plays a central role in the analysis of IPA based on $\varphi \in \Phi_1$ was already realized in [28] and later formalized in [44, Lemma 4.1(ii)], which shows that for any $\varphi \in \Phi_1$ one has

$$\langle c - b, \nabla_1 d_\varphi(b, a)\rangle \leq \varphi''(1)[K(c, a) - K(c, b)]. \tag{3.9}$$

Therefore (2.6) is verified for $H = \varphi''(1)K$ and it follows that for any $\varphi \in \Phi_1$ one has $(d_\varphi, \varphi''(1)K) \in \mathcal{F}_+(\overline{C})$ and all the convergence results of section 2 apply for the corresponding IPA. We note parenthetically that the induced proximal distance K can also be obtained from the φ-divergence with the kernel φ_1. In fact this should not be surprising, since it can be verified that $d_\varphi = D_h$ if and only if $h(x) = \sum_{j=1}^n \varphi_1(x_j)$.

Regularized class $\mathbf{\Phi_1}$. Let $\varphi \in \Phi_1$ and define $d(x, y) = d_\varphi(x, y) + 2^{-1}\nu\|x-y\|^2$, with $\nu > 0$. This proximal distance was recently considered in [2] in the context of Lotka–Volterra dynamical systems with the choice $\varphi = \varphi_2$. As shown there, one can verify that with $H(x, y) = K(x, y) + \frac{\nu}{2}\|x - y\|^2$, one has $(d_{\varphi_2}, H) \in \mathcal{F}_+(\overline{C})$.

The class $\mathbf{\Phi_2}$: Second order homogeneous proximal distances [5, 10, 45]**.** Let $\varphi(t) = \mu p(t) + \frac{\nu}{2}(t - 1)^2$ with $\nu \geq \mu > 0$, $p \in \Phi_2$, and let the associated proximal distance be defined by

$$d_\varphi(x, y) = \sum_{j=1}^n y_j^2 \varphi\left(\frac{x_j}{y_j}\right).$$

In particular, $p(t) = -\log t + t - 1$ gives the so-called logarithmic-quadratic proximal distance [5]. Obviously $d_\varphi \in \mathcal{D}(C)$, and from the key inequality [4, Lemma 3.4] one has

$$\langle c - b, \nabla_1 d(b, a)\rangle \leq \eta(\|c - a\|^2 - \|c - b\|^2) \quad \forall a, b \in \mathbb{R}_{++},\ \forall c \in \mathbb{R}_+$$

with $\eta = 2^{-1}(\mu + \nu)$. Therefore with $H(x, y) = \eta\|x - y\|^2$ it follows that $(d_\varphi, H) \in \mathcal{F}_+(\overline{C})$.

4. Interior gradient methods.

When $\overline{C} = \mathbb{R}^n$, Correa and Lemarechal [17] and Robinson [38] have remarked that the PA can be viewed as an ϵ-subgradient descent method. This idea was recently extended by Auslender and Teboulle [7] for the logarithmic-quadratic proximal method which allows us to handle linear inequality constraints directly. Given the framework developed in section 2, we extend these results for more general constraints and with various classes of proximal distances.

We first give the main convergence result. We then present applications and examples which allow us to improve some known interior gradient–based methods as well as to derive new and simple convergent algorithms for conic optimization problems.

4.1. A general convergence theorem.

To solve problem (P) $\inf\{f(x) \mid x \in \overline{C}\}$ we consider the following general projected subgradient-based algorithm (PSA).

Take $d \in \mathcal{D}(C)$. Let $\lambda_k > 0$, $\epsilon_k \geq 0$, and $m \in (0, 1]$, and for $k \geq 1$ generate the sequence $\{x^k, g^k\}$ such that

$$x^{k-1} \in C, \quad g^{k-1} \in \partial_{\varepsilon_k} f(x^{k-1}), \tag{4.1}$$

$$x^k \in \operatorname{argmin}\{\lambda_k\langle g^{k-1}, x\rangle + d(x, x^{k-1}) \mid x \in C\}, \tag{4.2}$$

$$f(x^k) \leq f(x^{k-1}) + m(\langle g^{k-1}, x^k - x^{k-1}\rangle - \epsilon_k). \tag{4.3}$$

Let us briefly recall why the sequence $\{x^k\}$, constructed by the exact IPA ($\epsilon_k = 0$) in section 2 via (2.3) and (2.4), fits in PSA (see, e.g., [17, 7] for more details). Starting

IPA with $x^0 \in C$, one has $x^k \in C$, and it can be verified that $g^k \in \partial f(x^k)$ is equivalent to saying that

$$g^k \in \partial_{\epsilon_k^*} f(x^{k-1}) \quad \text{with } \epsilon_k^* = f(x^{k-1}) - f(x^k) + \langle g^k, x^k - x^{k-1} \rangle \geq 0.$$

Therefore (2.3) and (2.4) are nothing else but (4.1) and (4.2). Then, with $m = 1$ and with ϵ_k^* as defined above, inequality (4.3) holds as an equality, showing that the sequence $\{x^k\}$ generated by IPA satisfies (4.1), (4.2), and (4.3).

Building on the material developed in section 2 it is now possible to establish convergence results of PSA for various instances of the triple $[C, d, H]$, extending recent convergence results given in [7, Theorem 4.1]. Before doing so, we first note that by using the same arguments as in the proof of Proposition 2.1, it is easily seen that the existence of $x^k \in C$ is guaranteed.

THEOREM 4.1. *Let $\{x^k\}$ be a sequence generated by PSA with $(d, H) \in \mathcal{F}(C)$. Set $\sigma_n = \sum_{k=1}^n \lambda_k$ and $\alpha_k = \langle g^{k-1}, x^{k-1} - x^k \rangle$. Then,*

(i) *$\sum_{k=1}^\infty \alpha_k < \infty$, $\sum_{k=1}^\infty \epsilon_k < \infty$, and $\alpha_k \geq \lambda_k^{-1} H(x^k, x^{k-1}) \geq 0 \; \forall k \in \mathbb{N}$.*

(ii) *$\forall z \in C$, $f(x^n) - f(z) \leq \sigma_n^{-1}[H(z, x^0) + \sum_{k=1}^n \lambda_k(\alpha_k + \varepsilon_k)]$.*

(iii) *The sequence $\{f(x^k)\}$ is nonincreasing and converges to f_* as $\sigma_n \to \infty$.*

(iv) *Suppose that the optimal set X_* is nonempty and $\sigma_n \to \infty$. Then the sequence $\{x^k\}$ is bounded with all its limit points in X_* under either one of the following conditions:*

(a) *X_* is bounded.*

(b) *$(d, H) \in \mathcal{F}(\overline{C})$ and $\sum_{k=1}^\infty \lambda_k \epsilon_k < +\infty$ (which in particular is true if $\{\lambda_k\}$ is bounded above).*

In addition, if $(d, H) \in \mathcal{F}_+(\overline{C})$, then $\{x^k\}$ converges to an optimal solution of (P).

Proof. (i) From the optimality conditions, (4.2) is equivalent to

$$\lambda_k g^{k-1} + \nabla_1 d(x^k, x^{k-1}) = 0.$$

Since $H(\cdot, \cdot) \geq 0$ and $H(a, a) = 0$, from (2.6) with $c = a = x^{k-1}$, $b = x^k$ we then obtain

$$\lambda_k \alpha_k = \langle \nabla_1 d(x^k, x^{k-1}), x^k - x^{k-1} \rangle \geq H(x^{k-1}, x^k) \geq 0.$$

Furthermore, from (4.3) we obtain

$$m(\alpha_k + \epsilon_k) \leq f(x^{k-1}) - f(x^k), \tag{4.4}$$

which also shows that $\{f(x^k)\}$ is nonincreasing. Summing over $k = 1, \dots, n$ in the last inequality it follows that

$$m \sum_{k=1}^n (\alpha_k + \epsilon_k) \leq f(x^0) - f(x^n) \leq f(x^0) - f_*, \tag{4.5}$$

proving (i). Now since $\sigma_n = \sum_{k=1}^n \lambda_k$, using $\sigma_k = \lambda_k + \sigma_{k-1}$ (with $\sigma_0 = 0$), multiplying (4.4) by σ_{k-1}, and summing over $k = 1, \dots n$, we obtain

$$\sum_{k=1}^n [(\sigma_k - \lambda_k) f(x^k) - \sigma_{k-1} f(x^{k-1})] \leq 0,$$

which reduces to

$$\sigma_n f(x^n) - \sum_{k=1}^{n} \lambda_k f(x^k) \le 0. \tag{4.6}$$

Now, since $g^{k-1} \in \partial_{\epsilon_k} f(x^{k-1})$, then for any $z \in C$ one has

$$\begin{aligned} f(z) - f(x^{k-1}) + \epsilon_k &\ge \langle g^{k-1}, z - x^{k-1}\rangle \\ &= \langle g^{k-1}, z - x^k\rangle + \langle g^{k-1}, x^k - x^{k-1}\rangle \\ &= -\frac{1}{\lambda_k}\langle z - x^k, \nabla_1 d(x^k, x^{k-1})\rangle - \alpha_k \\ &\ge \frac{1}{\lambda_k}[H(z, x^k) - H(z, x^{k-1})] - \alpha_k, \end{aligned}$$

where the last inequality uses (2.6) with $b = x^k$, $a = x^{k-1}$. Since $f(x^k) \le f(x^{k-1})$, it then follows that

$$\lambda_k(f(x^k) - f(z)) \le H(z, x^{k-1}) - H(z, x^k) + \lambda_k(\alpha_k + \varepsilon_k).$$

Summing the above inequality over $k = 1, \dots, n$, we obtain

$$-\sigma_n f(z) + \sum_{k=1}^{n} \lambda_k f(x^k) \le H(z, x^0) - H(z, x^n) + \sum_{k=1}^{n} \lambda_k(\alpha_k + \epsilon_k).$$

Adding this inequality to (4.6) and dividing by σ_n one obtains

$$f(x^n) - f(z) \le \frac{H(z, x^0)}{\sigma_n} + \sum_{k=1}^{n} \frac{\lambda_k(\alpha_k + \epsilon_k)}{\sigma_n} \quad \forall z \in C.$$

This proves (ii). Suppose $\sigma_n \to \infty$. Since the sequences $\{\alpha_k\}$ and $\{\epsilon_k\}$ converge to 0, invoking Lemma 2.2 and passing to the limit we obtain

$$\lim_{n\to\infty} f(x^n) = \limsup_{n\to\infty} f(x^n) \le \inf\{f(x) \mid x \in C\} = f_*,$$

proving (iii). The rest of the proof is exactly the same as in the proof of Theorems 2.1 and 2.2. □

Using (ii) of Theorem 4.1 with (4.5), we obtain the following corollary.

COROLLARY 4.1. *Let* $(d, H) \in \mathcal{F}(\overline{C})$ *and let* $\{x^k\}$ *be the sequence produced by PSA. Suppose that* $X_* \ne \emptyset$ *and* $0 < \lambda_* \le \lambda_k \le \lambda^*$. *Then we have the global estimation* $f(x^n) - f_* = O(n^{-1})$.

4.2. Conic optimization: Interior projected gradient methods with strongly convex proximal distance.

4.2.1. Preliminaries. We consider now the problem

$$\text{(M)} \qquad \inf\{f(x) \mid x \in \overline{C} \cap \mathcal{V}\},$$

where $\mathcal{V} = \{x : Ax = b\}$, with $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m\times n}$, $n \ge m$, $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is convex and lsc, and we assume that $\exists x^0 \in \operatorname{dom} f \cap C : Ax^0 = b$.

When $\overline{C}$ is a convex cone, problem (M) is the standard conic optimization problem (see, e.g., [33]), while whenever $\mathcal{V} = \mathbb{R}^n$ it is just a pure conic optimization problem.

In the following subsection, we assume also that f is continuously differentiable with ∇f Lipschitz on $C \cap \mathcal{V}$ and Lipschitz constant L, i.e., $\exists L > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \quad \forall x, y \in C \cap \mathcal{V}. \tag{4.7}$$

We consider now $(d, H) \in \mathcal{F}(C)$ such that d satisfies the following properties:

(s1) $\exists \sigma > 0 : \forall y \in C \cap \mathcal{V}$, $d(\cdot, y)$ is σ-strongly convex over $C \cap \mathcal{V}$, i.e.,

$$\langle \nabla_1 d(x_1, y) - \nabla_1 d(x_2, y), x_1 - x_2 \rangle \geq \sigma \|x_1 - x_2\|^2 \quad \forall x_1, x_2 \in C \cap \mathcal{V}, \tag{4.8}$$

for some norm $\|\cdot\|$ in $\mathbb{R}^n$.

(s2) $\forall y \in C \cap \mathcal{V}$, $d(\cdot, y)$ is C^2 on C with Hessian function denoted by $\nabla_1^2 d(\cdot, y)$.

Therefore with the same arguments as the ones given in the proof of Proposition 2.1, it follows that for each $x \in C \cap \mathcal{V}$, for each $v \in \mathbb{R}^n$ there exists a unique (by strong convexity) point $u(v, x) \in C \cap \mathcal{V}$ solving

$$u(v, x) = \operatorname{argmin}\{\langle v, z \rangle + d(z, x) \mid z \in \mathcal{V}\}. \tag{4.9}$$

Then from the optimality conditions for the convex problem (4.9) (see, e.g., [39, section 28]), $\exists \mu := \mu(v, x) \in \mathbb{R}^m$ such that[2]

$$v + A^t \mu + \nabla_1 d(u(v, x), x) = 0, \quad Au(v, x) = b. \tag{4.10}$$

Clearly, problem (M) can be equivalently formulated in the form of problem (P) as follows:

$$f_* = \min\{f_0(x) \mid x \in \overline{C}\} \quad \text{with} \quad f_0 = f + \delta_{\mathcal{V}}.$$

Define $\mathcal{V}_0 = \{x : Ax = 0\}$. Note that for any $w \in \mathcal{V}$ one has $f(w) = f_0(w)$ and

$$\gamma(\eta, x) := (\nabla f(x) + A^t \eta) \in \partial f_0(x) \quad \forall x \in C \cap \mathcal{V},\ \forall \eta \in \mathbb{R}^m. \tag{4.11}$$

Indeed, for any $z, x \in \mathcal{V}$ we have $z - x \in \mathcal{V}_0$ and thus, for any $\eta \in \mathbb{R}^m$,

$$\begin{aligned} f_0(z) = f(z) \geq f(x) + \langle \nabla f(x), z - x \rangle &= f_0(x) + \langle \nabla f(x) + A^t \eta, z - x \rangle \\ &= f_0(x) + \langle \gamma(\eta, x), z - x \rangle. \end{aligned}$$

Since for $z \notin \mathcal{V}$ this inequality obviously holds, (4.11) is verified.

4.2.2. Algorithms. We can now propose for solving problem (M) the basic iteration of our algorithm. Given a step-size rule for choosing λ_k at each step k, starting from a point $x^0 \in C \cap \mathcal{V}$ we generate iteratively the sequence $x^k \in C \cap \mathcal{V}$ by the relation

$$x^k = u(\lambda_k \nabla f(x^{k-1}), x^{k-1}). \tag{4.12}$$

As a consequence of the above discussion relations (4.1) and (4.2) are satisfied with f replaced by f_0, $\epsilon_k = 0$, and

$$g^{k-1} = \gamma\left(\frac{\mu(\lambda_k \nabla f(x^{k-1}), x^{k-1})}{\lambda_k}, x^{k-1}\right) \in \partial f_0(x^{k-1}). \tag{4.13}$$

[2] Note that the first relation in the optimality condition can be rewritten equivalently as $v + \nabla_1 d(u(v, x), x) \in \mathcal{V}_0^{\perp}$, where $\mathcal{V}_0 = \{x : Ax = 0\}$.

We propose now two step-size rules, and for each rule we will show that inequality (4.3) holds and $\sum_{k=1}^{\infty} \lambda_k = \infty$. As a consequence we will be able to apply Theorem 4.1 and then to devise two convergent interior gradient projection algorithms which naturally extend the results of Auslender and Teboulle [7].

Algorithm 1 (constant step-size rule). Let $\epsilon \in \,]0,1[$ and set $\lambda^* := 2\epsilon\sigma L^{-1}$, $\lambda_* \in (0, \lambda^*)$. Start from a point $x^0 \in C \cap \mathcal{V}$ and generate the sequence $\{x^k\} \in C \cap \mathcal{V}$ as follows: if $\nabla f(x^{k-1}) \in \mathcal{V}_0^{\perp}$, stop. Otherwise, compute

$$x^k = x^k(\lambda_k) := u(\lambda_k \nabla f(x^{k-1}), x^{k-1}) \quad \text{with } \lambda_k \in (\lambda_*, \lambda^*]. \tag{4.14}$$

Theorem 4.2. *Let $\{x^k\}$ be the sequence produced by Algorithm* 1. *If at step k one has $\nabla f(x^{k-1}) \in \mathcal{V}_0^{\perp}$, then x^{k-1} is an optimal solution. Otherwise, the sequence $\{f(x^k)\}$ is nonincreasing and converges to f_*. Moreover, suppose that the optimal set X_* is nonempty; then*

(a) *if X_* is bounded, the sequence $\{x^k\}$ is bounded with all its limit points in X_*;*

(b) *if $(d, H) \in \mathcal{F}_+(\overline{C})$, the sequence $\{x^k\}$ converges to an optimal solution of* (P).

Proof. First, if $\nabla f(x^{k-1}) \in \mathcal{V}_0^{\perp}$, since $x^{k-1} \in C \cap \mathcal{V}$ then obviously, from the optimality conditions (4.10), it follows that x^{k-1} is also an optimal solution. Suppose now that $\nabla f(x^{k-1}) \notin \mathcal{V}_0^{\perp}$. Since $\lambda_k \geq \lambda_*$, then $\sigma_n = \sum_{k=1}^{n} \lambda_k \to \infty$. Thus, it remains to show (4.3), and our result would follow as a direct consequence of Theorem 4.1. Since ∇f is Lipschitz, by the well-known descent lemma (see, e.g., [12, p. 667]) one has

$$f(x^k) \leq f(x^{k-1}) + \langle \nabla f(x^{k-1}), x^k - x^{k-1} \rangle + \frac{L}{2}\|x^k - x^{k-1}\|^2. \tag{4.15}$$

Now, we first remark that

$$(x^k - x^{k-1}) \in \mathcal{V}_0. \tag{4.16}$$

Then using (4.8), with $x_1 = y = x^{k-1} \in C \cap \mathcal{V}$, $x_2 = u(v, x^{k-1}) \in C \cap \mathcal{V}$, and $v = \lambda_k \nabla f(x^{k-1})$; (4.10); and g^{k-1} as defined in (4.13) (recalling that $\nabla_1 d(y, y) = 0$), it follows that

$$\lambda_k \langle g^{k-1}, x^{k-1} - x^k \rangle = \lambda_k \langle \nabla f(x^{k-1}), x^{k-1} - x^k \rangle \geq \sigma \|x^k - x^{k-1}\|^2.$$

This combined with (4.15) yields

$$f(x^k) \leq f(x^{k-1}) + \langle x^k - x^{k-1}, g^{k-1} \rangle \left(1 - \frac{L\lambda_k}{2\sigma}\right),$$

so that with $f_0(x^k) = f(x^k)$, $f_0(x^{k-1}) = f(x^{k-1})$ we get

$$f_0(x^k) \leq f_0(x^{k-1}) + \langle x^k - x^{k-1}, g^{k-1} \rangle \left(1 - \frac{L\lambda_k}{2\sigma}\right).$$

Then with $\lambda^* = \frac{2\epsilon\sigma}{L}$, we get $f_0(x^k) \leq f_0(x^{k-1}) + \langle x^k - x^{k-1}, g^{k-1} \rangle (1 - \epsilon)$, showing that (4.3) holds with $m = 1 - \epsilon$. □

The second algorithm extends the method proposed in [7] and allows us to use a generalized step-size rule, reminiscent of the one used in the classical projected gradient method as studied by Bertsekas [11].

Algorithm 2 (Armijo–Goldstein step-size rule). Let $\beta \in (0, 1)$, $m \in (0, 1)$, and $s > 0$ be fixed chosen scalars. Start from a point $x^0 \in C \cap \mathcal{V}$ and generate the sequence $\{x^k\} \in C \cap \mathcal{V}$ as follows: if $\nabla f(x^{k-1}) \in \mathcal{V}_0^\perp$ stop. Otherwise, with $x^k(\lambda) = u(\lambda \nabla f(x^{k-1}), x^{k-1})$, set $\lambda_k = \beta^{j_k} s$, where j_k is the first nonnegative integer j such that

$$f(x^k(\beta^j s)) - f(x^{k-1}) \le m\langle \nabla f(x^{k-1}), x^k(\beta^j s) - x^{k-1}\rangle. \tag{4.17}$$

Then set $x^k = x^k(\lambda_k)$.

In order to show that this step-size rule is well defined, we need the following proposition.

PROPOSITION 4.1. *For any $x \in C \cap \mathcal{V}$, any $v \in \mathbb{R}^n$, and $\lambda > 0$, the unique solution $u(\lambda v, x)$ defined by* (4.9) *satisfies $u(0, x) = x$ and the following properties hold:*

(i) $\sigma \|x - u(\lambda v, x)\|^2 \le \lambda \langle x - u(\lambda v, x), v\rangle$,

(ii) $\frac{\|u(\lambda v,x)-x\|}{\lambda} \le \sigma^{-1}\|v\|$,

(iii) $\lim_{\lambda\to 0^+} \frac{u(\lambda v,x)-x}{\lambda}$ *exists and is equal to $\rho(v, x) = u$, where $u \in \mathcal{V}_0$ satisfies*

$$Q(x)u + v \in \mathcal{V}_0^\perp \tag{4.18}$$

with $Q(x) = \nabla_1^2 d(x, x)$,

(iv) $\langle -\rho(v, x), v\rangle \ge \sigma \|\rho(v, x)\|^2$.

Proof. Fix any $x \in C \cap \mathcal{V}$. By (4.9), we have $u(0, x) = \operatorname{argmin}\{d(z, x) \mid z \in \mathcal{V}\}$, and thus by optimality conditions (4.10) with $\mu = 0$ it follows that $u(0, x) = x$. Furthermore, from (4.10) we have

$$\langle \lambda v + \nabla_1 d(u(\lambda v, x), x), x - u(\lambda v, x)\rangle = 0,$$

from which the inequality in (i) follows immediately by using the strong convexity inequality (4.8) at $y = x_1 = x$, $x_2 = u(\lambda v, x)$, and (ii) follows from (i) and the Cauchy–Schwarz inequality.

(iii) Since $d(\cdot, y)$ is strongly convex on $C \cap \mathcal{V}$ it follows from (4.8) that

$$\langle Q(x)h, h\rangle \ge \sigma \|h\|^2 \quad \forall h \in \mathcal{V}_0. \tag{4.19}$$

As a consequence of the Lax–Milgram theorem (see, for example, [14, Corollary 5.8]), (4.18) admits exactly one solution $\rho(v, x)$. Note that $\nabla_1 d(x, x) = 0$. Then, since by (4.10) we have

$$\lambda v + \nabla_1 d(u(\lambda v, x), x) + A^t \mu(\lambda v, x) = 0,$$

it follows that

$$\forall h \in \mathcal{V}_0 : \quad \lambda^{-1}\langle \nabla_1 d(u(\lambda v, x), x) - \nabla_1 d(x, x), h\rangle = \langle -v, h\rangle.$$

Denote $s(\lambda) := \frac{u(\lambda v,x)-x}{\lambda}$. Now, from (ii) the generalized sequence $\{s(\lambda)\}_{\lambda>0}$ is bounded. Since $u(\lambda v, x) = x + \lambda s(\lambda)$, taking the limit as $\lambda \to 0^+$ in the last equation and using the definition of the derivative (recall that here $d(\cdot, y) \in C^2$ for every $y \in C \cap \mathcal{V}$), it follows that any limit point u of the generalized sequence $\{s(\lambda)\}_{\lambda>0}$ satisfies $u \in \mathcal{V}_0$ such that

$$\langle \nabla_1^2 d(x, x)u, h\rangle = \langle Q(x)u, h\rangle = -\langle v, h\rangle \quad \forall h \in \mathcal{V}_0,$$

which is equivalent to (4.18). As a consequence $u = \rho(v, x)$ and $\lim_{\lambda\to 0^+} s(\lambda)$ exists and is equal to $\rho(v, x)$. To prove (iv), take $h = \rho(v, x) \in \mathcal{V}_0$ in the last equality and use (4.19). □

We can now prove the convergence of Algorithm 2.

THEOREM 4.3. *Let $\{x^k\}$ be the sequence generated by Algorithm 2. If at step k one has $\nabla f(x^{k-1}) \in \mathcal{V}_0^\perp$, then x^{k-1} is an optimal solution. Otherwise, the algorithm is well defined, i.e., there exists an integer j_k such that $\lambda_k = \beta^{j_k}$, and the sequence $\{\lambda_k\}$ is bounded below by $\lambda_* = \min(2\sigma\beta L^{-1}(1-m), s) > 0$. Furthermore, Theorem 4.2 holds for the sequence produced by Algorithm 2.*

Proof. We have only to prove that the algorithm is well defined and that $\lambda_k \geq \lambda_*$ (so that $\lim_{n\to\infty} \sigma_n = +\infty$). Indeed, if we set $\epsilon_k = 0$ and g^{k-1} as given in (4.13), then by definition of Algorithm 2 the sequence $\{x^k\}$ satisfies relations (4.1), (4.2), and (4.3). To simplify the notation set $x := x^{k-1}$, $v = \nabla f(x^{k-1})$, $x(\lambda) = u(\lambda v, x)$. First, if $v \in \mathcal{V}_0^\perp$, since $x \in C \cap \mathcal{V}$, then obviously, from optimality conditions (4.10), it follows that x is also an optimal solution. Suppose now that $v \notin \mathcal{V}_0^\perp$ and that (4.17) does not hold. That is,

$$f(x(\beta^j s)) - f(x) > m\langle x(\beta^j s) - x, v\rangle \quad \forall j \in \mathbb{N}. \tag{4.20}$$

Invoking the mean value theorem, $\exists z_j \in]x, x(\beta^j s)[$ such that

$$\left\langle \nabla f(z_j), \frac{x(\beta^j s) - x}{\beta^j s} \right\rangle > m \left\langle \frac{x(\beta^j s) - x}{\beta^j s}, v \right\rangle \quad \forall j \in \mathbb{N}.$$

But by Proposition 4.1(i) it follows that $\lim_{j\to\infty} z_j = x$. Moreover, passing to the limit in the last inequality and using (iii) and (iv) of the same proposition, we obtain

$$\sigma(1-m)\|\rho(v,x)\|^2 \leq (1-m)\langle -v, \rho(v,x)\rangle \leq 0,$$

which implies that $\rho(v,x) = 0$, and hence by (4.18) it follows that $v \in \mathcal{V}_0^\perp$, and we have reached a contradiction. Now let us prove that $\lambda_k \geq \lambda_*$. As for the case of the constant step-size rule, with the same arguments (using again the descent lemma; cf. (4.15)) we obtain

$$f_0(x^k(\lambda)) - f_0(x^{k-1}) \leq \langle g^{k-1}, x^k(\lambda) - x^{k-1}\rangle \left(1 - \frac{L\lambda}{2\sigma}\right) \quad \forall \lambda > 0,$$

where $x^k(\lambda) = u(\lambda \nabla f(x^{k-1}), x^{k-1})$ so that (4.17) holds for all $j \in \mathbb{N}$ with $\beta^j s \leq 2\sigma L^{-1}(1-m)$. But since by definition if $j_k \neq 0$, $\lambda_k \beta^{-1}$ does not satisfy (4.17), then $\lambda_k \beta^{-1} > 2\sigma L^{-1}(1-m)$, it follows that $\lambda_k \geq \lambda_*$ $\forall k$. From here, we can then proceed with the same statements and conclusions of Theorem 4.2 for Algorithm 2. □

In general $u(v,x)$ is not given explicitly and has to be computed by an algorithm. However, there are important cases where the function d is such that $u(v,x)$ is given explicitly by an analytic formula, making these algorithms particularly attractive, which we now describe.

4.2.3. Application examples.

Consider the functions d with $(d,H) \in \mathcal{F}(C)$ which are regularized distances of the form

$$d(x,y) = p(x,y) + \frac{\sigma}{2}\|x-y\|^2, \tag{4.21}$$

with $p \in \mathcal{D}(C)$.

Note that the log-quad function belongs to this class. Such a class has been recently introduced and studied by Bolte and Teboulle [13] in the context of gradient-like continuous dynamical systems for constrained minimization.

We begin with two pure conic optimization problems, i.e., with $\mathcal{V} = \mathbb{R}^n$. We then consider the semidefinite and second order conic problems. To the best of our knowledge, this leads to the first explicit interior gradient methods for these problems with convergence results. The last application considers convex minimization over the unit simplex.

A. Convex minimization over $\boldsymbol{C = \mathbb{R}^n_{++}}$. Let $\varphi \in \Phi_r$ with $r = 1, 2$ and let d be given by (4.21) with $p(z, x) = \mu \sum_{j=1}^n x_j^r \varphi(x_j^{-1} z_j)$, $\sigma \geq \mu > 0$ for $(z, x) \in C \times C$. Take for example $\varphi(t) = -\log t + t - 1$ and $r = 2$, namely the log-quad function. Then, (4.21) can be written as

$$d(z, x) = \sum_{j=1}^n x_j^2 \omega(x_j^{-1} z_j) \quad \text{with} \quad \omega(t) = \frac{\sigma}{2}(t-1)^2 + \mu(t - \log t - 1).$$

Solving (4.9), one easily obtains (see also [7, eq. (2.3), p. 4]) the following explicit formulas:

$$\begin{gathered} \forall i = j, \ldots, n, \quad u_j(v, x) = x_j (\omega^*)'(-v_j x_j^{-1}) \\ \text{with} \quad (\omega^*)'(s) = (2\sigma)^{-1}\{(\sigma - \mu) + s + \sqrt{((\sigma-\mu)+s)^2 + 4\mu\sigma}\}. \end{gathered}$$

In the case $r = 1$, (4.9) reduces to solve the equation in $z \equiv u(v, x) > 0$ given by

$$v + \mu(1 - x_j z_j^{-1}) + \sigma(z_j - x_j) = 0, \quad j = 1, \ldots, n.$$

A simple calculation then yields the unique positive solution of this quadratic equation:

$$u_j(v, x) = (2\sigma)^{-1}\left[\sigma x_j - \mu - v_j + \sqrt{(\sigma x_j - \mu - v_j)^2 + 4\sigma\mu x_j}\right] \quad \forall j = 1, \ldots, n.$$

B. Semidefinite programming, $\boldsymbol{C = S^n_{++}}$. Take (as in section 3.2)

$$\begin{aligned} p(x, y) &= \operatorname{tr}(-\log x + \log y + xy^{-1}) - n \quad \forall x, y \in S^n_{++} \\ &= +\infty \quad \text{otherwise}, \end{aligned}$$

which is obtained from the Bregman kernel $h : S^n_{++} \to \mathbb{R}$ defined by $h(x) = -\ln \det(x)$. Using the fact that $\nabla h(x) = -x^{-1}$, the optimality conditions for (4.9) allow us to solve for $z \equiv u(v, x)$ the matrix equation

$$\sigma z - z^{-1} = \rho \quad \text{with} \quad \rho := \sigma x - v - x^{-1}.$$

A direct calculation shows that the matrix

$$u(v, x) = (2\sigma)^{-1}(\rho + \sqrt{\rho^2 + 4\sigma I}) \quad \forall x \in S^n_{++}, \ \forall v \in S^n$$

(where I denotes the $n \times n$ identity matrix) is the unique solution of this equation, with $u(v, x) \in S^n_{++}$, since its eigenvalues are positive.

C. Second order cone programming, $\boldsymbol{C = L^n_{++}}$. As in section 3, we take $h_\nu(x) = -\log(x^T J_n x) + \frac{\nu}{2}\|x\|^2$, with J_n a diagonal matrix with its first $(n-1)$ entries being -1 and the last entry being 1. Consider the associated Bregman distance $D \equiv D_{h_\nu}$ (as given by (3.5), with $\nu \equiv 2\sigma > 0$, the multiplication by 2 being just for computational convenience):

$$D(x, y) = -\log \frac{x^T J_n x}{y^T J_n y} + \frac{2x^T J_n y}{y^T J_n y} - 2 + \sigma\|x - y\|^2 \quad \forall x, y \in L^n_{++}.$$

Moreover, we use the following notation. For any $\xi \in \mathbb{R}^n$, we set $\tau(\xi) := \xi^T J_n \xi$ and we write $\xi := (\bar{\xi}, \xi_n) \in \mathbb{R}^{n-1} \times \mathbb{R}$. Writing the optimality conditions for (4.9), we have to find the unique solution $u(v,x) \equiv z \in L^n_{++}$ (namely with $\tau(z) > 0$) solving

$$v + \nabla h(z) - \nabla h(x) + 2\sigma(z - x) = 0. \tag{4.22}$$

Using $\nabla h(z) = -2\tau(z)^{-1} J_n z$ and defining $w := (\nabla h(x) + 2\sigma x - v)/2 := (\bar{w}, w_n) \in \mathbb{R}^{n-1} \times \mathbb{R}$, (4.22) reduces to

$$\sigma z - \tau(z)^{-1} J_n z = w. \tag{4.23}$$

Decomposing (4.23) in the product space $\mathbb{R}^{n-1} \times \mathbb{R}$ yields

$$\sigma \bar{z} + \tau(z)^{-1} \bar{z} = \bar{w}, \quad \sigma z_n - \tau(z)^{-1} z_n = w_n, \tag{4.24}$$

and by eliminating $\tau(z) > 0$ from these last two equations we obtain

$$(2\sigma z_n - w_n)\bar{z} = z_n \bar{w} \iff (2\sigma \bar{z} - \bar{w}) z_n = w_n \bar{z}. \tag{4.25}$$

Now, multiplying (4.23) by z, we obtain $\sigma\|z\|^2 - w^T z - 1 = 0$, which after completing the square can be rewritten as $\|2\sigma\bar{z} - \bar{w}\|^2 + (2\sigma z_n - w_n)^2 = \|w\|^2 + 4\sigma$. Using (4.25) and defining $\zeta := 2\sigma z_n - w_n$, the last equation reads

$$\frac{w_n^2 \|\bar{w}\|^2}{\zeta^2} + \zeta^2 = \|w\|^2 + 4\sigma. \tag{4.26}$$

Now, it is easy to verify that $\zeta > 0$. Indeed, since $z \in L^n_{++}$, then $z_n > 0$, and by (4.24) one also has $w_n < \sigma z_n$, and it follows that $\zeta = 2\sigma z_n - w_n > \sigma z_n - w_n > 0$. Out of the two remaining solutions of (4.26), a direct computation (using the fact that $(\|w\|^2 + 4\sigma)^2 - 4w_n^2\|\bar{w}\|^2 = (w_n^2 - \|\bar{w}\|^2 + 4\sigma)^2 + 16\sigma\|\bar{w}\|^2$) shows that the unique positive solution of (4.26) that will warrant $\tau(z) > 0$ is given by the following:

$$\zeta = \left(\frac{\|w\|^2 + 4\sigma + \sqrt{(\|w\|^2 + 4\sigma)^2 - 4w_n^2\|\bar{w}\|^2}}{2} \right)^{1/2}.$$

Therefore using (4.25) it follows that the unique solution $u \equiv z \in L^n_{++}$ of (4.22) is given by $z = (\bar{z}, z_n)$ with

$$\bar{z} = \frac{z_n}{\zeta}\bar{w} = \frac{1}{2\sigma}\left(1 + \frac{w_n}{\zeta}\right)\bar{w}, \quad z_n = \frac{1}{2\sigma}(w_n + \zeta). \tag{4.27}$$

Remark 4.1. It is worthwhile to mention that an alternative derivation of (4.27) could also have been obtained by using properties and facts on Jordan algebra associated with the second order cone; see, e.g., [22, 23].

D. Convex minimization over the unit simplex. An interesting special case of a conic optimization, with $\mathcal{V} \neq \mathbb{R}^n$, where $u(v,x)$ can be explicitly given, and where all this theory applies, is when $\overline{C} = \mathbb{R}^n_+$ and $A = e^T$, $b = 1$, i.e., $\mathcal{V} = \{x \in \mathbb{R}^n \mid \sum_{j=1}^n x_j = 1\}$, so that problem (M) reduces to a convex minimization problem over the unit simplex $\Delta = \{x \in \mathbb{R}^n \mid \sum_{j=1}^n x_j = 1,\ x \geq 0\}$. This problem arises in important applications. In [9], Ben-Tal, Margalit, and Nemirovski demonstrated that an algorithm based on the mirror descent (MDA) can be successfully used to solve very large-scale instances of computerized tomography problems, modeled through (M). Recently, Beck and Teboulle [8] have shown that the MDA can be viewed as a projection subgradient algorithm with strongly convex Bregman proximal distances. As a result, to handle the simplex constraints Δ, they proposed to use a Bregman proximal distance based on the entropy kernel

$$\psi(x) = \begin{cases} \sum_{j=1}^n x_j \log x_j & \text{if } x \in \Delta, \\ +\infty & \text{otherwise} \end{cases} \tag{4.28}$$

to produce an entropic mirror descent algorithm (EMDA). It was shown in [8] that the EMDA preserved the same computational efficiency as the MDA (grows slowly with the dimension of the problem), but has the advantage of being given explicitly by a simple formula, since the problem

$$u(v,x) = \operatorname*{argmin}_{z \in \Delta} \{\langle v, z\rangle + D_\psi(z,x)\} \tag{4.29}$$

can be easily solved analytically and yields

$$u_j(v,x) = \frac{x_j \exp(-v_j)}{\sum_{i=1}^n x_i \exp(-v_i)}, \quad j = 1, \dots, n. \tag{4.30}$$

The resulting EMDA of [8] was then defined as follows: for each $j = 1, \dots, n$ with $v_j = \frac{\partial f}{\partial x_j}(x^{k-1})$,

$$x_j^k(\lambda_k) = u_j(\lambda_k v, x^{k-1}) = \frac{x_j^{k-1} \exp\left(-\lambda_k \frac{\partial f}{\partial x_j}(x^{k-1})\right)}{\sum_{i=1}^n x_i^{k-1} \exp\left(-\lambda_k \frac{\partial f}{\partial x_i}(x^{k-1})\right)}, \tag{4.31}$$

$$\lambda_k = \frac{\sqrt{2 \log k}}{L_f \sqrt{k}}, \tag{4.32}$$

where the objective function was supposed to be Lipschitz on Δ and L_f is the Lipschitz constant.

We can modify the EMDA with an Armijo–Goldstein step-size rule. Such a version of the EMDA can be more practical, since we do not need to know/compute the constant L_f. Indeed, it is well known (see, e.g., [8]) that

$$\langle \nabla\psi(x) - \nabla\psi(y), x - y\rangle \geq \|x - y\|_1^2 \quad \forall x, y \in \Delta_+ = \left\{ x \in \mathbb{R}^n \;\middle|\; \sum_{j=1}^n x_j = 1,\ x > 0 \right\},$$

namely ψ is 1-strongly convex with respect to the norm $\|\cdot\|_1$, and hence so is $d = H = D_\psi$. Therefore we can apply Theorem 4.3, proving that the sequence $\{x^k\}$ defined by (4.31), and with λ_k defined by the Armijo–Goldstein step-size rule (4.17), converges to an optimal solution of (M).

5. Interior gradient methods with improved efficiency.

In this section, we further analyze the global convergence rate of interior gradient methods, and we propose a new interior scheme which improves their efficiency. The classical gradient method for minimizing a continuously differentiable function over $\mathbb{R}^n$ with Lipschitz gradient is known to exhibit an $O(k^{-1})$ global convergence rate estimate for function values. In [34], Nesterov developed what he called an "optimal algorithm" for smooth convex minimization and was able to improve the efficiency of the gradient method by constructing a method that keeps the simplicity of the gradient method but with the faster rate $O(k^{-2})$. Inspired by this work, it is thus natural to ask if this kind of result can be extended to interior gradient methods. We answer this question positively for a class of interior gradient methods. We propose an algorithm that provides a natural extension of the results of [34] and leads to a simple "optimal" interior gradient method for convex conic problems.

Consider the conic optimization problem as described in section 4.2.1, i.e.,

$$\text{(M)} \qquad \inf\{f(x) : x \in \overline{C} \cap \mathcal{V}\},$$

where $\mathcal{V} := \{x \in \mathbb{R}^n \mid Ax = b\}$, with $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$, $n \geq m$, $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is convex and lsc, and we assume that $\exists x^0 \in \operatorname{dom} f \cap C : Ax^0 = b$. We assume also that f is continuously differentiable with ∇f Lipschitz on $C \cap \mathcal{V}$ and Lipschitz constant L, i.e., satisfying (4.7).

The basic idea is to generate a sequence of functions $\{q_k\}$ that approximate the function f in such a way that at each step $k \geq 0$ the difference $q_k(x) - f(x)$ is reduced by a fraction $(1 - \alpha_k)$, where $\alpha_k \in [0, 1)$, that is,

$$q_{k+1}(x) - f(x) \leq (1 - \alpha_k)(q_k(x) - f(x)) \quad \forall x \in \overline{C} \cap \mathcal{V}. \tag{5.1}$$

Whenever (5.1) holds, we then obtain

$$q_k(x) - f(x) \leq \gamma_k(q_0(x) - f(x)) \quad \forall x \in \overline{C} \cap \mathcal{V}, \tag{5.2}$$

where

$$\gamma_k := \prod_{l=0}^{k-1}(1 - \alpha_l). \tag{5.3}$$

Thus, if at step k we have a sequence $\{x^k\} \in C \cap \mathcal{V}$ such that $f(x^k) \leq \inf_{z \in \overline{C} \cap \mathcal{V}} q_k(z) := q_k^*$, assuming that the optimal solution set X_* of problem (P) is nonempty, we obtain from (5.2) the global convergence rate estimate

$$f(x^k) - f(x^*) \leq \gamma_k(q_0(x^*) - f(x^*)). \tag{5.4}$$

From the latter inequality it follows that if $\gamma_k \to 0$, then the sequence $\{x^k\}$ is a minimizing sequence for f and the convergence rate of $f(x^k)$ to $f(x^*)$ is measured by the magnitude of γ_k. Therefore to construct algorithms based on the above scheme which was proposed in [34] we need

- to generate an appropriate sequence of functions $\{q_k(\cdot)\}$,
- to guarantee that at each iteration k one can guarantee

$$f(x^k) \leq \min_{z \in \overline{C} \cap \mathcal{V}} q_k(z) := q_k^*.$$

We begin by constructing the sequence of functions $\{q_k(\cdot)\}$. For that purpose, we take here $d \equiv H \in \mathcal{D}(C)$, where H is a Bregman proximal distance (cf. (3.1)) with kernel h such that

(h1) $\operatorname{dom} h = \overline{C}$,

(h2) h is σ-strongly convex on $C \cap \mathcal{V}$.

For every $k \geq 0$ and for any $x \in \overline{C} \cap \mathcal{V}$, we construct the sequence $\{q_k(x)\}$ recursively via

$$q_0(x) = f(x^0) + cH(x, x^0), \tag{5.5}$$

$$q_{k+1}(x) = (1 - \alpha_k)q_k(x) + \alpha_k l_k(x, y^k), \tag{5.6}$$

$$l_k(x, y^k) = f(y^k) + \langle x - y^k, \nabla f(y^k)\rangle. \tag{5.7}$$

Here, $c > 0$ and $\alpha_k \in [0, 1)$. The point x^0 is chosen such that $x^0 \in C \cap \mathcal{V}$, while the point $y_k \in C$ is arbitrary and will be generated in a specific way later. We first show that the sequence of functions $\{q_k(\cdot)\}$ satisfies (5.1).

Lemma 5.1. *The sequence* $\{q_k(x)\}$ *defined by* (5.5)–(5.7) *satisfies*

$$q_{k+1}(x) - f(x) \leq (1 - \alpha_k)(q_k(x) - f(x)) \quad \forall x \in \overline{C} \cap \mathcal{V}.$$

Proof. Since f is convex, we have $f(x) \geq l_k(x, y^k)$ $\forall x \in \overline{C} \cap \mathcal{V}$, and together with (5.6) we thus obtain

$$q_{k+1}(x) \leq (1 - \alpha_k)q_k(x) + \alpha_k f(x) \quad \forall x \in \overline{C} \cap \mathcal{V},$$

from which the desired result follows. □

Using the notation of section 4, we recall that for each $z \in C \cap \mathcal{V}$, for each $v \in \mathbb{R}^n$ there exists a unique (by strong convexity of $H(\cdot, z)$) point $u(v, z) \in C \cap \mathcal{V}$ solving

$$u(v, z) = \operatorname{argmin}\{\langle v, x\rangle + H(x, z) \mid x \in \overline{C} \cap \mathcal{V}\}. \tag{5.8}$$

The next result is crucial and shows that the sequence $\{q_k(\cdot)\}$ admits a simple generic form.

Lemma 5.2. *For any* $k \geq 0$*, one has*

$$q_k(x) = q_k^* + c_k H(x, z^k) \quad \forall x \in \overline{C} \cap \mathcal{V} \tag{5.9}$$

with

$$z^k = \operatorname*{argmin}_{x \in \overline{C} \cap \mathcal{V}} q_k(x),\ q_k^* = q_k(z^k),\ c_0 = c,\ z^0 = x^0 \in C \cap \mathcal{V}. \tag{5.10}$$

Furthermore, the sequence $\{z^k\} \in C \cap \mathcal{V}$ *is uniquely defined by*

(5.11)
$$z^{k+1} = \operatorname{argmin}\left\{\left\langle x, \frac{\alpha_k}{c_{k+1}}\nabla f(y^k)\right\rangle + H(x, z^k) \,\middle|\, x \in \overline{C} \cap \mathcal{V}\right\} \equiv u\left(\frac{\alpha_k}{c_{k+1}}\nabla f(y^k), z^k\right),$$

where the positive sequence $\{c_k\}$ *satisfies* $c_{k+1} = (1 - \alpha_k)c_k$.

Proof. The proof is by induction and will use key identity (3.2). For $k = 0$, since $z^0 = x^0$ by (5.5), one has $q_0(x) = f(x^0) + cH(x, z^0)$. Then since $c\nabla_1 H(z^0, z^0) = 0$ (recall the properties of H), and since $z_0 \in C \cap \mathcal{V}$, the optimality conditions imply that $z^0 = \operatorname{argmin}_{x \in \overline{C} \cap \mathcal{V}} q_0(x)$. Now suppose that (5.9) holds for some k and let us prove that for any $x \in \overline{C} \cap \mathcal{V}$,

$$q_{k+1}(x) = q_{k+1}^* + c_{k+1}H(x, z^{k+1}). \tag{5.12}$$

Substituting (5.9) into (5.6) and using $c_{k+1} = (1 - \alpha_k)c_k$, one obtains

$$q_{k+1}(x) = (1 - \alpha_k)q_k^* + c_{k+1}H(x, z^k) + \alpha_k l_k(x, y^k). \tag{5.13}$$

Then by definition of z^{k+1} we have

$$z^{k+1} = \operatorname*{argmin}_{x\in \overline{C}\cap\mathcal{V}} q_{k+1}(x) = u\left(\frac{\alpha_k}{c_{k+1}}\nabla f(y^k), z^k\right)$$

with $z^{k+1} \in C \cap \mathcal{V}$, and

$$q^*_{k+1} = q_{k+1}(z^{k+1}) = (1-\alpha_k)q^*_k + c_{k+1}H(z^{k+1}, z^k) + \alpha_k l_k(z^{k+1}, y^k). \tag{5.14}$$

Subtracting (5.14) from (5.13), one obtains, using (5.7),

$$\begin{aligned} q_{k+1}(x) &= q^*_{k+1} + c_{k+1}[H(x,z^k) - H(z^{k+1},z^k)] + \alpha_k[l_k(x,y^k) - l_k(z^{k+1},y^k)] \\ &= q^*_{k+1} + c_{k+1}[H(x,z^k) - H(z^{k+1},z^k)] + \alpha_k\langle z^{k+1} - x, -\nabla f(y^k)\rangle. \qquad (5.15)\end{aligned}$$

Now, since $z^{k+1} = \operatorname{argmin}_{x\in\overline{C}\cap\mathcal{V}} q_{k+1}(x)$, then writing the optimality conditions for (5.13) (recalling the properties of H) yields

$$c_{k+1}\langle \nabla_1 H(z^{k+1}, z^k), z^{k+1} - x\rangle = -\langle \alpha_k \nabla f(y^k), z^{k+1} - x\rangle \quad \forall x \in \overline{C}\cap\mathcal{V}. \tag{5.16}$$

Using (5.16) in (5.15), it follows that for any $x \in \overline{C} \cap \mathcal{V}$,

$$q_{k+1}(x) = q^*_{k+1} + c_{k+1}[H(x,z^k) - H(z^{k+1},z^k) + \langle z^{k+1} - x, \nabla_1 H(z^{k+1},z^k)\rangle]. \tag{5.17}$$

Invoking the identity (3.2) at $c = x$, $b = z^{k+1}$, and $a = z^k$, the right-hand side of (5.17) reduces to $q_{k+1}(x) = q^*_{k+1} + c_{k+1}H(x, z^{k+1})$, and the lemma is proved. □

The next result is fundamental to determining the main steps of the algorithm, namely the formulas needed to update the sequence $\{x^k\}$ and to determine the choice of the intermediary point y^k.

THEOREM 5.1. *Let $\sigma > 0$, $L > 0$ be given. Suppose that for some $k \geq 0$ we have a point $x^k \in C \cap \mathcal{V}$ such that $f(x^k) \leq q^*_k = \min\{q_k(x) : x \in \overline{C}\cap\mathcal{V}\}$. Let $\alpha_k \in [0,1)$, $c_{k+1} = (1-\alpha_k)c_k$, and $C \cap \mathcal{V} \ni \{z^k\}$ be given by (5.11). Define*

$$y^k = (1-\alpha_k)x^k + \alpha_k z^k, \tag{5.18}$$

$$x^{k+1} = (1-\alpha_k)x^k + \alpha_k z^{k+1}. \tag{5.19}$$

Then, the following inequality holds:

$$q^*_{k+1} \geq f(x^{k+1}) + \frac{1}{2}\left(\frac{c_{k+1}\sigma}{\alpha_k^2} - L\right)\|x^{k+1} - y^k\|^2.$$

Proof. Let $x \in \overline{C}\cap\mathcal{V}$. Since $q_k(x) = q^*_k + c_k H(x, z^k)$, then by (5.6) and using $c_{k+1} = (1-\alpha_k)c_k$ one has

$$q_{k+1}(x) = (1-\alpha_k)q^*_k + c_{k+1}H(x, z^k) + \alpha_k l_k(x, y^k),$$

and with $z^{k+1} = \operatorname{argmin}_{x\in\overline{C}\cap\mathcal{V}} q_{k+1}(x)$ one obtains

$$q_{k+1}(z^{k+1}) = q^*_{k+1} = (1-\alpha_k)q^*_k + c_{k+1}H(z^{k+1}, z^k) + \alpha_k l_k(z^{k+1}, y^k). \tag{5.20}$$

Under our assumption, we have $q^*_k \geq f(x^k)$, and thus using the gradient inequality for f we have

$$q^*_k \geq f(x^k) \geq f(y^k) + \langle x^k - y^k, \nabla f(y^k)\rangle,$$

and it follows from (5.20) and (5.7) that

$$q^*_{k+1} \geq f(y^k) + c_{k+1}H(z^{k+1}, z^k) + \langle \nabla f(y^k), r^k \rangle, \tag{5.21}$$

where $r^k = \alpha_k(z^{k+1} - y^k) + (1 - \alpha_k)(x^k - y^k)$. Noting that r^k can be written as

$$r^k = (1 - \alpha_k)x^k + \alpha_k z^k - y^k + \alpha_k(z^{k+1} - z^k),$$

and since by definition one has $(1 - \alpha_k)x^k + \alpha_k z^k - y^k = 0$, then (5.21) reduces to

$$q^*_{k+1} \geq f(y^k) + c_{k+1}H(z^{k+1}, z^k) + \langle \alpha_k(z^{k+1} - z^k), \nabla f(y^k) \rangle. \tag{5.22}$$

Using the definition of $y^k, x^{k+1} \in C \cap \mathcal{V}$ given in (5.18)–(5.19), one has $x^{k+1} - y^k = \alpha_k(z^{k+1} - z^k)$. Since by hypothesis (h2) h is σ-strongly convex, it follows that $H(z^{k+1}, z^k) \geq \sigma/2\|z^{k+1} - z^k\|^2$, and then from (5.22) we have obtained

$$q^*_{k+1} \geq f(y^k) + \frac{1}{2}\frac{c_{k+1}\sigma}{\alpha_k^2}\|x^{k+1} - y^k\|^2 + \langle \nabla f(y^k), x^{k+1} - y^k \rangle. \tag{5.23}$$

Now, since we assumed that f in $C^{1,1}(C \cap \mathcal{V})$, then by the descent lemma (cf. (4.15)) we have

$$f(y^k) + \langle x^{k+1} - y^k, \nabla f(y^k) \rangle \geq f(x^{k+1}) - \frac{L}{2}\|x^{k+1} - y^k\|^2. \tag{5.24}$$

Combining the latter inequality with (5.23) we obtain

$$q^*_{k+1} \geq f(x^{k+1}) + \frac{1}{2}\left(\frac{c_{k+1}\sigma}{\alpha_k^2} - L\right)\|x^{k+1} - y^k\|^2. \qquad \square$$

Therefore by taking a sequence $\{\alpha_k\}$ with $\sigma c_{k+1} \geq L\alpha_k^2$ we can guarantee that $q^*_{k+1} \geq f(x^{k+1})$. In particular, we can choose $L\alpha_k^2 = \sigma c_k(1 - \alpha_k)$, and this leads to the following improved interior gradient algorithm.

Improved interior gradient algorithm (IGA).

Step 0. Choose a point $x^0 \in C \cap \mathcal{V}$ and a constant $c > 0$. Define $z^0 = x^0 = y^0$, $c_0 = c$, $\lambda = \sigma L^{-1}$.

Step k. For $k \geq 0$, compute the following:

$$\begin{aligned}
\alpha_k &= \frac{\sqrt{(c_k\lambda)^2 + 4c_k\lambda} - \lambda c_k}{2},\\
y^k &= (1 - \alpha_k)x^k + \alpha_k z^k,\\
c_{k+1} &= (1 - \alpha_k)c_k,\\
z^{k+1} &= \operatorname*{argmin}_{x \in \overline{C} \cap \mathcal{V}} \left\{ \left\langle x, \frac{\alpha_k}{c_{k+1}}\nabla f(y^k) \right\rangle + H(x, z^k) \right\} = u\left(\frac{\alpha_k}{c_{k+1}}\nabla f(y^k), z^k\right),\\
x^{k+1} &= (1 - \alpha_k)x^k + \alpha_k z^{k+1}.
\end{aligned}$$

Note that the computational work of this algorithm is exactly the same as that of the interior gradient method in section 4 via the computation of z^{k+1}, since the remaining steps involve trivial computations. To estimate the rate of convergence we need the following simple lemma on the sequence α_k; see [34] for a proof.

LEMMA 5.3. *Let $\lambda_k > 0$, $c_k > 0$ with $c_0 = c$, and let $\{\alpha_k\}$ be the sequence with $\alpha_k \in [0,1[$ defined by $\alpha_k^2 = \lambda_k c_k(1-\alpha_k)$ with $c_{k+1} = (1-\alpha_k)c_k$. Set $\gamma_k := \prod_{l=0}^{k-1}(1-\alpha_l)$. Then*

$$\gamma_k \le \left(1 + \frac{\sqrt{c}}{2}\sum_{l=0}^{k-1}\sqrt{\lambda_l}\right)^{-2}.$$

In particular, with $\lambda_l = \lambda\ \forall l$ we have $\gamma_k \le 4(k\sqrt{\lambda c}+2)^{-2}$.

We thus obtain a convergent interior gradient method with an improved convergence rate estimate.

THEOREM 5.2. *Let $\{x^k\}, \{y^k\}$ be the sequences generated by IGA and let x^* be an optimal solution of* (P). *Then for any $k \ge 0$ we have*

$$f(x^k) - f(x^*) \le \frac{4L}{\sigma k^2 c}C(x^*, x^0) = O\left(\frac{1}{k^2}\right),$$

where $C(x^*, x^0) = c_0 H(x^*, x^0) + f(x^0) - f(x^*)$ and the sequence $\{x^k\}$ is minimizing, i.e., $f(x^k) \to f(x^*)$.

Proof. By Lemma 5.1, the sequence of functions $\{q_k(\cdot)\}$ satisfies (5.1) and thus (5.4) holds; i.e., using (5.5) we have

$$f(x^k) - f(x^*) \le \gamma_k(q_0(x^*) - f(x^*)) = \gamma_k(f(x^0) + c_0H(x^*, x^0) - f(x^*)) = \gamma_k C(x^*, x^0).$$

Specializing Lemma 5.3 with $\lambda_k = \sigma L^{-1}$, we obtain

$$\gamma_k \le \frac{4L}{\left(k\sqrt{\sigma c} + 2\sqrt{L}\right)^2} \le \frac{4L}{\sigma c k^2},$$

from which the desired result follows. □

Thus, to solve (P) to accuracy $\varepsilon > 0$, one needs no more than $\lfloor O(1/\sqrt{\varepsilon})\rfloor$ iterations of IGA, which is a significant reduction (by a squared root factor) in comparison to the interior gradient method of section 4. In particular, we note that IGA can be used to solve convex minimization over the unit simplex with this improved global convergence rate estimate for the EMDA of section 4.

REFERENCES

[1] F. ALVAREZ, J. BOLTE, AND O. BRAHIC, *Hessian Riemannian gradient flows in convex programming*, SIAM J. Control Optim., 43 (2004), pp. 477–501.

[2] H. ATTOUCH AND M. TEBOULLE, *A regularized Lotka Volterra dynamical system as a continuous proximal-like method in optimization*, J. Optim. Theory Appl., 121 (2004), pp. 541–570.

[3] A. AUSLENDER AND M. HADDOU, *An interior proximal method for convex linearly constrained problems and its extension to variational inequalities*, Math. Program., 71 (1995), pp. 77–100.

[4] A. AUSLENDER, M. TEBOULLE, AND S. BEN-TIBA, *A logarithmic-quadratic proximal method for variational inequalities*, Comput. Optim. Appl., 12 (1999), pp. 31–40.

[5] A. AUSLENDER, M. TEBOULLE, AND S. BEN-TIBA, *Interior proximal and multiplier methods based on second order homogeneous kernels*, Math. Oper. Res., 24 (1999), pp. 645–668.

[6] A. AUSLENDER AND M. TEBOULLE, *Asymptotic Cones and Functions in Optimization and Variational Inequalities*, Springer Monogr. Math., Springer-Verlag, New York, 2003.

[7] A. AUSLENDER AND M. TEBOULLE, *Interior gradient and epsilon-subgradient methods for constrained convex minimization*, Math. Oper. Res., 29 (2004), pp. 1–26.

[8] A. BECK AND M. TEBOULLE, *Mirror descent and nonlinear projected subgradient methods for convex optimization*, Oper. Res. Lett., 31 (2003), pp. 167–175.

[9] A. Ben-Tal, T. Margalit, and A. Nemirovski, *The ordered subsets mirror descent optimization method with applications to tomography*, SIAM J. Optim., 12 (2001), pp. 79–108.

[10] A. Ben-Tal and M. Zibulevsky, *Penalty/barrier methods for convex programming problems*, SIAM J. Optim., 7 (1997), pp. 347–366.

[11] D. P. Bertsekas, *On the Goldstein-Levitin-Polyak gradient projection method*, IEEE Trans. Automat. Control, 21 (1976), pp. 174–183.

[12] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, MA, 1999.

[13] J. Bolte and M. Teboulle, *Barrier operators and associated gradient-like dynamical systems for constrained minimization problems*, SIAM J. Control Optim., 42 (2003), pp. 1266–1292.

[14] H. Brezis, *Analyse Fonctionnelle: Theorie et applications*, Masson, Paris, 1987.

[15] Y. Censor and S. Zenios, *The proximal minimization algorithm with D-functions*, J. Optim. Theory Appl., 73 (1992), pp. 451–464.

[16] G. Chen and M. Teboulle, *Convergence analysis of a proximal-like minimization algorithm using Bregman functions*, SIAM J. Optim., 3 (1993), pp. 538–543.

[17] R. Correa and C. Lemarechal, *Convergence of some algorithm for convex programming*, Math. Program., 62 (1993), pp. 261–275.

[18] M. Doljansky and M. Teboulle, *An interior proximal algorithm and the exponential multiplier method for semidefinite programming*, SIAM J. Optim., 9 (1998), pp. 1–13.

[19] J. Eckstein, *Nonlinear proximal point algorithms using Bregman functions, with applications to convex programming*, Math. Oper. Res., 18 (1993), pp. 202–226.

[20] J. Eckstein, *Approximate iterations in Bregman-function-based proximal algorithms*, Math. Program., 83 (1998), pp. 113–123.

[21] P. P. B. Eggermont, *Multiplicatively iterative algorithms for convex programming*, Linear Algebra Appl., 130 (1990), pp. 25–42.

[22] J. Faraut and A. Korányi, *Analysis on Symmetric Cones*, Oxford Math. Monogr., The Claredon Press, Oxford University Press, New York, 1994.

[23] M. Fukushima, Z.-Q. Luo, and P. Tseng, *Smoothing functions for second-order-cone complementarity problems*, SIAM J. Optim., 12 (2001), pp. 436–460.

[24] O. Güler, *On the convergence of the proximal point algorithm for convex minimization*, SIAM J. Control Optim., 29 (1991), pp. 403–419.

[25] A. N. Iusem, *Interior point multiplicative methods for optimization under positivity constraints*, Acta Appl. Math., 38 (1995), pp. 163–184.

[26] A. N. Iusem, B. F. Svaiter, and M. Teboulle, *Multiplicative interior gradient methods for minimization over the nonnegative orthant*, SIAM J. Control Optim., 34 (1996), pp. 389–406.

[27] A. N. Iusem, B. Svaiter, and M. Teboulle, *Entropy-like proximal methods in convex programming*, Math. Oper. Res., 19 (1994), pp. 790–814.

[28] A. N. Iusem and M. Teboulle, *Convergence rate analysis of nonquadratic proximal and augmented Lagrangian methods for convex and linear programming*, Math. Oper. Res., 20 (1995), pp. 657–677.

[29] K. C. Kiwiel, *Proximal minimization methods with generalized Bregman functions*, SIAM J. Control Optim., 35 (1997), pp. 1142–1168.

[30] B. Lemaire, *The proximal algorithm*, in New Methods in Optimization and Their Industrial Uses, Internat. Schriftenreihe Numer. Math. 87, J. P. Penot, ed., Birkhäuser, Basel, 1989, pp. 73–87.

[31] B. Martinet, *Regularisation d'inéquations variationnelles par approximations successives*, Rev. Française Informat. Recherche Opérationnelle, 4 (1970), pp. 154–158.

[32] A. Nemirovski and D. Yudin, *Problem Complexity and Method Efficiency in Optimization*, John Wiley, New York, 1983.

[33] Y. Nesterov and A. Nemirovskii, *Interior Point Polynomial Algorithms in Convex Programming*, SIAM, Philadelphia, 1994.

[34] Y. Nesterov, *On an approach to the construction of optimal methods of minimization of smooth convex functions*, Èkonom. i Mat. Metody, 24 (1988), pp. 509–517.

[35] B. T. Polyak, *Introduction to Optimization*, Optimization Software, New York, 1987.

[36] R. A. Polyak, *Nonlinear rescaling vs. smoothing technique in constrained optimization*, Math. Program., 92 (2002), pp. 197–235.

[37] R. A. Polyak and M. Teboulle, *Nonlinear rescaling and proximal-like methods in convex optimization*, Math. Program., 76 (1997), pp. 265–284.

[38] S. M. Robinson, *Linear convergence of epsilon subgradients methods for a class of convex functions*, Math. Program., 86 (1999), pp. 41–50.

[39] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[40] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.

[41] N. Z. Shor, *Minimization Methods for Nondifferentiable Functions*, Springer-Verlag, Berlin, 1985.

[42] P. J. da Silva e Silva, J. Eckstein, and C. Humes, Jr., *Rescaling and stepsize selection in proximal methods using separable generalized distances*, SIAM J. Optim., 12 (2001), pp. 238–261.

[43] M. Teboulle, *Entropic proximal mappings with applications to nonlinear programming*, Math. Oper. Res., 17 (1992), pp. 670–681.

[44] M. Teboulle, *Convergence of proximal-like algorithms*, SIAM J. Optim., 7 (1997), pp. 1069–1083.

[45] P. Tseng and D. P. Bertsekas, *On the convergence of the exponential multiplier method for convex programming*, Math. Program., 60 (1993), pp. 1–19.

SIAM J. OPTIM.
Vol. 16, No. 3, pp. 726–750

SOLVING LIFT-AND-PROJECT RELAXATIONS OF BINARY INTEGER PROGRAMS*

SAMUEL BURER† AND DIETER VANDENBUSSCHE‡

Abstract. We propose a method for optimizing the lift-and-project relaxations of binary integer programs introduced by Lovász and Schrijver. In particular, we study both linear and semidefinite relaxations. The key idea is a restructuring of the relaxations, which isolates the complicating constraints and allows for a Lagrangian approach. We detail an enhanced subgradient method and discuss its efficient implementation. Computational results illustrate that our algorithm produces tight bounds more quickly than state-of-the-art linear and semidefinite solvers.

Key words. integer programming, lift-and-project relaxations, semidefinite programming, augmented Lagrangian

AMS subject classifications. 90C10, 90C22, 90C27, 90C30

DOI. 10.1137/040609574

1. Introduction.

In the field of optimization, binary integer programs have proven to be an excellent source of challenging problems, and the successful solution of larger and larger problems over the past few decades has required significant theoretical and computational advances. One of the fundamental issues is how to obtain a "good" description of the convex hull of integer solutions, and many specific classes of integer programs have been solved by finding problem-specific ways to address this issue.

Researchers have also developed techniques for approximating the convex hull of integer solutions without any specific knowledge of the problem, i.e., techniques that apply to arbitrary binary integer programs. Some of the earliest work done in this direction was by Gomory [19] in generating linear inequalities that tighten the basic linear relaxation. A different idea, which has been advocated by several authors, is to approximate the convex hull as the projection of some polyhedron lying in a space of higher dimension. We refer the reader to [3, 38, 32, 4, 24, 7]. Connections between these works are explored in [27, 26].

Although these so-called *lift-and-project* methods are quite powerful theoretically, they present great computational challenges because one typically must optimize in the space of the lifting, i.e., the space of higher dimension. Computational issues are detailed in [4, 39, 11, 22, 14].

In this paper, we focus on the techniques proposed by Lovász and Schrijver (LS), including both linear and semidefinite relaxations. In particular, our main goal is to present improved computational methods for optimizing over the first-level LS relaxations. We are aware of only one study (by Dash [14]), which investigates the strength of these relaxations computationally. This shortage of computational experience is due to the dramatic size of these relaxations. For example, one specific semidefinite

*Received by the editors June 7, 2004; accepted for publication (in revised form) June 17, 2005; published electronically January 6, 2006.

http://www.siam.org/journals/siopt/16-3/60957.html

†Department of Management Sciences, University of Iowa, Iowa City, IA 52242-1000 (samuelburer@uiowa.edu). This author was supported in part by NSF grant CCR-0203426.

‡Department of Mechanical and Industrial Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 (dieterv@uiuc.edu).

relaxation that has been considered by Dash (and which we also consider in this paper) has well over 1.7 million constraints.

Since it is unlikely that these relaxations can be solved using direct algorithms, we instead adopt the paradigm of decomposition, which is common in large-scale optimization methods. (Dash [14] also considers a decomposition approach.) The main idea here is a clever decomposition of the LS relaxations, which allows for a Lagrangian approach. Instead of using the common subgradient algorithm, however, we propose to use an augmented Lagrangian algorithm, which is, in some sense, an enhanced subgradient method and which also has connections with the bundle method for convex optimization. We provide a theoretical explanation of the benefits of the augmented Lagrangian algorithm and give a detailed explanation of our implementation, which is demonstrated to outperform state-of-the-art subgradient, linear, and semidefinite solvers on certain classes of problems.

We remark that, while the idea of using the augmented Lagrangian method for linear programs is not new (see [35, 40]), little work has been done on the computational aspects of such a method. In this paper, we fill this gap concerning large-scale linear programs and also present an augmented Lagrangian method for semidefinite programs for the first time.

The paper is organized as follows. In section 2, we give background on the Lovász-Schrijver lift-and-project relaxations as well as propose the decomposition technique that will become the basis of our augmented Lagrangian algorithm. Then, in section 3, we discuss the augmented Lagrangian algorithm, including its theoretical benefits and specialization in the current context. Next, in section 4, we present the details of our implementation and computational results. We also discuss the strength of the LS relaxations on various problem classes, with one highlight being that, in practice, the LS semidefinite relaxation provides the strongest known bounds for a collection of problems in the Quadratic Assignment Problem Library [36]. Finally, we conclude with a few final remarks and suggestions for future research in section 5.

1.1. Notation and terminology.

In this section, we introduce some of the notation that will be used throughout the paper. $\mathbb{R}^n$ will refer to n-dimensional Euclidean space. The norm of a vector $x \in \mathbb{R}^n$ is denoted by $\|x\| := \sqrt{x^T x}$. We let $e_i \in \mathbb{R}^n$ represent the ith unit vector, and e is the vector of all ones. $\mathbb{R}^{n\times n}$ is the set of real $n \times n$ matrices, $\mathcal{S}^n$ is the set of symmetric matrices in $\mathbb{R}^{n\times n}$, while $\mathcal{S}^n_+$ is the set of positive semidefinite symmetric matrices. The special notation $\mathbb{R}^{1+n}$ and $\mathcal{S}^{1+n}$ is used to denote the spaces $\mathbb{R}^n$ and $\mathcal{S}^n$ with an additional "zeroth" entry prefixed or an additional zeroth row and zeroth column prefixed, respectively. The inner product of two matrices $A, B \in \mathbb{R}^{n\times n}$ is defined as $A \bullet B := \operatorname{tr}(A^T B)$, where $\operatorname{tr}(\cdot)$ denotes the sum of the diagonal entries of a matrix. The Frobenius norm of a matrix $A \in \mathbb{R}^{n\times n}$ is defined as $\|A\|_F := \sqrt{A \bullet A}$. $\operatorname{diag}(A)$ is defined as the vector with the diagonal of A as its entries.

2. The lift-and-project operators of Lovász and Schrijver.

When solving a 0-1 integer program of the form

$$\text{(IP)} \qquad \min\left\{c^T x \mid Ax \le b,\ x \in \{0,1\}^n\right\},$$

we are often interested in relaxations of the convex hull of integer solutions

$$P := \operatorname{conv}\{x \in \{0,1\}^n \mid Ax \le b\}.$$

Optimization over such relaxations provides lower bounds that can be used within branch-and-bound methods or allow one to assess the quality of feasible solutions of (IP). The trivial linear programming (LP) relaxation is obtained by replacing $x \in \{0,1\}^n$ with $x \in [0,1]^n$. In an effort to develop relaxations that are stronger than the LP relaxation, Lovász and Schrijver [32] introduced the lifted matrix variable

$$Y = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T = \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix}.$$

Given this definition relating Y and $x \in \{0,1\}^n$, we can observe a number of interesting properties of Y:

1. Y is symmetric and positive semidefinite, i.e., $Y \in \mathcal{S}_+^{1+n}$;
2. the diagonal of Y equals the zeroth column of Y, i.e., $\operatorname{diag}(Y) = Ye_0$;
3. if we multiply the constraints $Ax \le b$ of P by some x_i, we obtain the set of nonlinear inequalities $bx_i - Axx_i \ge 0$, which are valid for P; these inequalities can be written in terms of Y as

$$\left(b \middle| -A\right) Ye_i \ge 0 \quad \forall\, i = 1, \ldots, n;$$

4. analogously, multiplying $Ax \le b$ by $1 - x_i$ yields

$$\left(b \middle| -A\right) Y(e_0 - e_i) \ge 0 \quad \forall\, i = 1, \ldots, n.$$

Lovász and Schrijver [32] observed that these properties could be used to obtain relaxations of P. In particular, they homogenized the standard LP relaxation of (IP) by defining

$$K := \left\{ \begin{pmatrix} x_0 \\ x \end{pmatrix} \in \mathbb{R}^{1+n} \mid Ax \le x_0 b,\ 0 \le x \le x_0 e \right\}.$$

We remark that enforcing $x_0 = 1$ in K yields the LP relaxation and that the third and fourth properties above can be written as $Ye_i \in K$ and $Y(e_0 - e_i) \in K$. They then proposed the following sets:

$$M(K) := \left\{ Y \in \mathcal{S}^{1+n} \mid Ye_0 = \operatorname{diag}(Y), \quad Ye_i \in K,\ Y(e_0 - e_i) \in K \ \ \forall\, i = 1, \ldots, n \right\}$$
$$M_+(K) := \left\{ Y \in \mathcal{S}_+^{1+n} \mid Y \in M(K) \right\}.$$

Note that $M_+(K)$ differs from $M(K)$ only in that positive semidefiniteness is enforced on the Y variable. $M(K)$ now leads to a linear relaxation of P via the projected set

$$N(K) := \left\{ x \in \mathbb{R}^n \mid \begin{pmatrix} 1 \\ x \end{pmatrix} = \operatorname{diag}(Y) \ \text{ for some } \ Y \in M(K) \right\},$$

and $M_+(K)$ leads to an analogous semidefinite relaxation $N_+(K)$ of P. In particular, Lovász and Schrijver [32] showed that $P \subseteq N_+(K) \subseteq N(K)$ and that $N(K)$ is contained in the LP relaxation of (IP). Further, they showed that applying these relaxation procedures iteratively n times yields P exactly.

We remark that our definitions of $N(K)$ and $N_+(K)$ are actually slices (at $Y_{00} = 1$ and $x_0 = 1$) of the cones originally defined by Lovász and Schrijver [32].

Applying these ideas to the stable set problem, Lovász and Schrijver proved that some classes of inequalities for the stable set polytope are satisfied by all the points in $N(K)$, while other classes are only valid for $N_+(K)$. In turn, these results have significant implications for the complexity of finding maximum stable sets in various classes of graphs. Further, theoretical results concerning the strength of $N(K)$ and

$N_+(K)$, as well as the higher-order liftings, have also been established; see [41, 12, 18, 25, 30].

To compute lower bounds available from the relaxations $N(K)$ and $N_+(K)$, one must solve the LP

$$\min\left\{c^T x \mid x \in N(K)\right\} \tag{1}$$

or the semidefinite program (SDP)

$$\min\left\{c^T x \mid x \in N_+(K)\right\}. \tag{2}$$

Defining $\tilde{c} := \begin{pmatrix} 0 \\ c \end{pmatrix}$ and using the above definitions, (1) can be written explicitly as

$$\begin{aligned}
&\min && \tilde{c}^T Y e_0 && (3)\\
&\text{s.t.} && Y = Y^T, && (4)\\
& && Ye_0 = \operatorname{diag}(Y), && (5)\\
& && Ye_i \in K \qquad\quad \forall\, i = 1, \ldots, n, && (6)\\
& && Y(e_0 - e_i) \in K \quad \forall\, i = 1, \ldots, n, && (7)\\
& && Y_{00} = 1. && (8)
\end{aligned}$$

Likewise, (2) can be written as

$$\begin{aligned}
\min \quad & \tilde{c}^T Y e_0 \\
\text{s.t.} \quad & Y \in \mathcal{S}_+^{1+n} \\
& (4)\text{–}(8).
\end{aligned}$$

A couple of comments concerning (1) and (2) are in order. First, the constraints (6) and (7) imply $Ye_0 \in K$. Combined with (8), this in turn implies that each component of the zeroth column of Y is in $[0, 1]$. By (4), the same holds for the zeroth row of Y, which implies by (6) that all other components of Y are in $[0, 1]$. Hence, we may replace K with $\hat{K} := K \cap [0, 1]^{1+n}$ without affecting the optimal solution sets of (1) and (2).

Second, if K is defined by m constraints, including upper and lower bounds, then the LP described by (3)–(8) has $\mathcal{O}(n^2 + nm)$ constraints and $\mathcal{O}(n^2)$ variables. Consequently, solving this LP using, say, the simplex method or interior-point methods becomes too cumbersome even for problems with moderate n and m. This situation is further exacerbated when solving (2). In fact, using standard SDP methods, (2) will only be solvable for very small n and m. As a result, very little research has been done on actually solving (1) and (2). Work involving (1) is discussed in [39] and some computations of (2) for various 0-1 polytopes can be found in [14].

This second observation motivates us to investigate new optimization techniques for solving (1) and (2), and, in particular, we are interested in applying decomposition methods for large-scale optimization. We first show how this can be done for (1). Unfortunately, all the constraints (4)–(8) are tightly linked, and so the problem does not immediately lend itself to decomposition. To partially overcome this obstacle, however, we introduce the matrix variable

$$Z = YQ \in \mathbb{R}^{(1+n)\times n}, \qquad \text{where} \quad Q := \Big(e_0 - e_1 \Big| e_0 - e_2 \Big| \cdots \Big| e_0 - e_n\Big)$$

and reformulate (3)–(8) as

$$\min \quad \tilde{c}^T Y e_0$$

$$\text{s.t.} \quad Y = Y^T, \quad Ye_0 = \operatorname{diag}(Y), \quad Z = YQ \tag{9}$$

$$Ye_i \in \hat{K}, \quad Ze_i \in \hat{K} \quad \forall\, i = 1, \ldots, n \tag{10}$$

$$Y_{00} = 1. \tag{11}$$

Note that K has been replaced with $\hat{K}$ in accordance with the first comment above. Furthermore, it is clear that the constraints (10) are separable over the columns of Y and Z but that these same columns are linked via the constraints (9).

A reasonable idea is to apply Lagrangian relaxation to the constraints (9), and in order to simplify notation, we denote (9) by the collection of linear equations $h(Y, Z) = 0$. Letting λ denote the vector of unconstrained dual multipliers for $h(Y, Z) = 0$, we obtain the Lagrangian relaxation

$$L(\lambda) := \min \quad \tilde{c}^T Y e_0 + \lambda^T h(Z, Y)$$

$$\text{s.t.} \quad Ye_i \in \hat{K} \quad \forall\, i = 0, 1, \ldots, n \tag{12}$$

$$Ze_i \in \hat{K} \quad \forall\, i = 1, \ldots, n \tag{13}$$

$$Y_{00} = 1. \tag{14}$$

Note that we have added the constraint $Ye_0 \in K$, which is redundant for (9) and (10) but is included here in order to properly constrain the zeroth column of Y. It is now clear that $L(\lambda)$ is separable over the columns of Y and Z, and so to evaluate $L(\lambda)$ for any λ, we can simply solve $2n + 1$ separate linear optimizations over $\hat{K}$ (while respecting the simple constraint $Y_{00} = 1$). Furthermore, from standard LP theory, we know that the optimal value of

$$\max_{\lambda} \quad L(\lambda) \tag{15}$$

equals the optimal value of (1).

The semidefinite optimization (2) can be approached in a similar fashion, i.e., by introducing the auxiliary variable Z and then relaxing the linking constraints. However, we must also introduce a dual multiplier $S \in \mathcal{S}_+^{1+n}$ for the constraint that keeps Y positive semidefinite, which modifies the Lagrangian relaxation to read

$$L(\lambda, S) := \min \quad \tilde{c}^T Y e_0 + \lambda^T h(Z, Y) - S \bullet Y$$
$$\text{s.t.} \quad (12)\text{–}(14),$$

so that the resulting Lagrangian optimization is

$$\sup_{\lambda, S} \left\{ L(\lambda, S) \mid S \in \mathcal{S}_+^{1+n} \right\}. \tag{16}$$

It is well known that the dual SDP of (2) has an interior-point, i.e., it satisfies Slater's condition, which implies that there is no duality gap and that optimality is attained in (2)—although optimality may not be attained in the dual of (2). As a result, it is not difficult to see that the optimal value of (16) equals that of (2).

Theoretically, one can solve both (15) and (16) using a subgradient method. Our initial experiments, however, indicated that convergence of subgradient methods was sometimes slow. This motivated us to examine an augmented Lagrangian method, which overall proved to be more robust while still allowing us to exploit the structure inherent in (1) and (2). We discuss these issues in detail in sections 3 and 4.

3. The augmented Lagrangian method for linear conic programs.

In this section, we discuss the specialization of the augmented Lagrangian method—a standard tool of nonlinear programming (NLP)—to the case of linear optimization over linear and conic constraints, of which problems (1) and (2) are particular examples. More specifically, let $C \subseteq \mathbb{R}^q$ be a closed, convex cone, and let

$$X := \{y \in \mathbb{R}^q \mid Ey = f,\ y \in C\}. \tag{17}$$

We consider the following generic problem throughout this section:

$$\min \left\{c^T y \mid Ay = b,\ y \in X\right\}. \tag{18}$$

Here, $y \in \mathbb{R}^q$ is the optimization variable and $c \in \mathbb{R}^q$, $A \in \mathbb{R}^{p \times q}$, and $b \in \mathbb{R}^p$ are the data. We assume that an optimal solution exists and denote the optimal value by v^*.

We acknowledge that initially it may seem counterintuitive to apply a NLP algorithm to linear conic problems. In fact, we are aware of only two studies [35, 40], which consider the augmented Lagrangian method in such a context, in particular for $C = \mathbb{R}^q_+$. In this section, however, besides laying the groundwork for the augmented Lagrangian algorithm, we also hope to highlight the advantages of the method, which when combined with several computational ideas discussed in section 4 make it a good choice for optimizing (1) and (2).

3.1. The augmented Lagrangian method.

The augmented Lagrangian method can be seen as a combination of the standard subgradient and quadratic penalty methods. It is based on the following function, which is specified for fixed $\lambda \in \mathbb{R}^p$ and $\sigma > 0$:

$$L_{\lambda,\sigma}(y) = c^T y + \lambda^T (b - Ay) + \frac{\sigma}{2} \|b - Ay\|^2. \tag{19}$$

The augmented Lagrangian method is then stated as Algorithm 1. Roughly speaking, the augmented Lagrangian method runs the subgradient and quadratic penalty methods at the same time by alternating between the update of λ and σ (typically using some predefined update strategy, such as performing the nontrivial update of σ every 10 iterations). Some of the main advantages of the augmented Lagrangian algorithm over subgradient and penalty methods are that it yields both primal and dual solutions, as well as dual bounds, for (18).

Algorithm 1 Augmented Lagrangian algorithm

```
Set $\lambda^1 = 0$, $\sigma_1 = 1$
for $k = 1, 2, 3, \dots,$ do
    Calculate some $y^k \in \operatorname{Argmin}\left\{L_{\lambda^k,\sigma_k}(y) : y \in X\right\}$;
    Calculate the subgradient $d^k = b - Ay^k$;
    Choose either ($\alpha_k = \sigma_k$ and $\eta_k = 1$) or ($\alpha_k = 0$ and $\eta_k \gg 1$);
    Calculate $\lambda^{k+1} = \lambda^k + \alpha_k d^k$ and $\sigma_{k+1} = \eta_k \sigma_k$.
end for
```

There are also some additional, less obvious advantages of the augmented Lagrangian method. First, an important feature of the augmented Lagrangian, in contrast with the subgradient method, is that there is a definitive choice of step-size α_k in each iteration. This choice is dictated by convergence results for augmented Lagrangian methods (see [6]) and also seems to work well in practice. Second, it is well known in the study of NLP that the introduction of explicit dual variables into the quadratic penalty method tends to lessen the ill-conditioning encountered in penalty methods. In particular, it can be proved that if the iterates λ^k are sufficiently close to an optimal dual solution, then there is a finite value of σ that still guarantees convergence [6].

In fact, in the specific case of (18), where X is given by (17), it is possible to show a much stronger result, namely that Algorithm 1 converges even if σ_k is held constant at an arbitrary initial value $\sigma_1 > 0$. In order to state the result, we point out that the dual of (18) can be written explicitly as

$$\max \left\{ b^T\lambda + f^T\eta \mid A^T\lambda + E^T\eta + s = c,\ s \in C^* \right\}, \tag{20}$$

where

$$C^* := \left\{ s \in \mathbb{R}^q : s^T y \geq 0 \ \ \forall\, y \in C \right\}$$

is the dual cone of C, $\eta \in \mathbb{R}^m$ is the dual variable associated with the constraint $Ey = f$, and $s \in \mathbb{R}^q$ is the dual variable associated with the constraint $x \in C$. We also make the following assumptions: A has full row rank, and both (18) and (20) have Slater points, i.e., feasible points in the interior of C and C^*, respectively. In addition, we let η^k and s^k denote the optimal multipliers associated with $Ey = f$ and $y \in C$ gotten from the solution of the kth augmented Lagrangian subproblem. The result is the following theorem.

THEOREM 3.1. *Let X be given by* (17)*, and suppose that Algorithm* 1 *is executed so that $\sigma_k = \sigma_1$ for all $k \geq 1$, i.e., the dual multiplier is updated nontrivially in each iteration. Then any accumulation point of the combined sequence $\{(y^k, \lambda^{k+1}, \eta^k, s^k)\}$ constitutes a primal-dual optimal pair of* (18) *and* (20).

This result is proven by Poljak and Tret′jakov [35] for the specific case of $C = \mathbb{R}^q_+$, and the essence of their proof carries over to general C. Although it is not difficult to extend the result based on their ideas, we include a proof in section 3.4 for completeness.

We finish this subsection with a few observations. First, Theorem 3.1 shows that (1) and (2) can theoretically be solved without ever increasing σ, and computational experiments support this. In practice, however, it still may be advantageous to increase σ to moderate levels in order to facilitate convergence of the method, as we demonstrate in section 4.

Second, even with the theoretical and practical benefits of augmented Lagrangian methods, it is still important to keep in mind that the inner optimization over $y \in X$ in each iteration of the algorithm utilizes a convex quadratic objective, instead of a linear one as in the subgradient method. In some applications, this is a disadvantage of the augmented Lagrangian method that may preclude its use, but we will show that, in the case of (1) and (2), the use of the augmented Lagrangian method is indeed beneficial.

Third, in practice it is rarely the case that the inner minimization of Algorithm 1 is computed exactly. Nonetheless, convergence is typically observed in practice even if y^k is a "nearly" optimal solution [6]. This behavior will guide some of our implementation decisions, which we describe in section 4.

3.2. Variations on the augmented Lagrangian method.

Typically the augmented Lagrangian method is stated in terms of handling difficult equality constraints, such as the constraints $Ay = b$ in (18). Although there exist variations which can handle inequality constraints directly (see [34]), a standard approach for handling inequalities is to simply add slack variables and then to revert to the equality case. For example, consider the problem

$$\min \left\{ c^T y \mid Ay \le b,\ y \in X \right\}.$$

By introducing the variable $z \in \mathbb{R}^p_+$, we have the equivalent problem

$$\min \left\{ c^T y \mid Ay + z = b,\ (y, z) \in X \times \mathbb{R}^p_+ \right\}.$$

The augmented Lagrangian can now be applied directly to the second formulation. Of course, the inner optimization of the augmented Lagrangian algorithm is now slightly more complicated due to the addition of slack variables, but this complication is usually worth the trouble.

Further extensions of this idea can also be considered. If $Z \subseteq \mathbb{R}^p$ is an arbitrary set, a problem of the form

$$\min \left\{ c^T y \mid b - Ay \in Z,\ y \in X \right\}$$

can then be converted to

$$\min \left\{ c^T y : Ay + z = b,\ (y, z) \in X \times Z \right\},$$

after which the augmented Lagrangian method can be applied. Here again, the simplicity of the inner optimization over $(y, z) \in X \times Z$ is the key to the overall efficiency of the augmented Lagrangian algorithm. In particular, convergence will be theoretically and practically more reliable if Z is convex.

3.3. Relationship with the bundle method. In section 3.1, we have presented the augmented Lagrangian algorithm as an alternative to the standard subgradient algorithm. When the set X is convex, another well-known alternative to the subgradient algorithm is the bundle method (see [29, 23]). Like the subgradient method, the bundle method uses subgradient information to produce a sequence $\{\lambda^k\}$ of dual multipliers whose corresponding Lagrangian objective values converge to v^*. However, the bundle method differs from the subgradient method in the precise way that the subgradient information is used.

The bundle method is initialized with $\lambda^1 = 0$ and, at the kth iteration, the basic idea is to calculate λ^{k+1} by solving an approximation of the Lagrangian dual optimization

$$\sup_{\lambda \in \mathbb{R}^p} \min \left\{ c^T y + \lambda^T (b - Ay) \mid y \in X \right\}. \tag{21}$$

More specifically, the bundle method assumes that a current best point $\bar{\lambda}$ has been calculated (note that $\bar{\lambda}$ does not necessarily equal λ^k) and that a finite collection of dual points $\{\lambda^j \mid j \in J^k\}$ is available, for some finite index set J^k. For example, one may take $J^k = \{1, \ldots, k\}$ so that the dual points correspond to the dual solutions λ^j already produced by the algorithm in the first $k-1$ iterations, though other choices are possible. Defining $\tilde{X} := \operatorname{conv}\{y^j : j \in J^k\}$, the approximation of (21) is then given as

$$\sup_{\lambda \in \mathbb{R}^p} \min \left\{ c^T y + \lambda^T (b - Ay) \mid y \in \tilde{X} \right\}. \tag{22}$$

Generally speaking, the inner minimization of (22) (viewed as a function of λ) is only considered a reliable approximation of the inner minimization of (21) for those λ relatively close to $\bar{\lambda}$. Hence, the next iterate λ^{k+1} is not actually chosen as an optimal solution of (22), but rather as an optimal solution of

$$\max_{\lambda \in \mathbb{R}^p} \ \min \left\{ c^T y + \lambda^T (b - Ay) \mid y \in \tilde{X} \right\} - \frac{\rho}{2} \left\| \lambda - \bar{\lambda} \right\|^2, \tag{23}$$

where $\rho > 0$ is a proximity parameter. In other words, (23) is similar to (22) except that it penalizes points that are too far from the current best iterate $\bar{\lambda}$, and the parameter ρ controls the precise amount of penalization. Once λ^{k+1} has been calculated, the bundle method calculates the value of the Lagrangian function at λ^{k+1} and decides whether or not λ^{k+1} should become the new best iterate $\bar{\lambda}$.

With this description of the bundle method, it is not difficult to see that the bundle method's procedure for computing λ^{k+1} is similar to that of the augmented Lagrangian algorithm. Indeed, (23) can be rearranged as

$$\min \left\{ \max_{\lambda \in \mathbb{R}^p} \ c^T y + \lambda^T (b - Ay) - \frac{\rho}{2} \left\| \lambda - \bar{\lambda} \right\|^2 \ \mid y \in \tilde{X} \right\}. \tag{24}$$

Since the inner optimization of (24) is a concave maximization over $\lambda \in \mathbb{R}^p$, its optimal solution is given by $\bar{\lambda} + \rho^{-1}(b - Ay)$, which further simplifies (24) to

$$\min \left\{ c^T y + \bar{\lambda}^T (b - Ay) + \frac{1}{2\rho} \| b - Ay \|^2 \mid y \in \tilde{X} \right\}. \tag{25}$$

Letting $\sigma = \rho^{-1}$, we now easily see that (25) is similar in form to the kth augmented Lagrangian subproblem except that (25) approximates X by $\tilde{X}$. Next, once the bundle method calculates an optimal solution y^k of (25), λ^{k+1} is calculated by the formula

$$\lambda^{k+1} := \lambda^k + \rho^{-1} \left(b - Ay^k \right),$$

which matches the update formula used by the augmented Lagrangian algorithm.

As described above, in addition to the approximation $\tilde{X}$ of X, the bundle method differs from the augmented Lagrangian method in that it selectively keeps a current best iterate $\bar{\lambda}$ (which affects each stage of the algorithm), whereas the augmented Lagrangian algorithm simply generates each λ^k in succession. It is further interesting to note that the bundle method is known to converge for fixed ρ.

3.4. Proof of Theorem 3.1. In this subsection, we give the proof of Theorem 3.1, which has been stated in section 3.1. We remark that our proof is an extension of the proof given in [35] for the case $C = \mathbb{R}^q_+$.

The main idea of the theorem is that, when X is given by (17), the augmented Lagrangian algorithm converges without ever increasing the penalty parameter σ. For this, we consider the value $\sigma > 0$ to be fixed throughout the execution of Algorithm 1, i.e., $\sigma_k = \sigma$ for all $k \geq 1$. Also recall the following assumptions: A has full row rank, and (18) and (20) have Slater points. We will investigate three sequences produced by the algorithm:

(i) the sequence $\{y^k\}$ of primal estimates;

(ii) the shifted sequence $\{\lambda^{k+1}\}$ of dual multipliers; note that λ^{k+1} is calculated as a result of the kth augmented Lagrangian subproblem;

(iii) the sequence $\{(\eta^k, s^k)\}$ of optimal multipliers for the constraints $Ey = f$ and $y \in C$ in the sequence of augmented Lagrangian subproblems.

Because (18) and (20) each have a Slater point, strong duality holds and there exists a primal-dual solution (y, λ, η, s) that satisfies $s^T y = 0$. The kth augmented Lagrangian problem (with fixed σ) is

$$\min \left\{ c^T y + (\lambda^k)^T (b - Ay) + \frac{\sigma}{2} \|b - Ay\|^2 \ \mid Ey = f,\ y \in C \right\}, \tag{26}$$

and its dual (see [16] for QP duality in the case of $\mathbb{R}^q_+$) can be stated as

$$\begin{aligned} \max \quad & b^T \lambda^k + f^T \eta + \frac{\sigma}{2} \left(b^T b - v^T A^T A v \right) \\ \text{s.t.} \quad & A^T \left(\lambda^k + \sigma(b - Av) \right) + E^T \eta + s = c \\ & s \in C^*. \end{aligned} \tag{27}$$

Since (18) has a Slater point, so does (26). Furthermore, since A has full row rank, we can use a Slater point from (20) to construct such a point for (27). As a result, strong duality also holds between (26) and (27), and there exists a primal-dual solution (y, v, η, s) such that $v = y$ and $s^T y = 0$.

We first show that $Ay^k \to b$ via two lemmas and a proposition.

LEMMA 3.2. *Let $\bar{\lambda}$ and $\hat{\lambda}$ be arbitrary multipliers for $Ax = b$, and let $\bar{y}$ and $\hat{y}$ be optimal solutions of the corresponding augmented Lagrangian subproblems. Then*

$$\left(\bar{\lambda} - \hat{\lambda} \right)^T (A\bar{y} - A\hat{y}) \geq \sigma \|A\bar{y} - A\hat{y}\|^2.$$

Proof. Optimality of $\bar{y}$ with respect to $\bar{\lambda}$ implies

$$(c - A^T(\bar{\lambda} + \sigma(b - A\bar{y})))^T (y - \bar{y}) \geq 0$$

for all y such that $Ey = f, y \in C$. Likewise, for $\hat{y}$ and $\hat{\lambda}$:

$$(c - A^T(\hat{\lambda} + \sigma(b - A\hat{y})))^T (y - \hat{y}) \geq 0.$$

Applying these results with $y = \hat{y}$ and $y = \bar{y}$, respectively, summing the two resultant inequalities, and rearranging terms, we achieve the result. □

LEMMA 3.3. *Let (λ^*, η^*, s^*) be an optimal solution of* (20). *Then any $y \in X$ is optimal for the augmented Lagrangian subproblem corresponding to λ^* if and only if y is optimal for* (18).

Proof. Using dual feasibility, we have that

$$c^T y + (\lambda^*)^T (b - Ay) + \frac{\sigma}{2} \|b - Ay\|^2 = (\lambda^*)^T b + f^T \eta^* + (s^*)^T y + \frac{\sigma}{2} \|b - Ay\|^2.$$

Hence, ignoring the constant terms $b^T \lambda^* + f^T \eta^*$, the minimum value attainable by the augmented Lagrangian function is clearly bounded below by 0. Moreover, 0 is attained if and only if $(s^*)^T y = 0$ and $Ay = b$, which proves the result. □

PROPOSITION 3.4. *The sequence $\{Ay^k\}$ converges to b.*

Proof. Let (λ^*, η^*, s^*) be any optimal solution of (20), and let y^* be any optimal solution of (18). For all $k \geq 1$,

$$\begin{aligned} \|\lambda^{k+1} - \lambda^*\|^2 &= \|\lambda^k - \lambda^*\|^2 + 2\sigma(b - Ay^k)^T(\lambda^k - \lambda^*) + \sigma^2 \|b - Ay^k\|^2 \\ &\leq \|\lambda^k - \lambda^*\|^2 - 2\sigma^2 \|Ay^* - Ay^k\|^2 + \sigma^2 \|b - Ay^k\|^2 \\ &= \|\lambda^k - \lambda^*\|^2 - \sigma^2 \|b - Ay^k\|^2 \text{ (by Lemmas 3.2 and 3.3).} \end{aligned}$$

For arbitrary N, summing this inequality for $k = 1, \ldots, N$, we have

$$\begin{aligned} 0 &\le \sum_{k=1}^{N} \left(\|\lambda^k - \lambda^*\|^2 - \sigma^2 \|b - Ay^k\|^2 - \|\lambda^{k+1} - \lambda^*\|^2 \right) \\ &= \|\lambda^1 - \lambda^*\|^2 - \|\lambda^{N+1} - \lambda^*\|^2 - \sigma^2 \sum_{k=1}^{N} \|b - Ay^k\|^2, \end{aligned}$$

which implies $\sigma^2 \sum_{k=1}^{N} \|b - Ay^k\|^2 \le \|\lambda^1 - \lambda^*\|^2$. Hence, because N is arbitrary, Ay^k must converge to b. □

Now, with Proposition 3.4 and an additional lemma, we prove Theorem 3.1.

LEMMA 3.5. *For all* $k \ge 1$, y^k *and* $(\lambda^{k+1}, \eta^k, s^k)$ *are primal-dual optimal solutions of*

$$\min \left\{ c^T y \mid Ay = Ay^k,\ Ey = f,\ y \in C \right\}, \tag{28}$$

$$\max \left\{ (Ay^k)^T \lambda + f^T \eta \mid A^T \lambda + E^T \eta + s = c,\ s \in C^* \right\}. \tag{29}$$

Proof. Clearly, y^k is feasible for (28). Moreover, strong duality between (26) and (27) implies that (λ^k, η^k, s^k) is feasible for (27) such that $(s^k)^T y^k = 0$. Combining this with the definition $\lambda^{k+1} := \lambda^k + \sigma(b - Ay^k)$, we see that $(\lambda^{k+1}, \eta^k, s^k)$ is feasible for (29) and that strong duality holds between (28) and (29). This proves the result. □

Proof of Theorem 3.1. By Lemma 3.5, for each $k \ge 1$, $(y^k, \lambda^{k+1}, \eta^k, s^k)$ is a solution of the nonlinear system

$$\begin{aligned} &Ay = Ay^k,\ Ey = f,\ y \in C \\ &A^T \lambda + E^T \eta + s = c,\ s \in C^* \\ &y^T s = 0. \end{aligned}$$

By continuity, any accumulation point $(\bar{y}, \bar{\lambda}, \bar{\eta}, \bar{s})$ of $\{(y^k, \lambda^{k+1}, \eta^k, s^k)\}$ satisfies the above system with Ay^k replaced by its limit b (due to Proposition 3.4). In other words, $(\bar{y}, \bar{\lambda}, \bar{\eta}, \bar{s})$ is a primal-dual optimal solution for (18) and (20). □

4. Computational issues and results.

In this section, we discuss the implementation details of the augmented Lagrangian algorithm used to solve (1) and (2). We then demonstrate the effectiveness of this approach on various problem classes and also illustrate advantages of the augmented Lagrangian approach over other methods.

4.1. Optimizing over $N(K)$ and $N_+(K)$.

We have suggested in section 2 that one could consider a purely Lagrangian approach for solving the linear relaxation (1) since the calculation of $L(\lambda)$ is separable into $2n + 1$ LPs over the columns of the variables Y and Z. We have argued in section 3, however, that the augmented Lagrangian method has several advantages over the Lagrangian approach. In the case of (1), the kth augmented Lagrangian subproblem can be stated as

$$\min \quad \tilde{c}^T Y e_0 + (\lambda^k)^T h(Z, Y) + \frac{\sigma_k}{2} \|h(Z, Y)\|^2 \tag{30}$$

$$\text{s.t.} \quad Y e_i \in \hat{K} \quad \forall\ i = 0, 1, \ldots, n \tag{31}$$

$$Z e_i \in \hat{K} \quad \forall\ i = 1, \ldots, n \tag{32}$$

$$Y_{00} = 1. \tag{33}$$

An important observation is that, in contrast with $L(\lambda)$, (30)–(33) is a nonseparable convex QP. More precisely, the quadratic term in the objective (30) is the sole cause of the nonseparability.

Even with this complication, we still advocate the use of the augmented Lagrangian method. To exploit the structure inherent in the constraints (31)–(33), we propose to employ block coordinate descent for solving the subproblem, iteratively taking a descent step over a particular column of Y or Z, while keeping all other columns fixed. Block coordinate descent is known to be a convergent method; see Proposition 2.7.1 in [6].

When the amount of coupling between the blocks is small, block coordinate descent can be expected to converge quickly. One can observe that, because of the particular structure of $h(Y,Z)$, the amount of coupling between the columns of Y and Z is relatively small. In particular, the greatest amount of coupling is between Ye_i, Ze_i, and Ye_0 for $i = 1,\ldots,n$. As a result, we expect block coordinate descent to be a good choice for optimizing (30)–(33).

For optimizing over $N_+(K)$, we also advocate the augmented Lagrangian method, but at first glance, it is not clear how to handle the constraint that Y be positive semidefinite. This constraint is difficult not only because it involves positive semidefiniteness but also because it links the columns of Y. To handle this constraint, we follow the suggestions laid out in section 3.2. In particular, we introduce an "excess" variable U, which is required to be symmetric positive semidefinite and must also satisfy $U = Y$, and simultaneously drop the positive semidefiniteness constraint on Y. After introducing a symmetric matrix S of Lagrange multipliers for the constraint $U = Y$, the resulting kth augmented Lagrangian subproblem becomes

$$\begin{array}{ll} \min & \tilde{c}^T Ye_0 + (\lambda^k)^T h(Z,Y) + \dfrac{\sigma_k}{2}\|h(Y,Z)\|^2 + S^k \bullet (U - Y) + \dfrac{\sigma_k}{2}\|U - Y\|_F^2 \\ \text{s.t.} & \text{(31)–(33)}, \;\; U \succeq 0. \end{array}$$

Again we propose to solve this problem using block coordinate descent. For example, we first fix U and solve a series of QPs as we did in the case (30)–(33), one for each column of Y and Z. We then proceed by fixing Y and Z and solving the subproblem over U, which is equivalent to

$$\min\left\{2\,\sigma_k^{-1}\,S^k \bullet (U - Y) + \|U - Y\|_F^2 \;\;|\;\; U \succeq 0\right\}.$$

By completing the square, we see that

$$2\,\sigma_k^{-1}\,S^k \bullet (U - Y) + \|U - Y\|_F^2 = \|\sigma_k^{-1} S^k + (U - Y)\|_F^2 - \sigma_k^{-2}\,S^k \bullet S^k,$$

so that the above minimization is equivalent to solving

$$\min\left\{\left\|\sigma_k^{-1} S^k + (U - Y)\right\|_F^2 \;\;|\;\; U \succeq 0\right\},$$

which in turn is solved explicitly by projecting $Y - \sigma_k^{-1} S^k$ onto the cone of symmetric positive semidefinite matrices.

It is well known that calculating the projection M_+ of a symmetric matrix M onto the cone of symmetric positive semidefinite matrices can be done by calculating the spectral decomposition $M = QDQ^T$ and then forming the matrix $M_+ = QD_+Q^T$, where D_+ is derived from D by replacing all negative diagonal entries with 0. However, in our case, the matrix that we project, $M = Y - \sigma_k^{-1} S^k$, is not symmetric.

Table 1

Number of variables and constraints in the descriptions of $N(K)$ and $N_+(K)$ that serve as the basis of the augmented Lagrangian algorithm. Note that m represents the number of linear constraints (including lower and upper bounds) present in $\hat{K}$.

	Variables		Constraints		
			Linear	Linear	SDP
	Y, Z	U	enforced	relaxed	
$N(K)$	$(2n+1)(n+1)$	0	$1+(2n+1)m$	$n\left(\frac{3}{2}n+\frac{5}{2}\right)$	0
$N_+(K)$	$(2n+1)(n+1)$	$\frac{1}{2}(n+1)(n+2)$	$1+(2n+1)m$	$n\left(\frac{3}{2}n+\frac{5}{2}\right)+(n+1)^2$	1

Nevertheless, by using the identity

$$\begin{aligned}\|U-M\|_F^2 &= \left\|U-\frac{1}{2}(M+M^T)-\frac{1}{2}(M-M^T)\right\|_F^2 \\ &= \left\|U-\frac{1}{2}(M+M^T)\right\|_F^2 - \left(U-\frac{1}{2}(M+M^T)\right)\bullet\left(M-M^T\right)+\frac{1}{4}\left\|M-M^T\right\|_F^2 \\ &= \left\|U-\frac{1}{2}(M+M^T)\right\|_F^2+\frac{1}{4}\left\|M-M^T\right\|_F^2,\end{aligned}$$

where the final equality follows from the fact that the dot product of a symmetric matrix and a skew-symmetric matrix is zero, we can easily see that the projection of M is equal to the projection of $(M+M^T)/2$, which is itself symmetric.

Note that since S is the dual multiplier for the equality $U = Y$, it is unrestricted. However, from basic duality, it is not difficult to see that S will be constrained to be positive semidefinite in the dual problem. An illustration of this is as follows. Consider the generic SDP

$$\min\left\{C\bullet Y \mid \mathcal{A}(Y)=b,\ Y-U=0,\ U\succeq 0\right\},$$

which has the dual SDP

$$\max\left\{b^Ty \mid \mathcal{A}^*(y)+S=C,\ -S\preceq 0\right\};$$

here, $\mathcal{A}$ is a generic linear operator and $\mathcal{A}^*$ its adjoint. Thus, without loss of generality, we may restrict each S^k to be positive semidefinite, enforcing this by projection after each dual update.

Before moving onto the description of the specifics of the implementation in the next subsection, we would like to give some sense of the actual size of the LPs and SDPs that we are proposing to solve with the augmented Lagrangian method. This is given in Table 1. In the table, the quantity m represents the number of linear constraints (including lower and upper bounds) present in $\hat{K}$.

4.2. Implementation details. The augmented Lagrangian algorithm for (1) and (2) has been implemented in ANSI C under the Linux operating system on a Pentium 4 having a 2.4 GHz processor and 1 GB of RAM. The pivoting algorithm of CPLEX 8.1 for convex QP [21] has been employed for solving the $2n+1$ quadratic subproblems encountered during block coordinate descent, and LAPACK [1] has been utilized for the spectral decompositions required when projecting onto the positive semidefinite cone.

The choice of a pivoting algorithm for the quadratic subproblems—as opposed to an interior-point algorithm—was motivated by the warm-start capabilities of pivoting algorithms. In particular, it is not difficult to see that the objective functions of the $2n + 1$ quadratic subproblems change only slightly between loops of block coordinate descent or between iterations of the augmented Lagrangian algorithm. As a result, the ability to warm-start from an advance basis has proven to be invaluable for speeding up the overall algorithm.

Although Theorem 3.1 indicates that it is theoretically not necessary to increase the penalty parameter σ during the course of the algorithm, we have found that it is indeed advantageous to increase σ in order to enhance convergence. Our update rule is as follows:

> *Penalty update rule.* Every 500 iterations, σ is increased by a factor of 10.

We consider this to be a fairly conservative update rule.

Practically speaking, during the course of the algorithm, one can expect the norm of the constraint violation—$\|h(Y^k, Z^k)\|$ in the case of (1) and $(\|h(Y^k, Z^k)\|^2 + \|Y^k - U^k\|_F^2)^{1/2}$ in the case of (2)—to decrease towards 0, which is an indication that the algorithm is converging. As a result, we implement the following overall stopping criterion:

> *Stopping criterion.* The augmented Lagrangian algorithm is terminated once primal iterates are calculated such that the corresponding constraint violation is less than 10^{-6}.

We remark that, during early experiments on a handful of instances, this criterion was not achieved due to numerical difficulties caused by a large value of σ—typically around 10^8 or higher. As a result, we have also implemented the following:

> *Alternate stopping criterion.* The augmented Lagrangian algorithm is terminated once σ grows larger than 10^8.

Another implementation detail is how accurately the augmented subproblem is solved in each iteration. Recall from section 3 that it is not theoretically necessary to solve each subproblem exactly, and in fact, we have found in practice that it often suffices to solve them fairly loosely. In the computational results presented in section 4.4, our goal is to highlight the quality and speed of dual bounds provided by (1) and (2), rather than to calculate optimal solutions of high precision. In light of this goal, we decided to "solve" each augmented Lagrangian subproblem by performing exactly one cycle of block coordinate descent.

4.3. Problems. We have chosen four classes of 0-1 integer programs to illustrate the performance of the augmented Lagrangian algorithm.

4.3.1. Maximum stable set. Given an undirected, simple graph G with vertex set $V = \{1, \ldots, n\}$ and edge set $E \subseteq V \times V$, the (unweighted) maximum stable set problem is

$$\max\left\{e^T x \mid x_i + x_j \leq 1,\ (i, j) \in E,\ x \in \{0, 1\}^n\right\}.$$

As mentioned in section 2, the maximum stable set problem has been studied extensively by Lovász and Schrijver, and a number of theoretical results are known which illustrate the strength of (1) and (2).

We have collected a total of 26 graphs for testing; a basic description of these problems can be seen in Table 4. All graphs were obtained from the Center for Discrete Mathematics and Theoretical Computer Science [15] and originated as test instances for the maximum clique problem in the Second DIMACS Implementation Challenge. As such, we actually use the complement graphs as instances of the maximum stable set problem.

4.3.2. Problem of Erdös and Turán. We consider a 0-1 integer programming formulation of a problem studied by Erdös and Turán: calculate the maximum size of a subset of numbers in $\{1, \ldots, n\}$ such that no three numbers are in arithmetic progression. This number is the optimal value of

$$\max \left\{ e^T x \mid x_i + x_j + x_k \leq 2,\ i + k = 2j,\ i < j < k,\ x \in \{0,1\}^n \right\}.$$

For a full discussion, we refer the reader to [14], which includes background on the problem as well as some active set approaches for approximately optimizing (2) in this case. In the computational results, we consider 10 instances for $n = 60, 70, \ldots, 150$. It is interesting to note that the number of constraints in $N(K)$ and $N_+(K)$ for $n = 150$ is approximately 1.7 million.

4.3.3. Market share. The market share instances from MIPLIB [33], markshare1 and markshare2, have proven to be very small yet challenging instances of mixed integer programs. They are not pure binary integer programs, so one cannot apply the lift-and-project operator directly. It is easy, however, to generate very similar problems using only 0-1 variables. We first generate an $m \times n$ matrix, A, exactly as is done for the market share instances (see [13]). We also define the vector b by $b_i = \lfloor \frac{1}{2} \sum_{j=1}^{n} A_{ij} \rfloor$ for each $i = 1, \ldots, m$. The resulting IP is

$$\begin{aligned} \min \ & \sum_{i=1}^{m} \left(b_i - \sum_{j=1}^{n} A_{ij} x_j \right) \\ & Ax \leq b \\ & x \in \{0,1\}^n. \end{aligned}$$

We generated two instances of these problems of the same sizes as markshare1 (6×50) and markshare2 (7×60).

4.3.4. Quadratic assignment. Besides 0-1 linear integer programs, the lift-and-project relaxations provided by Lovász and Schrijver can easily be applied to 0-1 QPs with linear constraints. A quadratic objective $x^T Q x$ in the original problem becomes the linear objective

$$\begin{pmatrix} 0 & 0 \\ 0 & Q \end{pmatrix} \bullet Y$$

in the lifted problem. Using this technique, we can also consider the quadratic assignment problem (QAP), which is a problem of this type arising in location theory.

Because of its difficulty, QAP has attracted a large amount of attention and study; see [36, 10] and the recent survey by Anstreicher [2]. One of the most common forms of the QAP is the Koopmans–Beckmann form: given an integer p and matrices

$A, B \in \mathbb{R}^{p \times p}$, the QAP is

$$\begin{aligned}
\min\ & \sum_{i=1}^{p}\sum_{j=1}^{p}\sum_{k=1}^{p}\sum_{l=1}^{p} a_{ij}b_{kl}x_{ik}x_{jl} \\
\text{s.t.}\ & \sum_{i=1}^{p} x_{ik} = 1 \quad \forall\ k = 1, \dots, p \\
& \sum_{k=1}^{p} x_{ik} = 1 \quad \forall\ i = 1, \dots, p \\
& x_{ik} \in \{0, 1\} \quad \forall\ i, k = 1, \dots, p.
\end{aligned}$$

We have taken 91 test instances from QAPLIB, and all instances in QAPLIB are in Koopmans–Beckmann form.

In the effort to solve QAP to optimality, a variety of dual bounds have been developed for QAP. In particular, we will compare with three bounds from the literature: (i) the Gilmore–Lawler bound (denoted GLB) [17, 28]; the LP bound (denoted KCCEB) found in [22]; and the semidefinite programming bound (denoted RSB) of Rendl and Sotirov [37]. It is known that KCCEB is stronger than GLB, and it has been observed that RSB is stronger than KCCEB. As one might expect, however, RSB requires the most time to compute, while GLB takes the least.

The bound KCCEB is based on the first-level reformulation-linearization technique of Sherali and Adams [38] applied to QAP. It is not difficult to see that this relaxation is equivalent to (1), and so the bound provided by (1) and KCCEB are theoretically equal, although slight differences are observed in practice due to computational differences such as the level of precision to which the relaxation is solved.

The derivation of RSB is based on similar lift-and-project ideas as (2). However, RSB selectively includes certain constraints that are implied by $N_+(K)$, while adding in additional constraints that are not implied by $N_+(K)$ (at least not implied explicitly). It is currently unclear whether one bound is theoretically stronger than the other, although our computational results in the next subsection show that the Lovász–Schrijver bound is stronger than RSB on all test instances.

One additional comment regarding QAP is in order. Since QAP has equality constraints and upper bounds on the variables are redundant, it is possible to show that the constraints (7) of (1) and (2) are implied by (6). As a result, in this instance it is unnecessary to introduce the variable Z, which has the benefit of reducing the number of quadratic subproblems in the block coordinate descent from $2n+1$ to $n+1$. We have implemented these savings in the code and remark that the calculation of KCCEB has taken this into account as well.

4.4. Results. We first provide a comparison of our implementation with existing methods on a few problems selected from our test instances. We feel that these comparisons provide a fair indication of the advantages of our method in terms of both bound quality and computation time. Next, we provide detailed information on the performance of our method on the four problem classes discussed above.

4.4.1. Comparison with linear and semidefinite solvers. We directly solved some instances of (1), i.e., optimization over $N(K)$, using the dual simplex LP algorithm of CPLEX 8.1 [21] (with both default pricing and steepest-edge pricing), enforcing a time limit of 15,000 seconds. We show these results in Table 2 together with the running times and bounds obtained by the augmented Lagrangian approach

Table 2

Comparisons of bounds achieved for optimization over $N(K)$ by CPLEX 8.1 *dual simplex (with default and steepest-edge pricing) and the augmented Lagrangian method. Timings (in seconds) are also given. When prefixed to a timing, the symbol ($\star$) indicates that the algorithm did not terminate within the* 15,000 *seconds allotted.*

	Bound			Time(s)		
	CPLEX		Auglag	CPLEX		Auglag
Name	Default	Steep		Default	Steep	
MANN_a9	18.0000	18.0000	18.0000	3	5	5
johnson8-2-4	9.3333	9.3333	9.3333	0	0	6
johnson8-4-4	23.3333	23.3333	23.3333	88	85	90
hamming6-2	32.0000	32.0000	32.0000	11	10	11
hamming6-4	21.3333	21.3333	21.3334	12	12	134
johnson16-2-4	40.0000	40.0000	40.0001	1,505	1,521	763
keller4	57.0000	57.0000	57.0001	4,745	5,059	4,910
hamming8-2	128.9971	128.9971	128.0000	$\star$ 15,000	$\star$ 15,000	640
san200_09_1	86.3989	82.3446	70.1613	$\star$ 15,000	$\star$ 15,000	4,840
san200_09_2	83.7750	87.7951	66.6667	$\star$ 15,000	$\star$ 15,000	3,484
san200_09_3	80.4699	83.8000	66.6678	$\star$ 15,000	$\star$ 15,000	2,538
Erdös–Turán ($n = 60$)	34.2857	34.2857	34.6688	4,126	2,421	354
Erdös–Turán ($n = 70$)	40.0000	40.0000	40.7446	$\star$ 15,000	14,119	778
Erdös–Turán ($n = 80$)	46.5922	46.4502	46.7352	$\star$ 15,000	$\star$ 15,000	1,686
Erdös–Turán ($n = 90$)	54.5405	54.1181	53.0720	$\star$ 15,000	$\star$ 15,000	2,550
Erdös–Turán ($n = 100$)	63.8962	63.4383	59.6474	$\star$ 15,000	$\star$ 15,000	2,810

applied to (1). Note that we include both stable set and Erdös–Turán instances and that the instances are ordered roughly in terms of increasing size. We were unable to solve larger, denser instances with CPLEX as these required more than the available memory. For very large problems, these results clearly indicate the ability of our method to obtain bounds for $N(K)$ in much less time than is required by standard LP solvers.

We also attempted to carry out some optimizations of $N_+(K)$ using standard semidefinite solvers, such as CSDP developed by Borchers [8], but found that all but the smallest of instances would require more than the 1 GB of available memory on our computer.

4.4.2. Comparison with subgradient methods. We implemented a subgradient approach for optimizing over $N(K)$ using the volume algorithm developed in [5], for which an open source framework is available from the COIN-OR repository [31]. We found that this subgradient implementation in the current context was sensitive to the choice of initial dual multipliers. Consequently, some experimentation was required to settle upon appropriate starting duals. In particular, we found that starting duals of all -1's performed much better than all 0's.

In Table 3, we compare the subgradient algorithm and our augmented Lagrangian algorithm initialized with the same starting duals. We report not only the final bound and total running time for both algorithms but also the time at which one algorithm surpasses the best bound of the other. For example, on the *brock*200_1 instance, in 2,052 seconds the augmented Lagrangian algorithm achieved the same bound that the subgradient algorithm achieved after 8,603 seconds. A quick summary of Table 3 is that the subgradient algorithm outperforms augmented Lagrangian on the Erdös–Turán instances, whereas augmented Lagrangian outperforms subgradient on the stable set instances.

We found that increasing the number of cycles of coordinate descent (recall that we use only one cycle in all computations in this paper) improved the convergence

TABLE 3

Comparisons of bounds achieved for optimization over $N(K)$ by the subgradient method and the augmented Lagrangian method, along with timings (in seconds). Also shown is the time at which one algorithm surpassed the best bound of the other.

	Bound		Time(s)		Surpass Time(s)	
Instance	Subgrad	Auglag	Subgrad	Auglag	Subgrad	Auglag
MANN_a9	18.0098	18.0000	11	8		8
brock200_1	69.3607	66.6668	8,603	6,175		2,052
brock200_2	68.1062	66.6667	18,672	13,951		5,607
brock200_3	68.8632	66.6667	13,420	9,773		3,801
brock200_4	68.0411	66.6668	9,136	8,378		3,596
c-fat200-1	66.7608	66.6668	22,995	26,652		14,295
c-fat200-2	66.8576	66.6668	27,681	23,859		12,090
c-fat200-5	67.5605	66.6672	28,285	15,937		7,103
hamming6-2	32.0119	32.0000	31	20		20
hamming6-4	21.3543	21.3334	212	155		103
hamming8-2	128.0129	128.0000	2,407	1,118		1,065
hamming8-4	86.8582	85.3333	41,376	22,398		8,219
johnson16-2-4	40.0239	40.0003	972	684		522
johnson8-2-4	9.3875	9.3333	12	7		5
johnson8-4-4	23.3814	23.3333	177	98		69
keller4	57.8079	57.0001	5,275	4,960		1,865
p_hat300-1	102.1317	100.0013	198,263	98,646		35,030
p_hat300-2	102.7619	100.0036	129,663	61,445		17,471
p_hat300-3	102.8840	100.0016	64,609	26,660		7,005
san200_07_1	72.2302	66.6667	12,224	7,550		2,054
san200_07_2	66.9081	66.6670	72,026	7,306		3,846
san200_09_1	70.0103	70.2033	14,733	4,746	11,193	
san200_09_2	66.9371	66.6667	33,923	3,462		1,363
san200_09_3	66.9232	66.6668	31,469	2,528		1,353
sanr200_0_7	69.7920	66.6667	14,061	7,920		2,695
sanr200_0_9	66.8795	66.6667	32,368	3,459		1,410
Erdös–Turán ($n = 60$)	34.3553	37.5214	508	863	123	
Erdös–Turán ($n = 70$)	40.1841	43.9156	589	1,591	202	
Erdös–Turán ($n = 80$)	45.9178	49.3888	1,468	3,052	488	
Erdös–Turán ($n = 90$)	51.5557	57.4332	2,169	3,927	714	
Erdös–Turán ($n = 100$)	57.3410	62.7492	3,273	5,458	1,312	
Erdös–Turán ($n = 110$)	63.1194	67.4054	5,858	9,752	2,337	
Erdös–Turán ($n = 120$)	68.7601	71.9690	7,066	16,852	3,629	
Erdös–Turán ($n = 130$)	74.5256	79.6999	14,909	19,356	6,113	
Erdös–Turán ($n = 140$)	80.1107	88.7548	18,311	35,119	6,833	
Erdös–Turán ($n = 150$)	85.9341	96.5941	24,155	85,492	8,078	

of the augmented Lagrangian algorithm on the Erdös–Turán instances. However, the resultant timings were still not competitive with the subgradient algorithm on these instances. We remark that such convergence issues were much less prevalent in the semidefinite computation using augmented Lagrangian, as Table 5 will demonstrate.

We also experimented with our own implementation of a subgradient method to optimize over $N_+(K)$ but found that standard stepsize strategies for subgradient methods did not seem appropriate for the positive semidefinite multiplier S. Consequently, convergence was difficult to achieve in this case.

4.4.3. Maximum stable set. In Table 4, we give the dual bounds obtained by the augmented Lagrangian algorithm for the maximum stable set instances. The bounds and times (in seconds) to achieve those bounds are listed under N for (1) and N_+ for (2). When prefixed to the bound, the symbol (‡) indicates that the alternate stopping criterion for our method was enforced, i.e., σ had grown too large before primal feasibility had been obtained.

Table 4

Results on the maximum stable set problem, comparing bounds achieved by the N and N_+ procedures. Timings (in seconds) are also given. When prefixed to the bound, the symbol (‡) indicates that the alternate stopping criterion was enforced.

						Bound		Time(s)	
Name	$\lvert V\rvert$	$\lvert E\rvert$	α	ϑ_+	ϑ	N	N_+	N —	N_+
brock200_1	200	5066	21	27.2	27.5	66.6668	27.9874	5,119	28,590
brock200_2	200	10024	12		14.2	66.6671	‡ 17.0805	11,174	67,302
brock200_3	200	7852	15		18.8	66.6670	‡ 20.7928	8,674	51,665
brock200_4	200	6811	17	21.1	21.3	66.6681	22.8004	6,765	43,433
c-fat200-1	200	18366	12	12.0	12.0	66.6667	‡ 14.9735	18,125	126,103
c-fat200-2	200	16665	24	24.0	24.0	66.6686	24.0877	13,861	83,691
c-fat200-5	200	11427	58		60.3	66.6671	58.1798	16,774	44,483
hamming6-2	64	192	32	32.0	32.0	32.0000	32.0000	11	15
hamming6-4	64	1312	4	4.0	5.3	21.3334	‡ 4.5460	134	1,416
hamming8-2	256	1024	128	128.0	128.0	128.0000	128.0001	640	728
hamming8-4	256	11776	16	16.0	16.0	85.3340	‡ 20.5442	21,723	90,169
johnson8-2-4	28	168	4	4.0	4.0	9.3333	4.0052	6	59
johnson8-4-4	70	560	14	14.0	14.0	23.3333	14.0076	90	479
johnson16-2-4	120	1680	8	8.0	8.0	40.0001	‡ 10.2637	763	3,140
keller4	171	5100	11	13.5	14.0	57.0001	‡ 15.4119	4,910	19,319
MANN-a9	45	72	16		17.5	18.0000	17.1790	5	50
p_hat300-1	300	33917	8	10.0	10.1	100.0003	‡ 18.6697	129,437	322,287
p_hat300-2	300	22922	25		27.0	100.0004	‡ 30.1066	83,142	244,428
p_hat300-3	300	11460	36		41.2	100.0008	43.3282	33,554	101,995
san200_07_1	200	5970	30	30.0	30.0	66.6672	30.7071	7,995	31,049
san200_07_2	200	5970	18	18.0	18.0	66.6670	‡ 20.0176	7,710	37,102
san200_09_1	200	1990	70	70.0	70.0	70.1613	70.5464	4,840	6,947
san200_09_2	200	1990	60	60.0	60.0	66.6667	60.7250	3,484	6,977
san200_09_3	200	1990	44	44.0	44.0	66.6678	44.4080	2,538	12,281
sanr200_07	200	6032	18	23.6	23.8	66.6672	24.9716	7,946	36,576
sanr200_09	200	2037	42		49.3	66.6667	49.3156	3,335	9,428

In order to gauge the quality of the bounds in Table 4, we also include the size α of the maximum stable set (either obtained from the literature or computed using the IP solver of CPLEX) as well as the Lovász ϑ number of the graph (obtained by the algorithm of Burer and Monteiro [9]) and Schrijver's strengthening ϑ_+ of ϑ (obtained from Kim Toh (personal communication)). Note that the value of ϑ_+ was not available for all instances. The numbers ϑ and ϑ_+ are polynomial-time computable upper bounds on α, which are obtained by solving two related semidefinite programs. Theoretically, the N_+ bound is at least as strong as ϑ_+, which is at least as strong as ϑ, but computationally, ϑ takes less time to compute than ϑ_+, which in turn takes much less time than the N_+ bound (a reasonable estimate is roughly one order of magnitude less).

The results indicate that, at least on the majority of problems in this sample of graphs, the computed N_+ bound is significantly tighter than the computed N bound. Moreover, it is a real challenge for the augmented Lagrangian algorithm to optimize the N_+ bound fully, which is evidenced by the fact that the computed value for the N_+ bound is actually higher than ϑ on most instances. This demonstrates the possibility for further improvement of our optimization technique, perhaps by more sophisticated guidelines for choosing the number of cycles of block coordinate descent or for updating σ.

Nevertheless, we stress that our overall intention is to show that (1) and (2) can be (approximately) solved for general 0-1 integer programs. Accordingly, Table 4 serves to demonstrate that, given a specific problem (such as the stable set problem),

Table 5

Results on the problem of Erdös and Turán, comparing bounds achieved by the N and N_+ procedures. Timings (in seconds) are also given. Lower and upper bounds (LB and UB) were achieved by running the IP solver of CPLEX 8.1 *for at most* 15,000 *seconds.*

			Bound		Time(s)	
n	LB	UB	N	N_+	N	N_+
60	19	19	34.6688	31.4183	354	653
70	20	20	40.7446	36.6120	778	1,062
80	22	27	46.7352	41.6060	1,686	1,676
90	24	32	53.0720	46.5339	2,550	3,164
100	25	37	59.6474	51.8541	2,810	4,295
110	27	44	65.8540	57.0635	5,387	6,773
120	30	48	71.1935	62.1836	10,932	9,659
130	32	55	77.2897	67.2867	11,023	13,459
140	32	60	82.9586	72.3553	21,164	18,239
150	32	66	89.5872	77.2238	34,180	23,450

our method allows one to compute the N and N_+ bounds and hence to evaluate their quality.

4.4.4. Problem of Erdös and Turán. Table 5 lists the results of our algorithm on the Erdös–Turán instances, and the details of the table are the same as for Table 4. In order to assess the quality of the computed N and N_+ bounds, we ran CPLEX's IP solver on each instance for at most 15,000 seconds and report the best lower bound (LB) and best upper bound (UB) achieved.

Three things are interesting to note. First, the N_+ bounds are a significant improvement over the bounds reported by Dash [14] (for example, Dash gives a bound of 87.6 for $n = 150$). Second, the times for computing the N and N_+ bounds are not dramatically different from one another, and in fact, on some of the largest problems, the N_+ bound actually takes less time to compute. Third, the upper bound calculated by CPLEX's IP solver (after 15,000 seconds) is significantly tighter than the computed N and N_+ bounds, which puts into perspective the time dedicated to calculating N and N_+. Even still, as with the stable set instances in the previous subsection, the results demonstrate that the augmented Lagrangian greatly improves our capabilities for actually computing the Lovász–Schrijver bounds.

4.4.5. Market share. Besides solving both the N and N_+ relaxations of our market share instances, we also ran the default CPLEX IP solver for 15,000 on each instance in order to obtain lower and upper bounds. The results appear in the following table (recall that these are minimization problems).

			Bound		Time(s)	
Instance	lb	ub	N	N_+	N	N_+
Markshare1 (6×50)	0	3	0.0000	0.0000	20	17
Markshare2 (7×60)	0	11	0.0000	0.0000	37	31

Since the problems are quite small, optimizing over N and N_+ is done very quickly. However, neither relaxation improves on the basic LP bound of 0. This demonstrates that lift-and-project operators are not always able to strengthen the LP relaxation of an IP.

4.4.6. Quadratic assignment. Tables 6 and 7 list our results on the quadratic assignment instances, and the details of the table are similar to Table 4, except for

Table 6

Results on the quadratic assignment problem (I)*, comparing gaps achieved by three previous bounding techniques and the two techniques of this paper, N and N_+. Timings (in seconds) for N and N_+ are also given. Gaps are calculated with respect to the feasible value listed, which is known to be optimal unless the symbol (†) is prefixed. When prefixed to the gap, the symbol (‡) indicates that the alternate stopping criterion was enforced.*

		Gap (%)					Time(s)	
Name	Feas Val	GLB	KCCEB	RSB	N	N_+	N	N_+
bur026a	5,426,670	2.05	1.29		1.19	0.19	17,929	60,397
bur026b	3,817,852	2.70	1.69		1.57	0.22	18,248	53,749
bur026c	5,426,795	2.11	1.21		1.08	0.18	17,932	67,918
bur026d	3,821,225	2.87	1.64		1.51	0.21	18,064	69,804
bur026e	5,386,879	1.48	0.97		0.86	‡ 0.20	18,530	71,917
bur026f	3,782,044	1.99	1.27		1.14	0.10	18,195	78,748
bur026g	10,117,172	1.37	0.61		0.51	0.15	19,214	73,082
bur026h	7,098,658	1.77	0.75		0.63	0.09	18,385	70,939
chr012a	9,552	24.15	1.09		0.00	0.00	330	352
chr012b	9,742	26.65	0.00		0.00	0.00	293	363
chr012c	11,156	28.50	4.28		0.00	0.00	376	410
chr015a	9,896	43.16	11.65		4.35	0.14	1,415	1,461
chr015b	7,990	41.76	11.94		0.01	0.00	1,467	1,140
chr015c	9,504	35.13	3.80		0.00	0.00	901	1,188
chr018a	11,098	38.92	9.56		3.28	0.00	3,357	3,947
chr018b	1,534	0.00	0.00		0.00	0.13	1,277	6,239
chr020a	2,192	1.92	1.19		1.00	0.18	3,878	6,307
chr020b	2,298	4.44	1.70		0.87	0.13	4,030	9,778
chr020c	14,142	39.18	7.68		0.05	0.02	5,046	6,812
chr022a	6,156	3.77	0.58		0.24	0.03	6,762	12,711
chr022b	6,194	4.17	0.65		0.26	0.19	7,070	18,377
chr025a	3,796	27.16	4.98		0.53	0.37	13,901	33,402
els019	17,212,548	30.45	5.45		2.00	0.04	5,879	9,821
esc016a	68	44.12	39.71	13.24	29.41	5.88	452	1,195
esc016b	292	24.66	6.16	1.37	4.79	0.68	474	1,103
esc016c	160	48.13	43.13	11.25	26.25	3.75	538	1,981
esc016d	16	81.25	75.00	50.00	75.00	18.75	397	1,520
esc016e	28	57.14	57.14	17.86	50.00	3.57	349	1,318
esc016g	26	53.85	53.85	23.08	46.15	3.85	351	1,315
esc016h	996	37.25	29.32	2.61	29.32	1.91	618	1,369
esc016i	14	100.00	100.00	35.71	100.00	14.29	160	1,601
esc016j	8	87.50	75.00	12.50	75.00	0.00	345	1,331
esc032a	† 130	73.08			69.23	20.77	10,503	143,084
esc032b	† 168	42.86			42.86	21.43	5,308	130,393
esc032c	† 642	45.48			40.65	4.05	15,223	114,053
esc032d	† 200	47.00			44.00	4.50	10,767	117,556
esc032e	2	100.00			100.00	0.00	498	143,593
esc032f	2	100.00			100.00	0.00	496	144,820
esc032g	6	100.00			100.00	‡ 0.00	506	107,683
esc032h	† 438	41.32			33.79	3.20	15,031	140,406
had012	1,652	7.02	2.00	0.54	1.82	0.00	363	244
had014	2,724	8.52	2.31	0.33	2.13	0.00	818	1,092
had016	3,720	9.73	4.49	0.56	4.30	0.13	1,396	2,551
had018	5,358	10.86	5.23	0.77	5.06	0.11	2,518	4,966
had020	6,922	10.92	5.13	0.53	4.98	0.16	4,119	8,835

the following: (i) the best-known feasible value for the QAP is listed such that, if the symbol (†) is *not* prefixed, then the feasible value is actually optimal, while (†) is present when the value is not known to be optimal; (ii) instead of listing dual bounds, we give optimality gaps, i.e.,

$$\text{gap} = \frac{\text{feas val} - \text{bound}}{\text{feas val}} \times 100\%.$$

Table 7

Results on the quadratic assignment problem (II)*, comparing gaps achieved by three previous bounding techniques and the two techniques of this paper, N and N_+. Timings (in seconds) for N and N_+ are also given. Gaps are calculated with respect to the feasible value listed, which is known to be optimal unless the symbol (†) is prefixed. When prefixed to the gap, the symbol (‡) indicates that the alternate stopping criterion was enforced.*

		Gap (%)					Time(s)	
Name	Feas Val	GLB	KCCEB	RSB	N	N_+	N	N_+
kra030a	88,900	23.10	15.00	12.86	14.51	2.50	30,618	128,883
kra030b	91,420	24.45	16.61	11.23	16.10	4.07	30,648	136,187
kra032	88,700	24.02		10.19	16.06	3.51	40,543	225,053
lipa020a	3,683	0.00	0.00		0.00	0.00	1,337	3,699
lipa020b	27,076	0.00			0.00	0.00	1,556	4,276
lipa030a	13,178	0.00	0.00		0.01	0.02	20,584	121,748
lipa030b	151,426	0.00			0.00	0.00	10,783	77,868
nug012	578	14.71	9.86	3.63	9.52	1.73	315	469
nug014	1,014	15.98		2.17	8.97	0.39	837	1,093
nug015	1,150	16.26	10.17	2.43	9.48	0.78	1,043	1,543
nug016a	1,610	18.39	11.86	2.48	11.49	0.75	1,456	2,421
nug016b	1,240	17.58	12.74	4.19	12.26	1.69	1,338	2,143
nug017	1,732	19.86	13.51	3.64	13.05	1.44	1,932	3,379
nug018	1,930	19.48	14.20	4.04	13.89	1.92	2,393	4,932
nug020	2,570	19.96	15.45	4.63	15.14	2.49	3,772	7,577
nug021	2,438	24.82	17.64	4.72	17.10	2.46	5,150	13,791
nug022	3,596	30.95	21.19	4.34	20.88	2.34	6,110	18,074
nug024	3,488	23.28	18.09	5.10	17.75	2.61	8,581	25,887
nug025	3,744	23.37	18.16	5.58	17.76	3.29	10,457	31,082
nug027	5,234	29.29		5.14	21.63	2.33	14,406	69,574
nug028	5,166	26.71		5.13	21.58	2.92	16,501	81,564
nug030	6,124	25.39	21.86	5.24	21.60	3.10	21,762	127,011
rou012	235,528	14.12	5.09	5.03	4.79	0.11	566	759
rou015	354,210	15.71	8.64	5.91	8.30	1.13	1,377	2,027
rou020	725,522	17.31	11.59	8.50	11.36	4.19	5,112	10,997
scr012	31,410	11.31	5.96	6.65	5.07	0.00	489	496
scr015	51,140	12.52	5.07	4.51	3.70	0.00	1,403	1,197
scr020	110,030	30.23	14.12	13.66	13.58	3.88	5,182	10,564
ste036a	9,526	25.22	17.49		16.80	5.31	68,877	427,884
ste036b	15,852	45.41			30.65	7.61	73,431	358,981
ste036c	8,239,110	22.40			14.70	‡ 3.92	89,618	377,109
tai012a	224,416	12.70	1.61	0.73	1.02	0.00	563	414
tai012b	39,464,925	75.20	22.66		‡ 20.04	‡ 1.01	845	1,039
tai015a	388,214	15.64	9.34	6.04	9.12	2.86	1,402	2,022
tai015b	51,765,268	78.28	0.52		0.53	0.35	2,070	3,199
tai017a	491,812	16.08	10.23	8.23	10.01	3.11	2,521	4,414
tai020a	703,482	17.46	12.34	9.41	12.11	4.52	5,056	10,418
tai020b	122,455,319	88.40	24.44		‡ 23.06	‡ 4.11	7,981	15,591
tai025a	1,167,256	17.55	13.82	10.79	14.30	4.66	13,382	39,565
tai025b	344,355,646	86.15	56.93		‡ 55.82	‡ 11.70	16,855	65,262
tai030a	1,818,146	17.24	13.91	9.13	13.74	6.12	27,299	155,797
tai030b	† 637,117,113	93.57	78.46		‡ 78.46	‡ 18.47	31,333	247,114
tai035a	† 2,422,002	19.44	16.66		16.52	8.48	60,604	329,608
tai035b	† 283,315,445	88.49	64.05		‡ 66.40	‡ 15.42	63,888	430,914
tho030	149,936	39.59	33.40	9.26	32.84	4.75	28,180	99,265

A missing entry from the table (applicable only in the case of KCCEB and RSB) indicates that the gap was not available in the literature. In addition, note that the number contained in the names of the QAP instances is the size n of the QAP; for example, the instance $bur026a$ has $n = 26$.

Though theory predicts that the KCCEB and N gaps should equal one another, we do see some discrepancies in Tables 6 and 7, probably because of numerical differences in the algorithms. Typically, N is slightly better than KCCEB, but in compar-

TABLE 8

Results on the quadratic assignment problem (III)*, comparing gaps achieved by the bounding technique of Hahn et al.* [20] *(denoted by H) and the semidefinite technique of this paper, N_+. The problem instances are a subset of those shown in Tables* 6 *and* 7*.*

	Gap (%)	
Name	H	N_+
had16	0.00	0.13
had18	0.00	0.11
had20	0.00	0.16
kra30a	2.98	2.50
kra30b	4.72	4.07
nug12	0.00	1.73
nug15	0.00	0.78
nug20	3.23	2.49

	Gap (%)	
Name	H	N_+
nug30	6.11	3.10
rou15	0.00	1.13
rou20	3.60	4.19
tai20a	3.93	4.52
tai25a	6.48	4.66
tai30a	7.25	6.12
tho30	8.82	4.75

TABLE 9

Average number of iterations of the augmented Lagrangian algorithm over all problems on three of the four problem classes, for both $N(K)$ and $N_+(K)$.

	Stable	Erdös–Turán	Market Share	QAP
$N(K)$	371	760	251	2,796
$N_+(K)$	1,648	1,197	205	2,812

ison with timings reported in [22], the calculation of N takes more time. We should point out, however, that the algorithm used to compute KCCEB exploits the structure of QAP to a great extent (more than just the reduction of $2n+1$ subproblems to $n+1$ mentioned previously) and does not appear to be generalizable to other 0-1 problems. On the other hand, the augmented Lagrangian method can be applied to any 0-1 problem.

The tables also indicate that the N_+ gap is significantly tighter than the RSB gap. In fact, on a number of relatively small problems, the N_+ gap is 0, indicating that (2) solves the QAP exactly. Previous to these results, RSB had provided the strongest known bounds for problems in QAPLIB. Only partial timing results are given by Rendl and Sotirov [37], and so we are unable to make precise timing comparisons with RSB.

After the initial appearance of this paper, Hahn (personal communication) announced bounds for several of the instances in Tables 6 and 7, which were calculated with an algorithm of Hahn et al. [20] (but were not specifically presented there). The bounds are obtained by solving the second-level Sherali–Adams linear program for the QAP. We present the bounds compared with our semidefinite bounds in Table 8.

4.4.7. Some further details. The tables of the previous subsections include bounds and timings for the augmented Lagrangian runs. Some additional aggregate information on the number of iterations is given in Table 9. Here, "number of iterations" refers to the number of outermost loops in the augmented Lagrangian algorithm—indexed by k in the statement of Algorithm 1.

For problems with m much larger than n, the most computationally intensive part of the augmented Lagrangian algorithm (applied to both $N(K)$ and $N_+(K)$) is using CPLEX to solve the convex QP subproblems corresponding to the columns of Y and Z. We consider all stable and Erdös–Turán instances, which have $n \leq 300$ and often $m \approx \mathcal{O}(n^2)$, to be of this type. On the other hand, in the case of $N_+(K)$, the $\mathcal{O}(n^3)$ eigenvalue decompositions required by the semidefinite projections (two per iteration) will constitute a large part of the computation, especially for large n and small m. For example, for the QAP instances, n ranges from 144 to 1,225 and

$m = 3n + 2$. This helps to explain the fact that, while the average number of QAP iterations for $N(K)$ and $N_+(K)$ shown in Table 9 are not significantly different, the corresponding timings in Tables 6 and 7 are quite different.

5. Conclusions. In this paper, we propose a novel method to apply dual decomposition to the lift-and-project relaxations of binary integer programs introduced by Lovász and Schrijver [32]. We believe that this is some of the first work that focuses on developing effective tools for solving these very large relaxations. Rather than using subgradient techniques to solve the dual, we show how to use an augmented Lagrangian technique to obtain bounds from these relaxations in both the LP and semidefinite case. We extend a result by Poljak and Tret′jakov [35] to show that in the case of linear, conic programs, the augmented Lagrangian approach can use a constant penalty parameter and still guarantee convergence. Through extensive computational testing, we demonstrate the ability of this technique to outperform standard LP, SDP, and subgradient methods for various classes of problems. For some instances, such as QAP, the bounds computed from these relaxations are the tightest known to date.

As part of our future work in this area, we will study the possibility of using special purpose algorithms to solve the QP subproblems, especially in cases such as QAP where the constraints of the subproblems are simply a homogenization of the assignment polytope. We also intend to examine how these techniques may be used to yield tight relaxations of problems with a mix of binary and continuous variables and of continuous nonconvex QP's.

In addition to introducing some of the first effective solution techniques for linear and positive semidefinite lift-and-project relaxations, the success of this approach also demonstrates the applicability of augmented Lagrangian techniques even for linear, conic problems. We believe it will be interesting to investigate how well this technique performs on other large-scale linear, conic problems with block-angular structure.

Acknowledgments. The authors are in debt to two anonymous referees for careful and thorough comments that have greatly improved the paper.

REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, 3rd ed., Society for Industrial and Applied Mathematics, Philadelphia, 1999.

[2] K. M. Anstreicher, *Recent advances in the solution of quadratic assignment problems*, Math. Program., 97 (2003), pp. 27–42.

[3] E. Balas, *Disjunctive programming*, Ann. Discrete Math., 5 (1979), pp. 3–51.

[4] E. Balas, S. Ceria, and G. Cornuéjols, *A lift-and-project cutting plan algorithm for mixed* 0–1 *programs*, Math. Program., 58 (1993), pp. 295–324.

[5] F. Barahona and R. Anbil, *The volume algorithm: Producing primal solutions with a subgradient method*, Math. Program., 87 (2000), pp. 385–399.

[6] D. P. Bertsekas, *Nonlinear Programming*, 1st ed., Athena Scientific, Belmont, MA, 1995.

[7] D. Bienstock and M. Zuckerberg, *Subset algebra lift operators for* 0–1 *integer programming*, SIAM J. Optim., 15 (2004), pp. 63–95.

[8] B. Borchers, *CSDP, a C library for semidefinite programming*, Optim. Methods Softw., 11/12 (1999), pp. 613–623.

[9] S. Burer and R. Monteiro, *A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization*, Math. Program., 95 (2003), pp. 329–357.

[10] R. E. Burkard, S. Karisch, and F. Rendl, *QAPLIB—a quadratic assignment problem library*, J. Global Optim., 10 (1997), pp. 391–403.

[11] S. Ceria and G. Pataki, *Solving integer and disjunctive programs by lift-and-project*, in Proceedings of the Sixth International IPCO Conference, Lecture Notes in Comput. Sci. 1412, R. E. Bixby, E. A. Boyd, and R. Z. Rios-Mercato, eds., 1998, pp. 271–283.

[12] W. Cook and S. Dash, *On the matrix-cut rank of polyhedra*, Math. Oper. Res., 26 (2001), pp. 19–30.

[13] G. Cornuéjols and M. Dawande, *A class of hard small* 0–1 *programs*, INFORMS J. Comput., 11 (1999), pp. 205–210.

[14] S. Dash, *On the Matrix Cuts of Lovász and Schrijver and Their Use in Integer Programming*, Ph.D. thesis, Rice University, Houston, TX, 2001.

[15] See the website: http://dimacs.rutgers.edu/Challenges/.

[16] W. S. Dorn, *Duality in quadratic programming*, Quart. Appl. Math., 18 (1960/1961), pp. 155–162.

[17] P. C. Gilmore, *Optimal and suboptimal algorithms for the quadratic assignment problem*, J. Soc. Indust. Appl. Math., 10 (1962), pp. 305–313.

[18] M. X. Goemans and L. Tunçel, *When does the positive semidefiniteness constraint help in lifting procedures?*, Math. Oper. Res., 26 (2001), pp. 796–815.

[19] R. E. Gomory, *An algorithm for integer solutions to linear programs*, in Recent Advances in Mathematical Programming, R. Graves and P. Wolfe, eds., McGraw-Hill, New York, 1963, pp. 269–302.

[20] P. M. Hahn, W. L. Hightower, T. A. Johnson, M. Guignard-Spielberg, and C. Roucairol, *A level-*2 *reformulation-linearization techinque bound for the quadratic assignment problem*, manuscript, University of Pennsylvania, Philadelphia, PA, 2001.

[21] ILOG, Inc., *ILOG CPLEX* 8.1*, User's Manual*, 2002.

[22] S. E. Karisch, E. Çela, J. Clausen, and T. Espersen, *A dual framework for lower bounds of the quadratic assignment problem based on linearization*, Computing, 63 (1999), pp. 351–403.

[23] K. C. Kiwiel, *Methods of Descent for Nondifferentiable Optimization*, Springer, Berlin, 1985.

[24] J. B. Lasserre, *An explicit exact SDP relaxation for nonlinear* 0–1 *programs*, in Lecture Notes in Comput. Sci., 2081, K. Aardal and A. M. H. Gerards, eds., 2001, pp. 293–303.

[25] M. Laurent, *Tighter linear and semidefinite relaxations for max-cut based on the Lovász–Schrijver lift-and-project procedure*, SIAM J. Optim., 12 (2001/2002), pp. 345–375.

[26] M. Laurent, *A comparison of the Sherali–Adams, Lovász–Schijver, and Lasserre relaxation for* 0–1 *programming*, SIAM J. Optim., 28 (2003), pp. 470–496.

[27] M. Laurent and F. Rendl, *Semidefinite programming and integer programming*, Technical report PNA-R0210, CWI, Amsterdam, April 2002. To appear as chapter of the *Handbook on Discrete Optimization* edited by K. Aardal, G. Nemhauser and R. Weismantel.

[28] E. L. Lawler, *The quadratic assignment problem*, Management Sci., 9 (1962/1963), pp. 586–599.

[29] C. Lemaréchal, *Nonsmooth optimization and descent methods*, Technical report, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1977.

[30] L. Lipták and L. Tunçel, *The stable set problem and the lift-and-project rank of graphs*, Math. Program., 98 (2003), pp. 319–353.

[31] R. Lougee-Heimer, *The Common Optimization INterface for Operations Research*, IBM Journal of Research and Development, 47 (2003), pp. 57–66; also available online from www.coin-or.org.

[32] L. Lovász and A. Schrijver, *Cones of matrices and set-functions, and* 0–1 *optimization*, SIAM J. Optim., 1 (1991), pp. 166–190.

[33] MIPLIB, 2003. http://miplib.zib.de/.

[34] J. Nocedal and S. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.

[35] B. T. Poljak and N. V. Tret′jakov, *A certain iteration method of linear programming and its economic interpretation*, Èkonom. i Mat. Metody, 8 (1972), pp. 740–751.

[36] QAPLIB, http://www.seas.upenn.edu/qaplib/.

[37] F. Rendl and R. Sotirov, *Bounds for the quadratic assignment problem using the bundle method*, Technical report, Department of Mathematics, University of Klagenfurt, Klagenfurt, Austria, 2003.

[38] H. D. Sherali and W. P. Adams, *A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems*, SIAM J. Discrete Math., 3 (1990), pp. 411–430.

[39] H. D. Sherali and W. P. Adams, *A Reformulation-Linearization Technique (RLT) for Solving Discrete and Continuous Nonconvex Problems*, Kluwer, Dordrecht, The Netherlands, 1997.

[40] H. D. Sherali, B. Özdaryal, W. P. Adams, and N. Attia, *On using exterior penalty approaches for solving linear programming problems*, Comput. Oper. Res., 28 (2001), pp. 1049–1074.

[41] T. Stephen and L. Tunçel, *On a representation of the matching polytope via semidefinite liftings*, Math. Oper. Res., 24 (1999), pp. 1–7.

SIAM J. Optim.
Vol. 16, No. 3, pp. 751–765

A SUM OF SQUARES APPROXIMATION OF NONNEGATIVE POLYNOMIALS[*]

JEAN B. LASSERRE[†]

Abstract. We show that every real nonnegative polynomial f can be approximated as closely as desired (in the l_1-norm of its coefficient vector) by a sequence of polynomials $\{f_\epsilon\}$ that are sums of squares. The novelty is that each f_ϵ has a simple and explicit form in terms of f and ϵ.

Key words. real algebraic geometry, positive polynomials, sum of squares, semidefinite programming

AMS subject classifications. 12E05, 12Y05, 90C22

DOI. 10.1137/04061413X

1. Introduction.

The study of relationships between *nonnegative* and *sums of squares* (s.o.s.) polynomials, initiated by Hilbert, is of real practical importance in view of numerous potential applications, notably in polynomial programming. Indeed, checking whether a given polynomial is nonnegative is a NP-hard problem whereas checking whether it is s.o.s. reduces to solving a (convex) semidefinite programming (SDP) problem for which efficient algorithms are now available. (For instance, it is known that up to an a priori fixed precision, an SDP is solvable in time polynomial in the input size of the problem.)

For instance, recent results in real algebraic geometry, most notably by Schmüdgen [16], Putinar [13], Jacobi and Prestel [5], have provided s.o.s. representations of polynomials, positive on a compact semialgebraic set; the interested reader is referred to Prestel and Delzell [12] and Scheiderer [15] for a nice account of such results. This in turn has permitted the development of efficient SDP-relaxations in polynomial optimization (see, e.g., Lasserre [6, 7, 8], Parrilo [10, 11], Schweighofer [17], and the many references therein).

So, back to a comparison between nonnegative and s.o.s. polynomials, on the negative side, Blekherman [4] has shown that if a degree > 2 is *fixed* (and for a large fixed number of variables), then the cone of nonnegative polynomials is much *larger* than that of s.o.s. However, on the positive side, a denseness result [2] states that the cone of s.o.s. polynomials is *dense* in the space of polynomials that are nonnegative on $[-1, 1]^n$ (for the l_1-norm $\|f\|_1 = \sum_\alpha |f_\alpha|$ whenever f is written $\sum_\alpha f_\alpha x^\alpha$ in the usual canonical basis); see, e.g., Berg, Christensen, and Ressel [2, Theorem 9.1, p. 273]).

Contribution. We show that *every nonnegative* polynomial f is almost a s.o.s., namely we show that f can be approximated by a sequence of s.o.s. polynomials $\{f_\epsilon\}_\epsilon$, in the specific form

$$f_\epsilon \;=\; f + \epsilon \sum_{k=0}^{r(f,\epsilon)} \sum_{j=1}^{n} \frac{x_j^{2k}}{k!}, \tag{1.1}$$

for some $r(f, \epsilon) \in \mathbb{N}$, so that $\|f - f_\epsilon\|_1 \to 0$ as $\epsilon \downarrow 0$. (Notice that in (1.1), one may replace $r(f, \epsilon)$ with any $r \geq r(f, \epsilon)$ and still get the same result.)

[*] Received by the editors August 30, 2004; accepted for publication (in revised form) June 17, 2005; published electronically January 6, 2006.

http://www.siam.org/journals/siopt/16-3/61413.html

[†] LAAS-CNRS and Institute of Mathematics, LAAS 7 Avenue du Colonel Roche, 31077 Toulouse Cédex 4, France (lasserre@laas.fr).

This result is in the spirit of the previous denseness result. However, we provide in (1.1) an *explicit* converging approximation with a very specific (and simple) form; namely, it suffices to slightly perturbate f by adding a small coefficient $\epsilon > 0$ to each square monomial x_i^{2k} for all $i = 1, \dots, n$ and all $k = 0, 1, \dots, r$, with r sufficiently large. To prove this result we combine the following:

- a (generalized) Carleman's sufficient condition (due to Nussbaum [9]) for a moment sequence $\mathbf{y} = \{y_\alpha\}$ to have a unique *representing measure* μ (i.e., such that $y_\alpha = \int x^\alpha d\mu$ for all $\alpha \in \mathbb{N}^n$), and
- a duality result from convex optimization.

As a consequence, we may thus define a procedure to approximate the global minimum of a polynomial f, at least when there is a global minimizer x^* that satisfies $\|x^*\|_\infty \leq M$ for some known M. It consists of solving a sequence of SDP-relaxations which are simpler and easier to solve than those defined in Lasserre [6]; see section 3.

Finally, we also consider the case where f is a *convex* polynomial, nonnegative on a convex semialgebraic set $\mathbb{K}$ defined by (concave polynomial) inequalities $g_j \geq 0$. We show that the approximation f_ϵ of f, defined in (1.1), has a certificate of positivity on $\mathbb{K}$ (or a representation) similar to Putinar's s.o.s. representation [13], but in which the s.o.s. polynomial coefficients of the g_j's now become simple nonnegative *scalars*, the Lagrange multipliers of a related convex optimization problem.

2. Notation and definitions. For a real symmetric matrix A, the notation $A \succeq 0$ (resp., $A \succ 0$) stands for A positive semidefinite (resp., positive definite). The sup-norm $\sup_j |x_j|$ of a vector $x \in \mathbb{R}^n$, is denoted by $\|x\|_\infty$. Let $\mathbb{R}[x_1, \dots, x_n]$ be the ring of real polynomials, and let

$$v_r(x) := (1, x_1, x_2, \dots, x_n, x_1^2, x_1x_2, \dots, x_1x_n, x_2^2, x_2x_3, \dots, x_n^r) \tag{2.1}$$

be the canonical basis for the $\mathbb{R}$-vector space $\mathcal{A}_r$ of real polynomials of degree at most r, and let $s(r)$ be its dimension. Similarly, $v_\infty(x)$ denotes the canonical basis of $\mathbb{R}[x_1, \dots, x_n]$ as a $\mathbb{R}$-vector space, denoted $\mathcal{A}$. So a vector in $\mathcal{A}$ always has *finitely* many nonzero entries.

Therefore, a polynomial $p \in \mathcal{A}_r$ is written

$$x \mapsto p(x) \;=\; \sum_{\alpha \in \mathbb{N}^n} p_\alpha x^\alpha \;=\; \langle \mathbf{p}, v_r(x) \rangle, \qquad x \in \mathbb{R}^n,$$

(where $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$) for some vector $\mathbf{p} = \{p_\alpha\} \in \mathbb{R}^{s(r)}$, the vector of coefficients of p in the basis (2.1).

Extending $\mathbf{p}$ with zeros, we can also consider $\mathbf{p}$ as a vector indexed in the basis $v_\infty(x)$ (i.e., $\mathbf{p} \in \mathcal{A}$). If we equip $\mathcal{A}$ with the usual scalar product $\langle ., . \rangle$ of vectors, then for every $p \in \mathcal{A}$,

$$p(x) \;=\; \sum_{\alpha \in \mathbb{N}^n} p_\alpha x^\alpha \;=\; \langle \mathbf{p}, v_\infty(x) \rangle, \qquad x \in \mathbb{R}^n.$$

Given a sequence $\mathbf{y} = \{y_\alpha\}$ indexed in the basis $v_\infty(x)$, let $L_{\mathbf{y}} : \mathcal{A} \to \mathbb{R}$ be the linear functional

$$p \mapsto L_{\mathbf{y}}(p) \;:=\; \sum_{\alpha \in \mathbb{N}^n} p_\alpha y_\alpha \;=\; \langle \mathbf{p}, \mathbf{y} \rangle.$$

Given a sequence $\mathbf{y} = \{y_\alpha\}$ indexed in the basis $v_\infty(x)$, the *moment* matrix $M_r(\mathbf{y}) \in \mathbb{R}^{s(r)\times s(r)}$ with rows and columns indexed in the basis $v_r(x)$ in (2.1), satisfies

$$[M_r(\mathbf{y})(1,j) \,=\, y_\alpha \text{ and } M_r(y)(i,1) \,=\, y_\beta] \,\Rightarrow\, M_r(y)(i,j) \,=\, y_{\alpha+\beta}.$$

For instance, with $n = 2$,

$$M_2(\mathbf{y}) \,=\, \left[\begin{array}{cccccc} y_{00} & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\ y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\ y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} \end{array}\right].$$

A sequence $\mathbf{y} = \{y_\alpha\}$ has a *representing* measure $\mu_\mathbf{y}$ if

$$y_\alpha \,=\, \int_{\mathbb{R}^n} x^\alpha \, \mu_\mathbf{y}(dx) \qquad \forall\, \alpha \in \mathbb{N}^n. \tag{2.2}$$

In this case one also says that $\mathbf{y}$ is a *moment sequence.* In addition, if $\mu_\mathbf{y}$ is unique, then $\mathbf{y}$ is said to be a *determinate* moment sequence.

The matrix $M_r(\mathbf{y})$ defines a bilinear form $\langle .,. \rangle_\mathbf{y}$ on $\mathcal{A}_r$, by

$$\langle q,p\rangle_\mathbf{y} \,:=\, \langle \mathbf{q}, M_r(\mathbf{y})\mathbf{p}\rangle \,=\, L_\mathbf{y}(qp), \quad q,p \in \mathcal{A}_r,$$

and if $\mathbf{y}$ has a *representing* measure $\mu_\mathbf{y}$, then

$$\langle \mathbf{q}, M_r(\mathbf{y})\mathbf{q}\rangle \,=\, \int_{\mathbb{R}^n} q(x)^2 \, \mu_\mathbf{y}(dx) \,\geq\, 0, \tag{2.3}$$

so that $M_r(\mathbf{y}) \succeq 0$.

Next, given a sequence $\mathbf{y} = \{y_\alpha\}$ indexed in the basis $v_\infty(x)$, let $y_{2k}^{(i)} := L_\mathbf{y}(x_i^{2k})$ for every $i = 1,\ldots,n$ and every $k \in \mathbb{N}$. That is, $y_{2k}^{(i)}$ denotes the element in the sequence $\mathbf{y}$, corresponding to the monomial x_i^{2k}.

Of course not every sequence $\mathbf{y} = \{y_\alpha\}$ has a representing measure $\mu_\mathbf{y}$ as in (2.2). However, there exists a *sufficient* condition to ensure that it is the case. The following result is stated in Berg [3, Theorem 5, p. 117] is from Nussbaum [9], and is restated here, with our notation.

THEOREM 2.1. *Let* $\mathbf{y} = \{y_\alpha\}$ *be an infinite sequence such that* $M_r(\mathbf{y}) \succeq 0$ *for all* $r = 0, 1, \ldots$. *If*

$$\sum_{k=1}^{\infty} (y_{2k}^{(i)})^{-1/2k} \,=\, \infty, \qquad i = 1,\ldots,n, \tag{2.4}$$

then $\mathbf{y}$ *is a determinate moment sequence.*

The condition (2.4) in Theorem 2.1 is called *Carleman's condition* as it extends to the multivariate case the original Carleman's sufficient condition given for the univariate case.

3. Preliminaries. Let B_M be the closed ball

$$B_M \; = \; \{x \in \mathbb{R}^n \mid \quad \|x\|_\infty \; \leq \; M\}. \tag{3.1}$$

Proposition 3.1. *Let $f \in \mathbb{R}[x_1, \ldots, x_n]$ be such that $-\infty < f^* := \inf_x f(x)$. Then, for every $\epsilon > 0$ there is some $M_\epsilon \in \mathbb{N}$ such that*

$$f^*_M \; := \; \inf_{x \in B_M} \; f(x) \; < \; f^* + \epsilon \qquad \forall M \; \geq \; M_\epsilon.$$

Equivalently, $f^*_M \downarrow f^*$ as $M \to \infty$.

Proof. Suppose it is false. That is, there is some $\epsilon_0 > 0$ and an infinite sequence $\{M_k\} \subset \mathbb{N}$, with $M_k \to \infty$, such that $f^*_{M_k} \geq f^* + \epsilon_0$ for all k. But let $x_0 \in \mathbb{R}^n$ be such that $f(x_0) < f^* + \epsilon_0$. With any $M_k \geq \|x_0\|_\infty$, one obtains the contradiction $f^* + \epsilon_0 \leq f^*_{M_k} \leq f(x_0) < f^* + \epsilon_0$. □

To prove our main result (Theorem 4.1 below), we first introduce the following related optimization problems:

$$\mathbb{P}: \qquad f^* \; := \; \inf_{x \in \mathbb{R}^n} \; f(x), \tag{3.2}$$

and for $0 < M \in \mathbb{N}$,

$$\mathcal{P}_M : \; \inf_{\mu \in \mathcal{P}(\mathbb{R}^n)} \left\{ \int f \, d\mu \mid \quad \int \sum_{i=1}^n \mathrm{e}^{x_i^2} \, d\mu \leq n \mathrm{e}^{M^2} \right\}, \tag{3.3}$$

where $\mathcal{P}(\mathbb{R}^n)$ is the space of probability measures on $\mathbb{R}^n$. The respective optimal values of $\mathbb{P}$ and $\mathcal{P}_M$ are denoted $\inf \mathbb{P} = f^*$ and $\inf \mathcal{P}_M$, or $\min \mathbb{P}$ and $\min \mathcal{P}_M$ if the infimum is attained.

Proposition 3.2. *Let $f \in \mathbb{R}[x_1, \ldots, x_n]$ be such that $-\infty < f^* := \inf_x f(x)$, and consider the two optimization problems $\mathbb{P}$ and $\mathcal{P}_M$ defined in* (3.2) *and* (3.3), *respectively. Then, $\inf \mathcal{P}_M \downarrow f^*$ as $M \to \infty$. If f has a global minimizer $x^* \in \mathbb{R}^n$, then $\min \mathcal{P}_M = f^*$ whenever $M \geq \|x^*\|_\infty$.*

Proof. Let $\mu \in \mathcal{P}(\mathbb{R}^n)$ be admissible for $\mathcal{P}_M$. As $f \geq f^*$ on $\mathbb{R}^n$ then it follows immediately that $\int f d\mu \geq f^*$, and so, $\inf \mathcal{P}_M \geq f^*$ for all M.

As B_M is closed and bounded, it is compact and so with f^*_M as in Proposition 3.1, there is some $\hat{x} \in B_M$ such that $f(\hat{x}) = f^*_M$. In addition let $\mu \in \mathcal{P}(\mathbb{R}^n)$ be the Dirac probability measure at the point $\hat{x}$. As $\|\hat{x}\|_\infty \leq M$,

$$\int \sum_{i=1}^n \mathrm{e}^{x_i^2} \, d\mu \; = \; \sum_{i=1}^n \mathrm{e}^{(\hat{x}_i)^2} \; \leq \; n\mathrm{e}^{M^2},$$

so that μ is an admissible solution of $\mathcal{P}_M$ with value $\int f \, d\mu = f(\hat{x}) = f^*_M$, which proves that $\inf \mathcal{P}_M \leq f^*_M$. This latter fact, combined with Proposition 3.1 and with $f^* \leq \inf \mathcal{P}_M$, implies $\inf \mathcal{P}_M \downarrow f^*$ as $M \to \infty$, the desired result. The final statement is immediate by taking as a feasible solution for $\mathcal{P}_M$, the Dirac probability measure at the point $x^* \in B_M$ (with $M \geq \|x^*\|_\infty$). As its value is now f^*, it is also optimal, and so $\mathcal{P}_M$ is solvable with optimal value $\min \mathcal{P}_M = f^*$. □

Proposition 3.2 provides a rationale for introducing the following SDP problems. Let $2r_f$ be the degree of f and for every $r_f \leq r \in \mathbb{N}$, consider the SDP problem

$$(3.4) \qquad \mathbb{Q}_r \left\{ \begin{array}{lll} & \min\limits_{\mathbf{y}} L_{\mathbf{y}}(f) \left(= \sum\limits_{\alpha} f_\alpha y_\alpha \right) & \\ \text{s.t.} & M_r(\mathbf{y}) & \succeq 0 \\ & \sum\limits_{k=0}^{r} \sum\limits_{i=1}^{n} \frac{y_{2k}^{(i)}}{k!} & \leq n e^{M^2}, \\ & y_0 & = 1, \end{array} \right.$$

and its associated *dual* SDP problem

$$(3.5) \qquad \mathbb{Q}_r^* \left\{ \begin{array}{lll} & \max\limits_{\lambda \geq 0, \gamma, q} \gamma - n e^{M^2} \lambda & \\ \text{s.t.} & f - \gamma & = q - \lambda \sum\limits_{k=0}^{r} \sum\limits_{j=1}^{n} \frac{x_j^{2k}}{k!} \\ & q & \text{s.o.s. of degree } \leq 2r, \end{array} \right.$$

with respective optimal values $\inf \mathbb{Q}_r$ and $\sup \mathbb{Q}_r^*$ (or $\min \mathbb{Q}_r$ and $\max \mathbb{Q}_r^*$ if the optimum is attained, in which case the problems are said to be solvable). For more details on SDP theory, the interested reader is referred to the survey paper [18].

The SDP problem $\mathbb{Q}_r$ is a relaxation of $\mathcal{P}_M$, and we next show that in fact

- $\mathbb{Q}_r$ is solvable for all $r \geq r_0$,
- its optimal value $\min \mathbb{Q}_r \to \inf \mathcal{P}_M$ as $r \to \infty$, and
- $\mathbb{Q}_r^*$ is also solvable with same optimal value as $\mathbb{Q}_r$, for every $r \geq r_f$.

This latter fact will be crucial to prove our main result in the next section. Let l_∞ (resp., l_1) be the Banach space of bounded (resp., summable) infinite sequences with the sup-norm (resp., the l_1-norm).

THEOREM 3.3. *Let $f \in \mathbb{R}[x_1, \ldots, x_n]$ be of degree $2r_f$, with global minimum $f^* > -\infty$, and let $M > 0$ be fixed. Then:*

(i) *For every $r \geq r_f$, $\mathbb{Q}_r$ is solvable, and* $\min \mathbb{Q}_r \uparrow \inf \mathcal{P}_M$ *as $r \to \infty$.*

(ii) *Let $\mathbf{y}^{(r)} = \{y_\alpha^{(r)}\}$ be an optimal solution of $\mathbb{Q}_r$ and complete $\mathbf{y}^{(r)}$ with zeros to make it an element of l_∞. Every (pointwise) accumulation point $\mathbf{y}^*$ of the sequence $\{\mathbf{y}^{(r)}\}_{r \in \mathbb{N}}$ is a determinate moment sequence, that is,*

$$(3.6) \qquad y_\alpha^* = \int_{\mathbb{R}^n} x^\alpha \, d\mu^*, \qquad \alpha \in \mathbb{N}^n,$$

for a unique probability measure μ^*, and μ^* is an optimal solution of $\mathcal{P}_M$.

(iii) *For every $r \geq r_f$,* $\max \mathbb{Q}_r^* = \min \mathbb{Q}_r$.

For a proof, see section 5.1.

So, one can approximate the optimal value f^* of $\mathbb{P}$ as closely as desired, by solving SDP-relaxations $\{\mathbb{Q}_r\}$ for sufficiently large values of r and M. Indeed, $f^* \leq \inf \mathcal{P}_M \leq f_M^*$, with f_M^* as in Proposition 3.1. Therefore, let $\epsilon > 0$ be fixed, arbitrary. By Proposition 3.2, we have $f^* \leq \inf \mathcal{P}_M \leq f^* + \epsilon$ provided that M is sufficiently large. Next, by Theorem 3.3(i), one has $\inf \mathbb{Q}_r \geq \inf \mathcal{P}_M - \epsilon$ provided that r is sufficiently large, in which case, we finally have $f^* - \epsilon \leq \inf \mathbb{Q}_r \leq f^* + \epsilon$.

For instance, if the infimum f^* is attained and one knows an upper bound M on $\|x^*\|_\infty$ for some global minimizer x^*, then the sequence of SDP-relaxations $\mathbb{Q}_r$

in (3.4) with M being fixed, will suffice. Notice that the SDP-relaxations $\mathbb{Q}_r$ are simpler than the one defined in Lasserre [6]. Both have the same variables $\mathbf{y} \in \mathbb{R}^{s(2r)}$, but the former has *one* SDP constraint $M_r(\mathbf{y}) \succeq 0$ and one scalar inequality (as one substitutes y_0 with 1) whereas the latter has the same SDP constraint $M_r(\mathbf{y}) \succeq 0$ and one additional SDP constraint $M_{r-1}(\theta \mathbf{y}) \succeq 0$ for the localizing matrix associated with the polynomial $x \mapsto \theta(x) = M^2 - \|x\|^2$. This results in a significant simplification.

4. Sum of squares approximation.

Let $\mathcal{A}$ be equipped with the norm

$$f \;\mapsto\; \|f\|_1 \;:=\; \sum_{\alpha \in \mathbb{N}^n} |f_\alpha|, \qquad f \in \mathcal{A}.$$

THEOREM 4.1. *Let $f \in \mathbb{R}[x_1, \ldots, x_n]$ be nonnegative with global minimum f^*, that is,*

$$0 \;\leq\; f^* \;\leq\; f(x), \qquad x \in \mathbb{R}^n.$$

(i) *There is some $r_0 \in \mathbb{N}, \lambda_0 \geq 0$ such that, for all $r \geq r_0$ and $\lambda \geq \lambda_0$,*

$$f + \lambda \sum_{k=0}^{r} \sum_{j=1}^{n} \frac{x_j^{2k}}{k!} \qquad \textit{is a sum of squares.} \tag{4.1}$$

(ii) *For every $\epsilon > 0$, there is some $r(f, \epsilon) \in \mathbb{N}$ such that,*

$$f_\epsilon := f + \epsilon \sum_{k=0}^{r(f,\epsilon)} \sum_{j=1}^{n} \frac{x_j^{2k}}{k!} \qquad \textit{is a sum of squares.} \tag{4.2}$$

Hence, $\|f - f_\epsilon\|_1 \to 0$ as $\epsilon \downarrow 0$.

For a detailed proof, the reader is referred to section 5.2.

Remark 4.2. Notice that whenever $r \geq r(f, \epsilon)$ (with $r(f, \epsilon)$ as in Theorem 4.1(ii)), the polynomial

$$f_{\epsilon r} \;:=\; f + \epsilon \sum_{k=0}^{r} \sum_{j=1}^{n} \frac{x_j^{2k}}{k!}$$

is also a sum of squares, as we add up squares to f_ϵ in (4.2).

In some specific examples, one may even obtain ad hoc perturbations f_ϵ of f, simpler than the one in (4.2), and with same properties. This is illustrated in the following nice example, kindly provided by Bruce Reznick.

Example 1. Consider the Motzkin polynomial $(x, y) \mapsto f(x, y) = 1 + x^2y^2(x^2 + y^2 - 3)$, which is nonnegative, but *not* a sum of squares. Then, for all $n \geq 3$, the polynomial

$$f_n \;:=\; f + 2^{4-2n}x^{2n}$$

is a sum of squares, and $\|f - f_n\|_1 \to 0$ as $n \to \infty$. To prove this, write

$$f(x, y) = (xy^2 + x^3/2 - 3x/2)^2 + p(x),$$

with $p(x) = 1 - (x^3/2 - 3x/2)^2 = (1 - x^2)^2(1 - x^2/4)$. Next, the univariate polynomial $x \mapsto q(x) := p(x) + 2^{4-2n}x^{2n}$ is nonnegative on $\mathbb{R}$, hence a sum of squares. Indeed, if $x^2 \leq 4$, then $p \geq 0$ and so $q \geq 0$. If $x^2 > 4$, then $|p(x)| \leq (x^2)^2x^2/4 = x^6/4$. From

$$q(x) \geq 2^{4-2n}x^{2n} - |p(x)| \;\geq\; \frac{x^6}{4}((x^2/4)^{n-3} - 1),$$

and the fact that $n \geq 3, x^2 > 4$, we deduce that $q(x) \geq 0$.

Theorem 4.1(ii) is a *denseness* result in the spirit of Theorem 9.1 in Berg, Christensen, and Ressel [2, p. 273] which states that the cone of s.o.s. polynomials is dense (also for the norm $\|f\|_1$) in the cone of polynomials that are nonnegative on $[-1,1]^n$. (However, notice that Theorem 4.1(ii) provides an *explicit* converging sequence $\{f_\epsilon\}$ with a simple and very specific form.) One may thus wonder whether the specific s.o.s. approximation f_ϵ in (4.2) is also valid for polynomials f that are only nonnegative on $[-1,1]^n$, and not necessarily on the whole $\mathbb{R}^n$. The answer is *no.* To see this, let $x_0 \in \mathbb{R}^n$ be such that $f(x_0) < 0$, and let $M > \|x_0\|_\infty$. Observe that with f_ϵ as in (4.2), one has $f_\epsilon(x) < f(x) + \epsilon \sum_{i=1}^n \mathrm{e}^{x_i^2}$, for all $x \in \mathbb{R}^n$, no matter the value of the parameter $r(f,\epsilon)$. Therefore, $f_\epsilon(x_0) < f(x_0) + \epsilon n \mathrm{e}^{M^2}$. Hence, for $\epsilon < |f(x_0)|\mathrm{e}^{-M^2}/n$, we have $f_\epsilon(x_0) < 0$. On the other hand, other ad hoc perturbations of the same flavor, may work. Consider the following example, again kindly provided by Bruce Reznick.

Example 2. Let f be the univariate polynomial $x \mapsto f(x) := 1 - x^2$, nonnegative on $[-1,1]$. The following ad hoc perturbation $x \mapsto f_n(x) := f(x) + c_n x^{2n}$ is s.o.s. whenever $c_n \geq (n-1)^{n-1}/n^n$, and so, one may choose c_n so as to also obtain $\|f_n - f\|_1 \to 0$, as $n \to \infty$. In this case, one has to be very careful in the choice of the coefficient c_n. It cannot be too small because the degree of f_n is fixed a priori $(2n)$, whereas for a nonnegative polynomial f, the parameter ϵ can be fixed arbitrarily, and independently of f. On the other hand, $r(f,\epsilon)$ is not known.

We next consider the case of a convex polynomial, nonnegative on a convex semialgebraic set. Given $\{g_j\}_{j=1}^m \subset \mathbb{R}[x_1,\ldots,x_n]$, let $\mathbb{K} \subset \mathbb{R}^n$ be the semialgebraic set

$$\mathbb{K} := \{x \in \mathbb{R}^n \mid \quad g_j(x) \geq 0, \quad j = 1,\ldots,m\}. \tag{4.3}$$

COROLLARY 4.3. *Let* $\mathbb{K}$ *be as in* (4.3)*, where all the* g_j*'s are concave, and assume that Slater's condition holds, i.e., there exists* $x_0 \in \mathbb{K}$ *such that* $g_j(x_0) > 0$ *for all* $j = 1,\ldots,m$.

Let $f \in \mathbb{R}[x_1,\ldots,x_n]$ *be convex, nonnegative on* $\mathbb{K}$*, and with a minimizer on* $\mathbb{K}$*, that is,* $f(x^*) \leq f(x)$ *for all* $x \in \mathbb{K}$*, for some* $x^* \in \mathbb{K}$*. Then there exists a nonnegative vector* $\lambda \in \mathbb{R}^m$ *such that for every* $\epsilon > 0$*, there is some* $r_\epsilon = r(f,\lambda,g_1,\ldots,g_m,\epsilon) \in \mathbb{N}$ *for which*

$$f + \epsilon \sum_{k=0}^{r_\epsilon} \sum_{i=1}^{n} \frac{x_i^{2k}}{k!} \;=\; f_0 + \sum_{j=1}^{m} \lambda_j\, g_j, \tag{4.4}$$

with $f_0 \in \mathbb{R}[x_1,\ldots,x_n]$ *being a sum of squares. (Therefore, the degree of* f_0 *is less than* $\max[2r_\epsilon, \deg f, \deg g_1, \ldots, \deg g_m]$*.)*

Proof. Consider the convex optimization problem $f^* := \min\{f(x) \mid x \in \mathbb{K}\}$. As f is convex, $\mathbb{K}$ is a convex set and Slater's condition holds, the Karush–Kuhn–Tucker (KKT) optimality condition holds. That is, there exists a *nonnegative* vector $\lambda \in \mathbb{R}^m$ of Lagrange–KKT multipliers, such that

$$\nabla f(x^*) \;=\; \sum_{j=1}^{m} \lambda_j \nabla g_j(x^*); \quad \lambda_j g_j(x^*) \;=\; 0,\; j = 1,\ldots,m.$$

(See, e.g., Rockafellar [14].) In other words, x^* is also a (global) minimizer of the convex Lagrangian $L := f - \sum_{j=1}^m \lambda_j g_j$. Then $f^* = f(x^*) = L(x^*)$ is the (global) minimum of f over $\mathbb{K}$, as well as the global minimum of L over $\mathbb{R}^n$, i.e.,

$$f - \sum_{j=1}^{m} \lambda_j g_j - f^* \geq 0, \qquad x \in \mathbb{R}^n. \tag{4.5}$$

As $f \geq 0$ on $\mathbb{K}$, $f^* \geq 0$, and so $L \geq 0$ on $\mathbb{R}^n$. Then (4.4) follows from Theorem 4.1(ii), applied to the polynomial L. □

When $\mathbb{K}$ is *compact* (and so, f has necessarily a minimizer $x^* \in \mathbb{K}$), one may compare Corollary 4.3 with Putinar's representation [13] of polynomials, positive on $\mathbb{K}$. When f is nonnegative on $\mathbb{K}$ (compact), and with

$$f_\epsilon \,:=\, f + \epsilon \sum_{k=0}^{r_\epsilon} \sum_{i=1}^{n} \frac{x_i^{2k}}{k!}, \tag{4.6}$$

one may rewrite (4.4) as

$$f_\epsilon \,=\, f_0 + \sum_{j=1}^{m} \lambda_j \, g_j, \tag{4.7}$$

which is indeed a *certificate* of positivitity of f_ϵ on $\mathbb{K}$. In fact, as $f_\epsilon > 0$ on $\mathbb{K}$, (4.7) can be seen as a special form of Putinar's s.o.s. representation, namely

$$f_\epsilon \,=\, q_0 + \sum_{j=1}^{m} q_j \, g_j, \qquad \text{with } q_0, \ldots, q_m \text{ s.o.s.} \tag{4.8}$$

(which holds under an additional assumption on the g_j's). So, in the convex compact case, and under Slater's condition, Corollary 4.3 states that if $f \geq 0$ on $\mathbb{K}$, then its approximation f_ϵ in (4.6), has the simplified Putinar representation (4.7), in which the s.o.s. coefficients $\{q_j\}$ of the g_j's in (4.8), now become simple nonnegative *scalars* in (4.7), namely, the Lagrange–KKT multipliers $\{\lambda_j\}$.

5. Proofs.

5.1. Proof of Theorem 3.3. We will prove (i) and (ii) together. We first prove that $\mathbb{Q}_r$ is solvable. This is because the feasible set (which is closed) is compact. Indeed, the constraint

$$\sum_{k=0}^{r} \sum_{i=1}^{n} y_{2k}^{(i)}/k! \,\leq\, n\mathrm{e}^{M^2}$$

implies that every diagonal element $y_{2k}^{(i)}$ of of $M_r(\mathbf{y})$ is bounded by $\tau_r := nr!\mathrm{e}^{M^2}$. By Lemma 6.2, this in turn implies that its diagonal elements (i.e., $y_{2\alpha}$, with $|\alpha| \leq r$) are *all* bounded by τ_r.

This latter fact, and again $M_r(\mathbf{y}) \succeq 0$, also imply that in fact *every* element of $M_r(\mathbf{y})$ is bounded by τ_r, that is, $|y_\alpha| \leq \tau_r$ for all $|\alpha| \leq 2r$. Indeed, for a symmetric matrix $A \succeq 0$, every nondiagonal element A_{ij} satisfies $A_{ij}^2 \leq A_{ii}A_{jj}$ so that $|A_{ij}| \leq \max_i A_{ii}$.

Therefore the set of feasible solutions of $\mathbb{Q}_r$ is a closed bounded subset of $\mathbb{R}^{s(2r)}$, hence compact. As $L_{\mathbf{y}}(f)$ is linear in $\mathbf{y}$, the infimum is attained at some feasible point. Thus, for all $r \geq r_f$, $\mathbb{Q}_r$ is solvable with optimal value $\min \mathbb{Q}_r \leq \inf \mathcal{P}_M$. The latter inequality is because the moment sequence $\mathbf{y}$ associated with an arbitrary feasible solution μ of $\mathcal{P}_M$, is obviously feasible for $\mathbb{Q}_r$, and with value $L_{\mathbf{y}}(f) = \int f d\mu$.

Next, as the sequence $\{\min \mathbb{Q}_r\}_r$ is obviously monotone nondecreasing, one has $\min \mathbb{Q}_r \uparrow \rho^* \leq \inf \mathcal{P}_M$, as $r \to \infty$. We have seen that every entry of $M_r(\mathbf{y})$ is bounded by τ_r, and this bound holds for all $r \geq r_f$. Moreover, $M_r(\mathbf{y})$ is also a (north-west corner) submatrix of $M_s(\mathbf{y})$ for every $s > r$. Indeed, whenever $s > r$, one may write

$$M_s(\mathbf{y}) \;=\; \left[\begin{array}{ccc} M_r(\mathbf{y}) & | & B \\ - & | & - \\ B' & | & C \end{array}\right]$$

for some appropriate matrices B and C. Therefore, for the same reasons, any feasible solution $\mathbf{y}$ of $\mathbb{Q}_s$ satisfies $|y_\alpha| \leq \tau_r$ for all $\alpha \in \mathbb{N}^n$ such that $|\alpha| \leq 2r$. Therefore, for every $s \in \mathbb{N}$, and every feasible solution $\mathbf{y}$ of $\mathbb{Q}_s$, we have

$$|y_\alpha| \leq \tau_r \quad \forall \alpha \in \mathbb{N}^n, \quad 2r-1 \leq |\alpha| \leq 2r, \quad r = 1, \ldots, s.$$

Thus, given $\mathbf{y} = \{y_\alpha\}$, denote by $\hat{\mathbf{y}} = \{\hat{\mathbf{y}}_\alpha\}$ the new sequence obtained from $\mathbf{y}$ by the scaling

$$\hat{\mathbf{y}}_\alpha \;:=\; \mathbf{y}_\alpha/\tau_r \qquad \forall \alpha \in \mathbb{N}^n, \quad 2r-1 \;\leq |\alpha| \leq 2r, \quad r = 1, 2, \ldots.$$

So let $\mathbf{y}^{(r)} \;=\; \{y_\alpha^{(r)}\}$ be an optimal solution of $\mathbb{Q}_r$ and complete $\mathbf{y}^{(r)}$ with zeros to make it an element of l_∞. Hence, all the elements $\hat{\mathbf{y}}^{(r)}$ are in the unit ball B_1 of l_∞, defined by

$$B_1 \;=\; \{\mathbf{y} = \{y_\alpha\} \in l_\infty \,|\quad \|\mathbf{y}\|_\infty \leq 1\}.$$

By the Banach–Alaoglu theorem, this ball is sequentially compact in the $\sigma(l_\infty, l_1)$ (weak*) topology of l_∞ (see, e.g., Ash [1]). In other words, there exists an element $\hat{\mathbf{y}}^* \in B_1$ and a subsequence $\{r_k\} \subset \mathbb{N}$, such that $\hat{\mathbf{y}}^{(r_k)} \to \hat{\mathbf{y}}^*$ for the weak* topology of l_∞, that is, for all $\mathrm{u} \in \mathrm{l}_1$,

$$\langle \hat{\mathbf{y}}^{(r_k)}, \mathrm{u}\rangle \;\to\; \langle \hat{\mathbf{y}}^*, \mathrm{u}\rangle, \qquad \text{as k} \to \infty. \tag{5.1}$$

In particular, *pointwise* convergence holds, that is, for all $\alpha \in \mathbb{N}^n$,

$$\hat{y}_\alpha^{(r_k)} \;\to\; \hat{y}_\alpha^*, \qquad \text{as } k \to \infty,$$

and so, defining $\mathbf{y}^*$ from $\hat{\mathbf{y}}^*$ by

$$\mathbf{y}_\alpha^* \;=\; \tau_r\, \hat{\mathbf{y}}_\alpha^* \qquad \forall \alpha \in \mathbb{N}^n, \quad 2r-1 \;\leq |\alpha| \leq 2r, \quad r = 1, 2, \ldots$$

one also obtains the pointwise convergence

$$\forall \alpha \in \mathbb{N}^n, \quad y_\alpha^{(r_k)} \;\to\; y_\alpha^*, \qquad \text{as } k \to \infty. \tag{5.2}$$

We next prove that $\mathbf{y}^*$ is the moment sequence of an optimal solution μ^* of problem $\mathcal{P}_M$. From the pointwise convergence (5.2), we immediately get $M_r(\mathbf{y}^*) \succeq 0$ for all $r \geq r_f$, because $M_r(\mathbf{y})$ belongs to the cone of positive semidefinite matrices of size $s(r)$, which is closed. Next, and again by pointwise convergence, for every $s \in \mathbb{N}$,

$$\sum_{j=0}^{s}\sum_{i=1}^{n} (y^*)_{2j}^{(i)}/j! \;=\; \lim_{k\to\infty} \sum_{j=0}^{s}\sum_{i=1}^{n} (y^{(r_k)})_{2j}^{(i)}/j! \;\leq\; n\mathrm{e}^{M^2},$$

and so, by the monotone convergence theorem

$$\sum_{j=0}^{\infty}\sum_{i=1}^{n} (y^*)_{2j}^{(i)}/j! \;=\; \lim_{s\to\infty} \sum_{j=0}^{s}\sum_{i=1}^{n} (y^*)_{2j}^{(i)}/j! \;\leq\; n\mathrm{e}^{M^2}. \tag{5.3}$$

But (5.3) implies that $\mathbf{y}^*$ satisfies Carleman's condition (2.4). Indeed, from (5.3), for all $i = 1, \ldots, n$, we have $(y^*)^{(i)}_{2k} < \rho k!$ for all $k \in \mathbb{N}$, and so, as $k! \leq k^k = \sqrt{k}^{2k}$,

$$[(y^*)^{(i)}_{2k}]^{-1/2k} > (\rho)^{-1/2k}/\sqrt{k},$$

which in turn implies

$$\sum_{k=0}^{\infty} [(y^*)^{(i)}_{2k}]^{-1/2k} > \sum_{k=0}^{\infty} \frac{\rho^{-1/2k}}{\sqrt{k}} \; = \; +\infty.$$

Hence, by Theorem 2.1, $\mathbf{y}^*$ is a determinate moment sequence, that is, there exists a unique measure μ^* on $\mathbb{R}^n$, such that

$$y^*_\alpha \; = \; \int_{\mathbb{R}^n} x^\alpha \, d\mu^*, \qquad \alpha \in \mathbb{N}^n.$$

By (5.3),

$$\int \sum_{i=1}^{n} \mathrm{e}^{x_i^2} \, d\mu^* \; = \; \sum_{j=0}^{\infty} \sum_{i=1}^{n} (y^*)^{(i)}_{2j}/j! \; \leq \; n\mathrm{e}^{M^2},$$

which proves that μ^* is admissible for $\mathcal{P}_M$.

But then, again by the pointwise convergence (5.2) of $\mathbf{y}^{(r_k)}$ to $\mathbf{y}^*$, we get $L_{\mathbf{y}^{(r_k)}}(f) \to L_{\mathbf{y}^*}(f) = \int f d\mu^*$ as $k \to \infty$, which, in view of $L_{\mathbf{y}^{(r_k)}}(f) \leq \inf \mathcal{P}_M$ for all k, implies

$$\int f \, d\mu^* \; = \; L_{\mathbf{y}^*}(f) \; \leq \; \inf \mathcal{P}_M.$$

But this proves that μ^* is an optimal solution of $\mathcal{P}_M$ because μ^* is admissible for $\mathcal{P}_M$ with value $\int f d\mu^* \leq \inf \mathcal{P}_M$. As the converging subsequence $\{r_k\}$ was arbitrary, it is true for every limit point. Hence, we have proved (i) and (ii).

(iii) Let $\mathbf{y}$ be the moment sequence associated with the probability measure μ on the ball $B_{M/2} \subset \mathbb{R}^n$

$$B_{M/2} \; = \; \{x \in \mathbb{R}^n \mid \quad \|x\|_\infty \leq M/2\},$$

with uniform distribution. That is,

$$\mu(B) \; = \; M^{-n} \int_{B \cap B_{M/2}} dx, \qquad B \in \mathcal{B},$$

where $\mathcal{B}$ is the sigma-algebra of Borel subsets of $\mathbb{R}^n$.

As μ has a continuous density $f_\mu > 0$ on $B_{M/2}$, it follows easily that $M_r(\mathbf{y}) \succ 0$ for all $r \geq r_f$. In addition,

$$\sum_{k=0}^{r} \sum_{i=1}^{n} y^{(i)}_{2k}/k! \; < \; \int \sum_{i=1}^{n} \mathrm{e}^{x_i^2} \, d\mu \; < \; n\mathrm{e}^{M^2},$$

so that $\mathbf{y}$ is a strictly admissible solution for $\mathbb{Q}_r$. Hence, the SDP problem $\mathbb{Q}_r$ satisfies Slater's condition, and so, there is no duality gap between $\mathbb{Q}_r$ and $\mathbb{Q}_r^*$, and $\mathbb{Q}_r^*$ is solvable if $\inf \mathbb{Q}_r$ is finite; see, e.g., Vandenberghe and Boyd [18]. Thus, $\mathbb{Q}_r^*$ is solvable because we proved that $\mathbb{Q}_r$ is solvable. In other words, $\sup \mathbb{Q}_r^* = \max \mathbb{Q}_r^* = \min \mathbb{Q}_r$, the desired result.

5.2. Proof of Theorem 4.1. It suffices to prove (i) and (ii) for the case $f^* > 0$. Indeed, if $f^* = 0$ take $\epsilon > 0$ arbitrary, fixed. Then $f + n\epsilon \geq f^*_\epsilon = f^* + n\epsilon > 0$ and so, suppose that (4.1) holds for $f + n\epsilon$ (for some r_0, λ_0). In particular, pick $\lambda \geq \lambda_0 + \epsilon$, so that

$$f + n\epsilon + (\lambda - \epsilon) \sum_{k=0}^{r_\lambda} \sum_{j=1}^{n} \frac{x_j^{2k}}{k!} \; = \; q_\lambda,$$

(with q_λ s.o.s.), for $r_\lambda \geq r_0$. Equivalently,

$$f + \lambda \sum_{k=0}^{r_\lambda} \sum_{j=1}^{n} \frac{x_j^{2k}}{k!} \; = \; q_\lambda + \epsilon \sum_{k=1}^{r_\lambda} \sum_{j=1}^{n} \frac{x_j^{2k}}{k!} \; = \; \hat{q}_\lambda,$$

where $\hat{q}_\lambda$ is a s.o.s. Hence (4.1) also holds for f (with $\lambda_0 + \epsilon$ in lieu of λ_0).

Similarly, for (4.2). As $f^* = 0$, $f + n\epsilon > 0$ and so, suppose that (4.2) holds for $f + n\epsilon$. In particular,

$$f + n\epsilon + \epsilon \sum_{k=0}^{r_\epsilon} \sum_{j=1}^{n} \frac{x_j^{2k}}{k!} \; = \; q_\epsilon,$$

(with q_ϵ s.o.s.), for some r_ϵ. Equivalently,

$$f + 2\epsilon \sum_{k=0}^{r_\epsilon} \sum_{j=1}^{n} \frac{x_j^{2k}}{k!} \; = \; q_\epsilon + \epsilon \sum_{k=1}^{r_\epsilon} \sum_{j=1}^{n} \frac{x_j^{2k}}{k!} \; = \; \hat{q}_\epsilon,$$

where $\hat{q}_\epsilon$ is a s.o.s. Hence (4.2) also holds for f. Therefore, we will assume that $f^* > 0$.

(i) As $f^* > 0$, let M_0 be such that $f^* > 1/M_0$, and fix $M > M_0$. Consider the SDP problem $\mathbb{Q}_r^*$ defined in (3.5), associated with M. By Proposition 3.2, $f^* \leq \inf \mathcal{P}_M$. By Theorem 3.3, $\max \mathbb{Q}_r^* = \min \mathbb{Q}_r \uparrow \inf \mathcal{P}_M \geq f^*$. Therefore, there exists some $r_M \geq r_f$ such that $\max \mathbb{Q}^*_{r_M} \geq f^* - 1/M > 0$. That is, if $(q_M, \lambda_M, \gamma_M)$ is an optimal solution of $\mathbb{Q}^*_{r_M}$, then $\gamma_M - n\lambda_M \mathrm{e}^{M^2} \geq f^* - 1/M > 0$. In addition,

$$f - \gamma_M = q_M - \lambda_M \sum_{k=0}^{r_M} \sum_{j=1}^{n} \frac{x_j^{2k}}{k!},$$

that we rewrite

$$f - (\gamma_M - n\lambda_M \mathrm{e}^{M^2}) \; = \; q_M + \lambda_M \left(n\mathrm{e}^{M^2} - \sum_{k=0}^{r_M} \sum_{j=1}^{n} \frac{x_j^{2k}}{k!} \right). \tag{5.4}$$

Equivalently,

$$f + \lambda_M \sum_{k=0}^{r_M} \sum_{j=1}^{n} \frac{x_j^{2k}}{k!} \; = \; q_M + n\lambda_M \mathrm{e}^{M^2} + (\gamma_M - n\lambda_M \mathrm{e}^{M^2}).$$

Define $\hat{q}_M$ to be the s.o.s. polynomial

$$\hat{q}_M \; := \; q_M + n\lambda_M \mathrm{e}^{M^2} + (\gamma_M - n\lambda_M \mathrm{e}^{M^2}),$$

so that we obtain

$$f+\lambda_M\sum_{k=0}^{r_M}\sum_{j=1}^{n}\frac{x_j^{2k}}{k!}\;=\;\hat{q}_M, \tag{5.5}$$

the desired result.

If we now take $r>r_M$ and $\lambda\geq\lambda_M$ we also have

$$\begin{aligned}f+\lambda\sum_{k=0}^{r}\sum_{j=1}^{n}\frac{x_j^{2k}}{k!} &= f+\lambda_M\sum_{k=0}^{r_M}\sum_{j=1}^{n}\frac{x_j^{2k}}{k!}\\ &\quad+\lambda_M\sum_{k=r_M+1}^{r}\sum_{j=1}^{n}\frac{x_j^{2k}}{k!}+(\lambda-\lambda_M)\sum_{k=0}^{r}\sum_{j=1}^{n}\frac{x_j^{2k}}{k!}\\ &=\hat{q}_M+\lambda_M\sum_{k=r_M+1}^{r}\sum_{j=1}^{n}\frac{x_j^{2k}}{k!}+(\lambda-\lambda_M)\sum_{k=0}^{r}\sum_{j=1}^{n}\frac{x_j^{2k}}{k!}\\ &=\hat{\hat{q}}_M,\end{aligned}$$

that is,

$$f+\lambda\sum_{k=0}^{r}\sum_{j=1}^{n}\frac{x_j^{2k}}{k!}\;=\;\hat{\hat{q}}_M, \tag{5.6}$$

where $\hat{\hat{q}}_M$ is a s.o.s. polynomial, the desired result.

(ii) Let M be as in (i) above. Evaluating (5.4) at $x=0$, and writing $f(0)=f(0)-f^*+f^*$, yields

$$f(0)-f^*+f^*-(\gamma_M-n\lambda_M\mathrm{e}^{M^2})\;=\;q_M(0)+n\lambda_M\,(\mathrm{e}^{M^2}-1),$$

and as $1/M\geq f^*-(\gamma_M-n\lambda_M\mathrm{e}^{M^2})$,

$$\lambda_M\;\leq\;\frac{1/M+f(0)-f^*}{n\,(\mathrm{e}^{M^2}-1)}.$$

Now, letting $M\to\infty$, yields $\lambda_M\to 0$.

Now, let $\epsilon>0$ be fixed, arbitrary. There is some $M>M_0$ such that $\lambda_M\leq\epsilon$ in (5.5). Therefore, (4.2) is just (5.6) with $\lambda:=\epsilon>\lambda_M$ and $r=r_\epsilon\geq r_M$. Finally, from this, we immediately have

$$\|f-f_\epsilon\|_1\;\leq\;\epsilon\sum_{i=1}^{n}\sum_{k=0}^{\infty}\frac{1}{k!}\;=\;\epsilon\,n\mathrm{e}\to 0,\quad\text{ as }\epsilon\downarrow 0.$$

6. Appendix. In this section we derive two auxiliary results that are helpful in the proofs of Theorems 3.3 and 4.1 in section 5.

LEMMA 6.1. *Let $n=2$ and let $\mathbf{y}$ be a sequence (with $y_0=1$) indexed in the basis* (2.1)*, and such that $M_r(\mathbf{y})\succeq 0$. Then all the diagonal entries of $M_r(\mathbf{y})$ are bounded by $\tau_r:=\max_{k=1,\ldots,r}\max\,[y_{2k,0},y_{0,2k}]$.*

Proof. It suffices to prove that all the entries $y_{2\alpha,2\beta}$ with $\alpha+\beta=r$ are bounded by $s_r:=\max\,[y_{2r,0},y_{0,2r}]$, and repeat the argument for entries $y_{2\alpha,2\beta}$ with $\alpha+\beta=$

$r-1, r-2$, etc. Then, take $\tau_r := \max_{k=1,\ldots,r} s_k$. So, consider the odd case $r = 2p+1$, and the even case $r = 2p$.

- The odd case $r = 2p+1$. Let $\Gamma := \{(2\alpha, 2\beta) \mid \quad \alpha+\beta = r,\, \alpha, \beta \neq 0\}$, and notice that

$$\Gamma \;=\; \{(2r-2k, 2k) \mid \quad k = 1, \ldots, r-1\} \;=\; \Gamma_1 \cup \Gamma_2$$

with

$$\Gamma_1 \;:=\; \{(r,0) + (r-2k, 2k), \mid \quad k = 1, \ldots, p\},$$

and

$$\Gamma_2 := \{(0,r) + (2j, r-2j), \mid \quad j = 1, \ldots, p\}.$$

Therefore, consider the two rows (and columns) corresponding to the indices $(r,0)$ and $(r-2k, 2k)$, or $(0,r)$ and $(2j, r-2j)$. In view of $M_r(\mathbf{y}) \succeq 0$, one has

$$\left\{ \begin{array}{lcll} y_{2r,0} \times y_{2r-4k,4k} & \geq & (y_{2r-2k,2k})^2, & \quad k = 1, \ldots, p, \\ y_{0,2r} \times y_{4j,2r-4j} & \geq & (y_{2j,2r-2j})^2, & \quad j = 1, \ldots, p. \end{array} \right. \tag{6.1}$$

Thus, let $s := \max\{y_{2\alpha,2\beta} \,|\alpha+\beta = r, \alpha\beta \neq 0\}$, so that either $s = y_{2r-2k^*,2k^*}$ for some $1 \leq k^* \leq p$, or $s = y_{2j^*,2r-2j^*}$ for some $1 \leq j^* \leq p$. But then, in view of (6.1), and with $s_r := \max\,[y_{2r,0}, y_{0,2r}]$,

$$s_r \times s \geq y_{2r,0} \times y_{2r-4k^*,4k^*} \geq (y_{2r-2k^*,2k^*})^2 = s^2,$$

or,

$$s_r \times s \geq y_{0,2r} \times y_{4j^*,2r-4j^*} \geq (y_{2j^*,2r-2j^*})^2 = s^2,$$

so that $s \leq s_r$, the desired result.

- The even case $r = 2p$. Again, the set $\Gamma := \{(2\alpha, 2\beta) \,|\alpha+\beta = r,\, \alpha, \beta \neq 0\}$ can be written $\Gamma = \Gamma_1 \cup \Gamma_2$, with

$$\Gamma_1 \;:=\; \{(r,0) + (r-2k, 2k), |k = 1, \ldots, p\},$$

and

$$\Gamma_2 := \{(0,r) + (2j, r-2j), \mid \quad j = 1, \ldots, p\}.$$

The only difference with the odd case is that $\Gamma_1 \cap \Gamma_2 = (2p, 2p) \neq \emptyset$. But the rest of the proof is the same as in the odd case. □

LEMMA 6.2. *Let $r \in \mathbb{N}$ be fixed, and let $\mathbf{y}$ be a sequence (with $y_0 = 1$) such that the associated moment matrix $M_r(\mathbf{y})$ is positive semidefinite, i.e., $M_r(\mathbf{y}) \succeq 0$. Assume that there is some $\tau_r \in \mathbb{R}$ such that the diagonal elements $\{y^{(i)}_{2k}\}$ satisfy $y^{(i)}_{2k} \leq \tau_r$ for all $k = 1, \ldots, r$, and all $i = 1, \ldots, n$.*

Then, the diagonal elements of $M_r(\mathbf{y})$ are all bounded by τ_r (i.e., $y_{2\alpha} \leq \tau_r$ for all $\alpha \in \mathbb{N}^n$, with $|\alpha| \leq r$).

Proof. The proof is by induction on the the number n of variables. By our assumption it is true for $n = 1$, and by Lemma 6.1 it is true for $n = 2$. Thus, suppose it is true for $k = 1, 2, \ldots, n-1$ variables and consider the case of n variables (with $n \geq 3$).

By our induction hypothesis, it is true for all elements $y_{2\alpha}$, where at least one index, say α_i, is zero ($\alpha_i = 0$). Indeed, the submatrix $A_r^{(i)}(\mathbf{y})$ of $M_r(\mathbf{y})$, obtained from $M_r(\mathbf{y})$ by deleting all rows and columns corresponding to indices $\alpha \in \mathbb{N}^n$ in the basis (2.1), with $\alpha_i > 0$, is a moment matrix of order r, with $n-1$ variables $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$. Hence, by a permutation of rows and columns, we can write

$$M_r(\mathbf{y}) = \left[\begin{array}{ccc} A_r^{(i)}(\mathbf{y}) & | & B \\ - & | & - \\ B' & | & C \end{array}\right],$$

for some appropriate matrices B and C. In particular, all elements $y_{2\alpha}$ with $\alpha_i = 0$, are diagonal elements of $A_r^{(i)}(\mathbf{y})$. In addition, its diagonal elements $y_{2k}^{(j)}$, $j \neq i$, are all bounded by τ_r. And of course, $A_r^{(i)}(\mathbf{y}) \succeq 0$. Therefore, by our induction hypothesis, all its diagonal elements are bounded by τ_r. As i was arbitrary, we conclude that all elements $y_{2\alpha}$ with at least one index being zero, are all bounded by τ_r.

We next prove it is true for an arbitrary element $y_{2\alpha}$ with $|\alpha| \leq r$ and $\alpha > 0$, i.e., $\alpha_j \geq 1$ for all $j = 1, \ldots, n$. With no loss of generality, we assume that $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_n$.

Consider the two elements $y_{2\alpha_1,0,\beta}$ and $y_{0,2\alpha_2,\gamma}$, with $\beta, \gamma \in \mathbb{N}^{n-2}$ such that

$$|\beta| = |\alpha| - 2\alpha_1; \quad |\gamma| = |\alpha| - 2\alpha_2,$$

and

$$(2\alpha_1, 0, \beta) + (0, 2\alpha_2, \gamma) = (2\alpha_1, 2\alpha_2, \beta + \gamma) = 2\alpha.$$

So, for instance, take $\beta = (\beta_3, \beta_4, \ldots, \beta_n)$, $\gamma = (\gamma_3, \gamma_4, \ldots, \gamma_n)$, defined by

$$\beta := (\alpha_3 + \alpha_2 - \alpha_1, \alpha_4, \ldots, \alpha_n), \quad \gamma := (\alpha_3 + \alpha_1 - \alpha_2, \alpha_4, \ldots, \alpha_n).$$

By construction, we have $4\alpha_1 + 2|\beta| = 4\alpha_2 + 2|\gamma| = 2|\alpha| \leq 2r$, so that both $y_{4\alpha_1,0,2\beta}$ and $y_{0,4\alpha_2,2\gamma}$ are diagonal elements of $M_r(\mathbf{y})$ with at least one entry equal to 0. Hence, by the induction hypothesis,

$$y_{4\alpha_1,0,2\beta} \leq \tau_r, \quad y_{0,4\alpha_2,2\gamma} \leq \tau_r.$$

Next, consider the two rows and columns indexed by $(2\alpha_1, 0, \beta)$ and $(0, 2\alpha_2, \gamma)$. The constraint $M_r(\mathbf{y}) \succeq 0$ clearly implies

$$\tau_r^2 \geq y_{4\alpha_1,0,2\beta} \times y_{0,4\alpha_2,2\gamma} \geq (y_{2\alpha_1,2\alpha_2,\beta+\gamma})^2 = y_{2\alpha}^2.$$

Hence, $y_{2\alpha} \leq \tau_r$, the desired result. □

Acknowledgment. The authors wishes to thank the anonymous referees for their valuable comments and suggestions, as well as Bruce Reznick who kindly provided the two examples in section 4.

REFERENCES

[1] R. Ash, *Real Analysis and Probability*, Academic Press, New York, 1972.

[2] C. Berg, J. P. R. Christensen, and P. Ressel, *Positive definite functions on Abelian semigroups*, Math. Ann., 223 (1976), pp. 253–274.

[3] C. Berg, *The multidimensional problem and semigroups*, in Moments in Mathematics, AMS short course, San Antonio, 1987, Proc. Symp. Appl. Math., 37 (1987), pp. 110–124.

[4] G. Blekherman, *There are significantly more nonnegative polynomials than sums of squares*, Department of Mathematics, University of Michigan, Ann Arbor, MI, 2004.

[5] T. Jacobi and A. Prestel, *Distinguished representations of strictly positive polynomials*, J. Reine. Angew. Math., 532 (2001), pp. 223–235.

[6] J. B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM J. Optim., 11 (2000/01), pp. 796–817.

[7] J. B. Lasserre, *Polynomials nonnegative on a grid and discrete optimization*, Trans. Amer. Math. Soc., 354 (2002), pp. 631–649.

[8] J. B. Lasserre, *Semidefinite programming vs. LP relaxations for polynomial programming*, Math. Oper. Res., 27 (2002), pp. 347–360.

[9] A. E. Nussbaum, *Quasi-analytic vectors*, Ark. Mat., 6 (1965), pp. 179–191.

[10] P. A. Parrilo, *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*, Ph.D. thesis, California Institute of Technology, Pasadena, CA, 2000.

[11] P. A. Parrilo, *Semidefinite programming relaxations for semialgebraic problems*, Math. Program., 96 (2003), pp. 293–320.

[12] A. Prestel and C. N. Delzell, *Positive Polynomials*, Springer-Verlag, Berlin, 2001.

[13] M. Putinar, *Positive polynomials on compact semi-algebraic sets*, Indiana Univ. Math. J., 42 (1993), pp. 969–984.

[14] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[15] C. Scheiderer, *Positivity and Sums of Squares: A Guide to Some Recent Results*, Department of Mathematics, University of Duisburg, Duisburg, Germany, 2003.

[16] K. Schmüdgen, *The K-moment problem for compact semi-algebraic sets*, Math. Ann., 289 (1991), pp. 203–206.

[17] M. Schweighofer, *Optimization of polynomials on compact semialgebraic sets*, SIAM J. Optim, 15 (2005), pp. 805–825.

[18] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95.

SIAM J. OPTIM.
Vol. 16, No. 3, pp. 766–797

ENHANCED FRITZ JOHN CONDITIONS FOR CONVEX PROGRAMMING[*]

DIMITRI P. BERTSEKAS[†], ASUMAN E. OZDAGLAR[†], AND PAUL TSENG[‡]

Abstract. We consider convex constrained optimization problems, and we enhance the classical Fritz John optimality conditions to assert the existence of multipliers with special sensitivity properties. In particular, we prove the existence of Fritz John multipliers that are informative in the sense that they identify constraints whose relaxation, at rates proportional to the multipliers, strictly improves the primal optimal value. Moreover, we show that if the set of geometric multipliers is nonempty, then the minimum-norm vector of this set is informative and defines the optimal rate of cost improvement per unit constraint violation. Our assumptions are very general and, in particular, allow for the presence of a duality gap and the nonexistence of optimal solutions. In particular, for the case where there is a duality gap, we establish enhanced Fritz John conditions involving the dual optimal value and dual optimal solutions.

Key words. convex program, Fritz John multiplier, geometric multiplier, duality, sensitivity analysis

AMS subject classifications. 90C25, 90C31, 90C46

DOI. 10.1137/040613068

1. Introduction.

We consider the convex constrained optimization problem

$$\text{(P)} \qquad \begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in X, \quad g(x) = (g_1(x), \dots, g_r(x))' \leq 0, \end{array}$$

where X is a nonempty convex subset of $\Re^n$, and $f : X \to \Re$ and $g_j : X \to \Re$ are convex functions. Here and throughout the paper, we denote by $\Re$ the real line, by $\Re^n$ the space of n-dimensional real column vectors with the standard Euclidean norm, $\|\cdot\|$, and by $'$ the transpose of a vector. We say that a function $f : X \to \Re$ is convex (respectively, closed) if its epigraph $\text{epi}(f) = \big\{(x, w) \mid x \in X,\, f(x) \leq w\big\}$ is convex (respectively, closed). For some of our results, we will assume that $f, g_1, \dots, g_r$ are also closed. We note that our analysis readily extends to the case where there are affine equality constraints by replacing each affine equality constraint with two affine inequality constraints.

We refer to problem (P) as the *primal problem* and we consider the *dual problem*

$$\text{(D)} \qquad \begin{array}{ll} \text{maximize} & q(\mu) \\ \text{subject to} & \mu \geq 0, \end{array}$$

where q is the *dual function*:

$$q(\mu) = \inf_{x \in X} \big\{ f(x) + \mu' g(x) \big\}, \qquad \mu \in \Re^r.$$

[*] Received by the editors August 8, 2004; accepted for publication (in revised form) May 3, 2005; published electronically January 12, 2006. This research was supported by National Science Foundation grant ECS-0218328.

http://www.siam.org/journals/siopt/16-3/61306.html

[†] Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 (dimitrib@mit.edu).

[‡] Department of Mathematics, University of Washington, Seattle, WA 98195 (tseng@math.washington.edu).

We denote by f^* and q^* the optimal values of (P) and (D), respectively:

$$f^* = \inf_{x \in X,\, g(x) \le 0} f(x), \qquad q^* = \sup_{\mu \ge 0} q(\mu).$$

We write $f^* < \infty$ or $q^* > -\infty$ to indicate that (P) or (D), respectively, has at least one feasible solution. The weak duality theorem states that $q^* \le f^*$. If $q^* = f^*$, we say that there is *no duality gap*.

An important result (see, e.g., [BNO03, Proposition 6.6.1]) is that there always exist a scalar μ_0^* and a vector $\mu^* = (\mu_1^*, \dots, \mu_r^*)'$ satisfying the following conditions:

(i) $\mu_0^* f^* = \inf_{x \in X} \big\{ \mu_0^* f(x) + \mu^{*\prime} g(x) \big\}$.

(ii) $\mu_j^* \ge 0$ for all $j = 0, 1, \dots, r$.

(iii) $\mu_0^*, \mu_1^*, \dots, \mu_r^*$ are not all equal to 0.

This type of conditions traces its origin to Fritz John's work [Joh48], although John's original conditions associate the multiplier pair (μ_0^*, μ^*) with a specific optimal solution of problem (P), and condition (i) instead involves the first derivatives at this optimal solution being equal to zero (assuming $X = \Re^n$). Since John's work has been primarily responsible for popularizing the idea of using the extra multiplier μ_0^* without any constraint qualification, we call a pair (μ_0^*, μ^*) satisfying (i)–(iii) an *FJ-multiplier.*[1]

If the coefficient μ_0^* of an FJ-multiplier is nonzero, by normalization one can obtain an FJ-multiplier of the form $(1, \mu^*)$, and we have

$$\mu^* \ge 0, \qquad f^* = q(\mu^*).$$

A vector μ^* thus obtained is called a *geometric multiplier*. It is well known and readily seen from the weak duality theorem that μ^* is a geometric multiplier if and only if there is no duality gap and μ^* is an optimal solution of the dual problem. It is further known that the set of geometric multipliers is closed and coincides with the negative of the subdifferential of the *perturbation function*

$$p(u) = \inf_{x \in X,\, g(x) \le u} f(x)$$

at $u = 0$, provided that p is convex and proper and $p(0)$ is finite [Roc70, Theorem 29.1], [BNO03, Proposition 6.5.8]. If in addition the origin is in the relative interior of $\mathrm{dom}(p)$ (a constraint qualification that guarantees that the set of geometric multipliers is nonempty) and μ^* is the geometric multiplier of minimum norm, then either $\mu^* = 0$, in which case $0 \in \partial p(0)$ and $u = 0$ is a global minimum of p, or $\mu^* \ne 0$, in which case μ^* is a direction of steepest descent for p at $u = 0$; i.e., the directional derivative $p'(0; d)$ of p at 0 in the direction d satisfies

$$(1) \qquad \inf_{\|d\|=1} p'(0; d) = p'\big(0; \mu^*/\|\mu^*\|\big) = \frac{1}{\|\mu^*\|} \sup_{v \in \partial p(0)} \mu^{*\prime} v = -\|\mu^*\| < 0.$$

Thus, the minimum-norm geometric multiplier provides useful sensitivity information; namely, relaxing the inequality constraints at rates equal to the components of $\mu^*/\|\mu^*\|$ yields a decrease of the optimal value at the optimal rate, which is equal to $\|\mu^*\|$.

[1] Fritz John's conditions were independently derived earlier by Karush [Kar39], as noted in Kuhn's historical note [Kuh91, section 6].

On the other hand, if the origin is not a relative interior point of $\mathrm{dom}(p)$, there may be no direction of steepest descent, because the directional derivative function $p'(0;\cdot)$ is discontinuous, and the infimum over $\|d\| = 1$ in (1) may not be attained. This can happen even if there is no duality gap and there exists a geometric multiplier. As an example, consider the following two-dimensional problem:

$$\begin{array}{ll} \text{minimize} & -x_2 \\ \text{subject to} & x \in X = \{x \mid x_2^2 \le x_1\}, \quad g_1(x) = x_1 \le 0, \quad g_2(x) = x_2 \le 0. \end{array}$$

It can be verified that

$$\mathrm{dom}(p) = \{u \mid u_2^2 \le u_1\} + \{u \mid u \ge 0\},$$

and

$$p(u) = \begin{cases} -u_2 & \text{if } u_2^2 \le u_1, \\ -\sqrt{u_1} & \text{if } u_1 \le u_2^2,\ u_1 \ge 0,\ u_2 \ge 0, \\ \infty & \text{otherwise,} \end{cases}$$

while

$$q(\mu) = \begin{cases} -\frac{(\mu_2-1)^2}{4\mu_1} & \text{if } \mu_1 > 0, \\ 0 & \text{if } \mu_1 = 0,\ \mu_2 = 1, \\ -\infty & \text{otherwise.} \end{cases}$$

We have $f^* = q^* = 0$, and the set of geometric multipliers is

$$\{\mu \ge 0 \mid \mu_2 = 1\}.$$

However, the geometric multiplier of minimum norm, $\mu^* = (0,1)$, is not a direction of steepest descent, since starting at $u = 0$ and going along the direction $(0,1)$, $p(u)$ is equal to 0, so

$$p'(0;\mu^*) = 0.$$

In fact p has no direction of steepest descent at $u = 0$, because $p'(0;\cdot)$ is not continuous. To see this, note that directions of descent $d = (d_1, d_2)$ are those for which $d_1 > 0$ and $d_2 > 0$, and that along any such direction, we have

$$p'(0;d) = -d_2.$$

It follows that

$$\inf_{\|d\|=1} p'(0;d) = -1 = -\|\mu^*\|,$$

but there is no direction of descent that attains the infimum above. On the other hand, there are sequences $\{u^k\} \subset \mathrm{dom}(p)$ and $\{x^k\} \subset X$ of infeasible points (in fact, the sequences $u^k = x^k = (1/k^2,\ 1/k)$) such that

$$\lim_{k\to\infty} \frac{p(0) - p(u^k)}{\|u^{k+}\|} = \lim_{k\to\infty} \frac{f^* - f(x^k)}{\|g^+(x^k)\|} = \|\mu^*\| = 1,$$

where we denote

$$u_j^+ = \max\{0, u_j\}, \quad u^+ = (u_1^+, \dots, u_r^+)', g_j^+(x) = \max\{0, g_j(x)\},$$
$$g^+(x) = (g_1^+(x), \dots, g_r^+(x))'.$$

Thus, the minimum norm of geometric multipliers can still be interpreted as the optimal rate of improvement of the cost per unit constraint violation. However, this rate of improvement cannot be obtained by approaching 0 along a straight line but only by approaching it along a curve.

In this paper, we derive more powerful versions of the Fritz John conditions, which provide sensitivity information, like the one discussed above. In particular, in addition to conditions (i)–(iii) above, we obtain an additional necessary condition (e.g., condition (CV) of Proposition 2 in the next section) that narrows down the set of candidates for optimality. Furthermore, our conditions also apply in the exceptional case where the set of geometric multipliers is empty. In this case, we will show that a certain degenerate FJ-multiplier, i.e., one of the form $(0, \mu^*)$ with $\mu^* \neq 0$ and $0 = \inf_{x \in X} \mu^{*\prime} g(x)$, provides sensitivity information analogous to that provided by the minimum-norm geometric multiplier. In particular, there exists an FJ-multiplier $(0, \mu^*)$ such that, by relaxing the inequality constraints at rates proportional to the components of $\mu^*/\|\mu^*\|$, we can strictly improve the primal optimal value. Furthermore, $\|\mu^*\|$ is the optimal rate of improvement per unit constraint violation. In the case where there is a duality gap, we also prove dual versions of these results, involving the dual optimal value, and dual FJ-multipliers. To our knowledge, except for a preliminary version of our work that appeared in the book [BNO03], these are the first results that provide enhanced, sensitivity-related Fritz John conditions for convex programming and also derive the optimal sensitivity rate under very general assumptions, i.e., without any constraint qualification and even in the presence of a duality gap.

This paper is organized as follows. In section 2, we present enhanced Fritz John conditions for convex problems that have optimal solutions. In section 3, we present analogous results for convex problems that have dual optimal solutions. In particular, we show that the dual optimal solution of minimum norm provides useful sensitivity information, even in the presence of a duality gap. We also introduce the notion of pseudonormality, and we discuss its connections to classical constraint qualifications. In section 4, we present Fritz John conditions for problems that may not have optimal solutions. In section 5, we prove dual versions of these conditions involving the dual optimal value.

2. Enhanced Fritz John conditions. The existence of FJ-multipliers is often used as the starting point for the analysis of the existence of geometric multipliers. Unfortunately, these conditions in their classical form are not sufficient to deduce the existence of geometric multipliers under some of the standard constraint qualifications, such as when $X = \Re^n$ and the constraint functions g_j are affine. Recently, the classical Fritz John conditions have been enhanced through the addition of an extra necessary condition, and their effectiveness has been significantly improved (see Hestenes [Hes75, Theorem 10.5 on page 242] for the case $X = \Re^n$, Bertsekas [Ber99, Proposition 3.3.11] for the case where X is a closed convex set, and Bertsekas and Ozdaglar [BeO02] for the case where X is a closed set). All of these results assume that an optimal solution exists and that the cost and the constraint functions are smooth (but possibly nonconvex). In this section, we retain the assumption of existence of an optimal solution, and instead of smoothness we assume the following.

ASSUMPTION 1 (closedness). *The functions f and $g_1, \dots, g_r$ are closed.*

We note that f and $g_1, \dots, g_r$ are closed if and only if they are lower semicontinuous on X, i.e., for each $\bar{x} \in X$, we have

$$f(\bar{x}) \leq \liminf_{x \in X,\ x \to \bar{x}} f(x), \qquad g_j(\bar{x}) \leq \liminf_{x \in X,\ x \to \bar{x}} g_j(x), \quad j = 1, \dots, r$$

(see, e.g., [BNO03, Proposition 1.2.2]). Under the preceding assumption, we prove the following version of the enhanced Fritz John conditions. Because we assume that f and $g_1, \ldots, g_r$ are convex over X rather than over $\Re^n$, the lines of proof from the preceding references (based on the use of gradients or subgradients) break down. We use a different line of proof, which is based instead on minimax arguments. The proof also uses the following lemma.

LEMMA 1. *Consider the convex problem* (P) *and assume that* $-\infty < q^*$. *If* μ^* *is a dual optimal solution, then*

$$\frac{q^* - f(x)}{\|g^+(x)\|} \leq \|\mu^*\| \qquad \forall\ x \in X \text{ that are infeasible.}$$

Proof. For any $x \in X$ that is infeasible, we have from the definition of the dual function that

$$q^* = q(\mu^*) \leq f(x) + {\mu^*}' g(x) \leq f(x) + {\mu^*}' g^+(x) \leq f(x) + \|\mu^*\| \|g^+(x)\|. \qquad \square$$

Note that the preceding lemma shows that the minimum distance to the set of dual optimal solutions is an upper bound for the cost improvement/constraint violation ratio $\big(q^* - f(x)\big)/\|g^+(x)\|$. The next proposition shows that, under certain assumptions including the absence of a duality gap, this upper bound is sharp and is asymptotically attained by an appropriate sequence $\{x^k\} \subset X$. The same fact will also be shown in section 3, but under considerably more general assumptions (see Proposition 7).

PROPOSITION 2. *Consider the convex problem* (P) *under Assumption* 1 *(closedness), and assume that* x^* *is an optimal solution. Then there exists an FJ-multiplier* (μ_0^*, μ^*) *satisfying the following condition* (CV). *Moreover, if* $\mu_0^* \neq 0$, *then* μ^*/μ_0^* *must be the geometric multiplier of minimum norm.*

(CV) *If* $\mu^* \neq 0$, *then there exists a sequence* $\{x^k\} \subset X$ *of infeasible points that converges to* x^* *and satisfies*

$$f(x^k) \to f^*, \qquad g^+(x^k) \to 0, \tag{2}$$

$$\frac{f^* - f(x^k)}{\|g^+(x^k)\|} \to \begin{cases} \|\mu^*\|/\mu_0^* & \textit{if } \mu_0^* \neq 0, \\ \infty & \textit{if } \mu_0^* = 0, \end{cases} \tag{3}$$

$$\frac{g^+(x^k)}{\|g^+(x^k)\|} \to \frac{\mu^*}{\|\mu^*\|}. \tag{4}$$

Proof. For positive integers k and m, we consider the saddle function

$$L_{k,m}(x, \xi) = f(x) + \frac{1}{k^3}\|x - x^*\|^2 + \xi' g(x) - \frac{1}{2m}\|\xi\|^2.$$

We note that, for fixed $\xi \geq 0$, $L_{k,m}(x, \xi)$, viewed as a function from X to $\Re$, is closed and convex because of the closedness assumption. Furthermore, for a fixed x, $L_{k,m}(x, \xi)$ is negative definite quadratic in ξ. For each k, we consider the set

$$X^k = X \cap \big\{x \mid \|x - x^*\| \leq k\big\}.$$

Since f and g_j are closed and convex when restricted to X, they are closed, convex, and coercive when restricted to X^k. Thus, we can use the saddle point theorem (e.g., [BNO03, Proposition 2.6.9]) to assert that $L_{k,m}$ has a saddle point over $x \in X^k$ and $\xi \geq 0$. This saddle point is denoted by $(x^{k,m}, \xi^{k,m})$.

The infimum of $L_{k,m}(x, \xi^{k,m})$ over $x \in X^k$ is attained at $x^{k,m}$, implying that

$$\begin{aligned} & f(x^{k,m}) + \frac{1}{k^3}\|x^{k,m} - x^*\|^2 + \xi^{k,m\prime} g(x^{k,m}) \\ & \quad = \inf_{x \in X^k} \left\{ f(x) + \frac{1}{k^3}\|x - x^*\|^2 + \xi^{k,m\prime} g(x) \right\} \\ & \quad \leq \inf_{x \in X^k,\, g(x) \leq 0} \left\{ f(x) + \frac{1}{k^3}\|x - x^*\|^2 + \xi^{k,m\prime} g(x) \right\} \\ & \quad \leq \inf_{x \in X^k,\, g(x) \leq 0} \left\{ f(x) + \frac{1}{k^3}\|x - x^*\|^2 \right\} \\ & \quad = f(x^*). \end{aligned} \tag{5}$$

Hence, we have

$$\begin{aligned} L_{k,m}(x^{k,m}, \xi^{k,m}) &= f(x^{k,m}) + \frac{1}{k^3}\|x^{k,m} - x^*\|^2 + \xi^{k,m\prime} g(x^{k,m}) - \frac{1}{2m}\|\xi^{k,m}\|^2 \\ &\leq f(x^{k,m}) + \frac{1}{k^3}\|x^{k,m} - x^*\|^2 + \xi^{k,m\prime} g(x^{k,m}) \\ &\leq f(x^*). \end{aligned} \tag{6}$$

Since $L_{k,m}(x^{k,m}, \xi)$ is quadratic in ξ, the supremum of $L_{k,m}(x^{k,m}, \xi)$ over $\xi \geq 0$ is attained at

$$\xi^{k,m} = m g^+(x^{k,m}). \tag{7}$$

This implies that

$$\begin{aligned} L_{k,m}(x^{k,m}, \xi^{k,m}) &= f(x^{k,m}) + \frac{1}{k^3}\|x^{k,m} - x^*\|^2 + \frac{m}{2}\|g^+(x^{k,m})\|^2 \\ &\geq f(x^{k,m}) + \frac{1}{k^3}\|x^{k,m} - x^*\|^2 \\ &\geq f(x^{k,m}). \end{aligned} \tag{8}$$

From (6) and (8), we see that the sequence $\{x^{k,m}\}$, with k fixed, belongs to the set $\{x \in X^k \mid f(x) \leq f(x^*)\}$, which is compact. Hence, $\{x^{k,m}\}$ has a cluster point (as $m \to \infty$), denoted by $\overline{x}^k$, which belongs to $\{x \in X^k \mid f(x) \leq f(x^*)\}$. By passing to a subsequence if necessary, we can assume without loss of generality that $\{x^{k,m}\}$ converges to $\overline{x}^k$ as $m \to \infty$. For each k, the sequence $\{f(x^{k,m})\}$ is bounded from below by $\inf_{x \in X^k} f(x)$, which is finite by Weierstrass's theorem since f is closed and coercive when restricted to X^k. Also, for each k, $L_{k,m}(x^{k,m}, \xi^{k,m})$ is bounded from above by $f(x^*)$ (cf. (6)), so the equality in (8) implies that

$$\limsup_{m \to \infty} g_j(x^{k,m}) \leq 0 \qquad \forall\ j = 1, \ldots, r.$$

Therefore, by using the lower semicontinuity of g_j, we obtain $g(\overline{x}^k) \leq 0$, implying that $\overline{x}^k$ is a feasible solution of problem (P), so that $f(\overline{x}^k) \geq f(x^*)$. Using (6) and (8) together with the lower semicontinuity of f, we also have

$$f(\overline{x}^k) \leq \liminf_{m \to \infty} f(x^{k,m}) \leq \limsup_{m \to \infty} f(x^{k,m}) \leq f(x^*),$$

thereby showing that for each k,

$$\lim_{m\to\infty} f(x^{k,m}) = f(x^*).$$

Together with (6) and (8), this also implies that for each k,

$$\lim_{m\to\infty} x^{k,m} = x^*.$$

Combining the preceding relations with (6) and (8), for each k, we obtain

$$\lim_{m\to\infty} \left(f(x^{k,m}) - f(x^*) + \xi^{k,m'} g(x^{k,m})\right) = 0. \tag{9}$$

Denote

$$\delta^{k,m} = \sqrt{1 + \|\xi^{k,m}\|^2}, \qquad \mu_0^{k,m} = \frac{1}{\delta^{k,m}}, \qquad \mu^{k,m} = \frac{\xi^{k,m}}{\delta^{k,m}}. \tag{10}$$

Since $\delta^{k,m}$ is bounded from below by 1, by dividing (9) by $\delta^{k,m}$, we obtain

$$\lim_{m\to\infty} \left(\mu_0^{k,m} f(x^{k,m}) - \mu_0^{k,m} f(x^*) + \mu^{k,m'} g(x^{k,m})\right) = 0.$$

By the preceding relations, for each k we can find a sufficiently large integer m_k such that

$$\left|\mu_0^{k,m_k} f(x^{k,m_k}) - \mu_0^{k,m_k} f(x^*) + \mu^{k,m_k{}'} g(x^{k,m_k})\right| \le \frac{1}{k}, \tag{11}$$

and

$$\|x^{k,m_k} - x^*\| \le \frac{1}{k}, \quad |f(x^{k,m_k}) - f(x^*)| \le \frac{1}{k}, \quad \|g^+(x^{k,m_k})\| \le \frac{1}{k}. \tag{12}$$

Dividing both sides of the first relation in (5) by δ^{k,m_k}, we obtain

$$\begin{aligned} &\mu_0^{k,m_k} f(x^{k,m_k}) + \frac{1}{k^3 \delta^{k,m_k}} \|x^{k,m_k} - x^*\|^2 + \mu^{k,m_k{}'} g(x^{k,m_k}) \\ &\qquad \le \mu_0^{k,m_k} f(x) + \mu^{k,m_k{}'} g(x) + \frac{1}{k\delta^{k,m_k}} \qquad \forall\, x \in X^k, \end{aligned}$$

where we also use the fact that $\|x - x^*\| \le k$ for all $x \in X^k$ (see the definition of X^k). Since the sequence $\{(\mu_0^{k,m_k}, \mu^{k,m_k})\}$ is bounded, it has a cluster point, denoted by (μ_0^*, μ^*), which satisfies conditions (ii) and (iii) in the definition of an FJ-multiplier. For any $x \in X$, we have $x \in X^k$ for all k sufficiently large. Without loss of generality, we will assume that the entire sequence $\{(\mu_0^{k,m_k}, \mu^{k,m_k})\}$ converges to (μ_0^*, μ^*). Taking the limit as $k \to \infty$, and using (11), we obtain

$$\mu_0^* f(x^*) \le \mu_0^* f(x) + \mu^{*\prime} g(x) \qquad \forall\, x \in X.$$

Since $\mu^* \ge 0$, this implies that

$$\begin{aligned} \mu_0^* f(x^*) &\le \inf_{x\in X} \left\{\mu_0^* f(x) + \mu^{*\prime} g(x)\right\} \\ &\le \inf_{x\in X,\, g(x)\le 0} \left\{\mu_0^* f(x) + \mu^{*\prime} g(x)\right\} \\ &\le \inf_{x\in X,\, g(x)\le 0} \mu_0^* f(x) \\ &= \mu_0^* f(x^*). \end{aligned}$$

Thus we have

$$\mu_0^* f(x^*) = \inf_{x \in X} \{ \mu_0^* f(x) + \mu^{*\prime} g(x) \},$$

so that (μ_0^*, μ^*) also satisfies condition (i) in the definition of an FJ-multiplier.

If $\mu^* = 0$, then $\mu_0^* \neq 0$, (CV) is automatically satisfied, and $\mu^*/\mu_0^* = 0$ has minimum norm. Moreover, condition (i) yields $f^* = \inf_{x \in X} f(x)$, so that (CV) (in particular, (3)) is satisfied by only $\mu^* = 0$.

Assume now that $\mu^* \neq 0$, so that the index set $J = \{ j \neq 0 \mid \mu_j^* > 0 \}$ is nonempty. Then, for sufficiently large k, we have $\xi_j^{k,m_k} > 0$ and hence $g_j(x^{k,m_k}) > 0$ for all $j \in J$. Thus, for each k, we can choose the index m_k to further satisfy $x^{k,m_k} \neq x^*$, in addition to (11) and (12). Using (7), (10), and the fact that $\mu^{k,m_k} \to \mu^*$, we obtain

$$\frac{g^+(x^{k,m_k})}{\|g^+(x^{k,m_k})\|} = \frac{\mu^{k,m_k}}{\|\mu^{k,m_k}\|} \to \frac{\mu^*}{\|\mu^*\|}.$$

Using also (6) and $f(x^*) = f^*$, we have that

$$\frac{f^* - f(x^{k,m_k})}{\|g^+(x^{k,m_k})\|} \geq \frac{\xi^{k,m_k\prime} g(x^{k,m_k})}{\|g^+(x^{k,m_k})\|} = \|\xi^{k,m_k}\| = \frac{\|\mu^{k,m_k}\|}{\mu_0^{k,m_k}}. \tag{13}$$

If $\mu_0^* = 0$, then $\mu_0^{k,m_k} \to 0$, so (13) together with $\|\mu^{k,m_k}\| \to \|\mu^*\| > 0$ yields

$$\frac{f^* - f(x^{k,m_k})}{\|g^+(x^{k,m_k})\|} \to \infty.$$

If $\mu_0^* \neq 0$, then (13) together with $\mu_0^{k,m_k} \to \mu_0^*$ and $\|\mu^{k,m_k}\| \to \|\mu^*\|$ yields

$$\liminf_{k \to \infty} \frac{f^* - f(x^{k,m_k})}{\|g^+(x^{k,m_k})\|} \geq \frac{\|\mu^*\|}{\mu_0^*}.$$

Since μ^*/μ_0^* is a geometric multiplier and $f^* = q^*$, Lemma 1 implies that in fact μ^*/μ_0^* is of minimum norm and the inequality holds with equality. From (12), we have $f(x^{k,m_k}) \to f(x^*)$, $g^+(x^{k,m_k}) \to 0$, and $x^{k,m_k} \to x^*$. Hence, the sequence $\{x^{k,m_k}\}$ also satisfies conditions (4)–(5) of the proposition, concluding the proof. □

Note that (4) implies that, for all k sufficiently large,

$$g_j(x^k) > 0 \quad \forall\, j \in J \qquad g_j^+(x^k) = o\left(\min_{j \in J} g_j^+(x^k) \right) \quad \forall\, j \notin J,$$

where $J = \{ j \neq 0 \mid \mu_j^* > 0 \}$. Thus, the (CV) condition (complementarity violation) in Proposition 2 refines that used in [BNO03, section 5.7] by also estimating the rate of cost improvement. As an illustration of Proposition 2, consider the two-dimensional example of Duffin:

$$\begin{array}{ll} \text{minimize} & x_2 \\ \text{subject to} & x = (x_1, x_2)' \in \Re^2, \quad \|x\| - x_1 \leq 0. \end{array}$$

Here $f^* = 0$, and $x^* = (x_1^*, 0)$ is an optimal solution for any $x_1^* \geq 0$. Also, $q(\mu) = -\infty$ for all $\mu \geq 0$, so $q^* = -\infty$ and there is a duality gap. It can be seen that $\mu_0^* = 0$, $\mu^* = 1$ form an FJ-multiplier and, together with $x^k = (x_1^*, -1/k)'$, satisfy condition (CV).

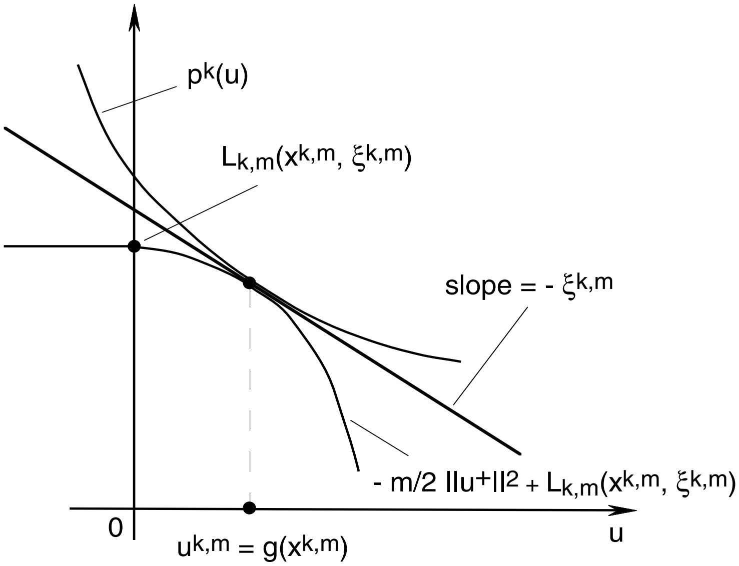


FIG. 1. *Illustration of the saddle point of the function $L_{k,m}(x, \xi)$ over $x \in X^k$ and $\xi \geq 0$ in terms of the function $p^k(u)$, which is the optimal value of problem* (14) *as a function of u.*

The proof of Proposition 2 can be explained in terms of the construction shown in Figure 1. Consider the function $L_{k,m}$ introduced in the proof,

$$L_{k,m}(x, \xi) = f(x) + \frac{1}{k^3}\|x - x^*\|^2 + \xi' g(x) - \frac{1}{2m}\|\xi\|^2.$$

Note that the term $(1/k^3)\|x - x^*\|^2$ ensures that x^* is a strict local minimum of the function $f(x) + (1/k^3)\|x - x^*\|^2$. To simplify the following discussion, let us assume that f is strictly convex, so that this term can be omitted from the definition of $L_{k,m}$. This assumption is satisfied by the above example if its cost function is changed to e^x, for which $f^* = 1$ and $q^* = 0$.

For any nonnegative vector $u \in \Re^r$, let $p^k(u)$ denote the optimal value of the problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g(x) \leq u, \\ & x \in X^k = X \cap \left\{x \;\middle|\; \|x - x^*\| \leq k\right\}. \end{array} \tag{14}$$

For each k and m, the saddle point of the function $L_{k,m}(x, \xi)$, denoted by $(x^{k,m}, \xi^{k,m})$, can be characterized in terms of $p^k(u)$ as follows.

The maximization of $L_{k,m}(x, \xi)$ over $\xi \geq 0$ for any fixed $x \in X^k$ yields

$$\xi = m g^+(x), \tag{15}$$

so that we have

$$\begin{aligned} L_{k,m}(x^{k,m}, \xi^{k,m}) &= \inf_{x \in X^k} \sup_{\xi \geq 0} \left\{ f(x) + \xi' g(x) - \frac{1}{2m}\|\xi\|^2 \right\} \\ &= \inf_{x \in X^k} \left\{ f(x) + \frac{m}{2}\|g^+(x)\|^2 \right\}. \end{aligned}$$

This minimization can also be written as

$$\begin{aligned} L_{k,m}(x^{k,m},\xi^{k,m}) &= \inf_{x\in X^k}\ \inf_{u\in\Re^r,\ g(x)\le u}\left\{f(x)+\frac{m}{2}\|u^+\|^2\right\} \\ &= \inf_{u\in\Re^r}\ \inf_{x\in X^k,\ g(x)\le u}\left\{f(x)+\frac{m}{2}\|u^+\|^2\right\} \qquad (16)\\ &= \inf_{u\in\Re^r}\left\{p^k(u)+\frac{m}{2}\|u^+\|^2\right\}. \end{aligned}$$

The vector $u^{k,m}=g(x^{k,m})$ attains the infimum in the preceding relation. This minimization can be visualized geometrically as in Figure 1. The point of contact of the graphs of the functions $p^k(u)$ and $L_{k,m}(x^{k,m},\xi^{k,m})-(m/2)\|u^+\|^2$ corresponds to the vector $u^{k,m}$ that attains the infimum in (16). A similar compactification-regularization technique is used in [Roc93, section 9].

We can also interpret $\xi^{k,m}$ in terms of the function p^k. In particular, the infimum of $L_{k,m}(x,\xi^{k,m})$ over $x\in X^k$ is attained at $x^{k,m}$, implying that

$$\begin{aligned} f(x^{k,m})+\xi^{k,m'}g(x^{k,m}) &= \inf_{x\in X^k}\{f(x)+\xi^{k,m'}g(x)\} \\ &= \inf_{u\in\Re^r}\{p^k(u)+\xi^{k,m'}u\}. \end{aligned}$$

Replacing $g(x^{k,m})$ by $u^{k,m}$ in the preceding relation, and using the fact that $x^{k,m}$ is feasible for problem (14) with $u=u^{k,m}$, we obtain

$$p^k(u^{k,m})\le f(x^{k,m})=\inf_{u\in\Re^r}\{p^k(u)+\xi^{k,m'}(u-u^{k,m})\}.$$

Thus, we see that

$$p^k(u^{k,m})\le p^k(u)+\xi^{k,m'}(u-u^{k,m}) \qquad \forall\ u\in\Re^r,$$

which, by the definition of the subgradient of a convex function, implies that

$$-\xi^{k,m}\in\partial p^k(u^{k,m})$$

(cf. Figure 1). It can be seen from this interpretation that the limit of $L_{k,m}(x^{k,m},\xi^{k,m})$ as $m\to\infty$ is equal to $p^k(0)$, which is equal to $f(x^*)$ for each k. The limit of the normalized sequence

$$\left\{\frac{(1,\xi^{k,m_k})}{\sqrt{1+\|\xi^{k,m_k}\|^2}}\right\}$$

as $k\to\infty$ yields the FJ-multiplier (μ_0^*,μ^*), and the sequence $\{x^{k,m_k}\}$ is used to construct the sequence that satisfies condition (CV) of the proposition.

3. Minimum-norm dual optimal solutions. In the preceding section we focused on the case where a primal optimal solution exists and we showed that the geometric multiplier of minimum norm is informative. Notice that a geometric multiplier is automatically a dual optimal solution. When there is a duality gap, there exists no geometric multiplier, even if there is a dual optimal solution. In this section we focus on the case where a dual optimal solution exists and we will see that, analogously, the dual optimal solution of minimum norm is informative. In particular, it satisfies a condition analogous to condition (CV), with primal optimal value f^* replaced by q^*. Consistent with our analysis in section 2, we call such a dual optimal solution *informative* [BNO03, section 6.6.2] since it indicates the constraints to relax and the rate of relaxation in order to obtain a primal cost reduction by an amount that is strictly greater than the size of the duality gap $f^* - q^*$.

We begin with the following proposition on the existence of an FJ-multiplier, which requires no additional assumptions on (P). It will be used to prove Lemma 4. This proposition is a direct extension of a well-known result [Lue79, page 217] and its proof may be found in [BNO03, Proposition 6.6.1]. A similar result is given in [Hes75, page 326], assuming (P) has an optimal solution.

PROPOSITION 3 (Fritz John conditions). *Consider the convex problem* (P)*, and assume that $f^* < \infty$. Then there exists an FJ-multiplier (μ_0^*, μ^*).*

If the scalar μ_0^* in the preceding proposition can be proved to be positive, then μ^*/μ_0^* is a geometric multiplier for problem (P). This can be used to show the existence of a geometric multiplier in the case where the Slater condition [Sla50] holds; i.e., there exists a vector $\overline{x} \in X$ such that $g(\overline{x}) < 0$. Indeed, in this case the scalar μ_0^* cannot be 0, since if it were, then according to the proposition, we would have

$$0 = \inf_{x \in X} \mu^{*\prime} g(x)$$

for some vector $\mu^* \geq 0$ with $\mu^* \neq 0$, while for this vector, we would also have $\mu^{*\prime} g(\overline{x}) < 0$, which is a contradiction.

Using Proposition 3, we have the following lemma which will be used to prove the next proposition, as well as Proposition 12 in the next section.

LEMMA 4. *Consider the convex problem* (P)*, and assume that $f^* < \infty$. For each $\delta > 0$, let*

$$f^\delta = \inf_{\substack{x \in X \\ g_j(x) \leq \delta,\, j=1,\dots,r,}} f(x). \tag{17}$$

Then the dual optimal value q^* satisfies $f^\delta \leq q^*$ for all $\delta > 0$ and

$$q^* = \lim_{\delta \downarrow 0} f^\delta.$$

Proof. We first note that either $\lim_{\delta \downarrow 0} f^\delta$ exists and is finite, or else $\lim_{\delta \downarrow 0} f^\delta = -\infty$, since f^δ is monotonically nondecreasing as $\delta \downarrow 0$, and $f^\delta \leq f^*$ for all $\delta > 0$. Since $f^* < \infty$, there exists some $\overline{x} \in X$ such that $g(\overline{x}) \leq 0$. Thus, for each $\delta > 0$ such that $f^\delta > -\infty$, the Slater condition is satisfied for problem (17), and by Proposition 3 and the subsequent discussion, there exists a $\mu^\delta \geq 0$ satisfying

$$\begin{aligned} f^\delta &= \inf_{x \in X} \left\{ f(x) + \mu^{\delta\prime} g(x) - \delta \sum_{j=1}^{r} \mu_j^\delta \right\} \\ &\leq \inf_{x \in X} \left\{ f(x) + \mu^{\delta\prime} g(x) \right\} \\ &= q(\mu^\delta) \\ &\leq q^*. \end{aligned}$$

For each $\delta > 0$ such that $f^\delta = -\infty$, we also have $f^\delta \leq q^*$, so that

$$f^\delta \leq q^* \qquad \forall\, \delta > 0.$$

By taking the limit as $\delta \downarrow 0$, we obtain

$$\lim_{\delta \downarrow 0} f^\delta \leq q^*.$$

To show the reverse inequality, we consider two cases: (1) $f^\delta > -\infty$ for all $\delta > 0$ that are sufficiently small, and (2) $f^\delta = -\infty$ for all $\delta > 0$. In case (1), for each $\delta > 0$ with $f^\delta > -\infty$, choose $x^\delta \in X$ such that $g_j(x^\delta) \leq \delta$ for all j and $f(x^\delta) \leq f^\delta + \delta$. Then, for any $\mu \geq 0$,

$$q(\mu) = \inf_{x \in X} \bigl\{ f(x) + \mu' g(x) \bigr\} \leq f(x^\delta) + \mu' g(x^\delta) \leq f^\delta + \delta + \delta \sum_{j=1}^{r} \mu_j.$$

Taking the limit as $\delta \downarrow 0$, we obtain

$$q(\mu) \leq \lim_{\delta \downarrow 0} f^\delta,$$

so that $q^* \leq \lim_{\delta \downarrow 0} f^\delta$. In case (2), choose $x^\delta \in X$ such that $g_j(x^\delta) \leq \delta$ for all j and $f(x^\delta) \leq -1/\delta$. Then, similarly, for any $\mu \geq 0$, we have

$$q(\mu) \leq f(x^\delta) + \mu' g(x^\delta) \leq -\frac{1}{\delta} + \delta \sum_{j=1}^{r} \mu_j,$$

so by taking $\delta \downarrow 0$, we obtain $q(\mu) = -\infty$ for all $\mu \geq 0$, and hence also $q^* = -\infty = \lim_{\downarrow 0} f^\delta$. □

Using Lemmas 1 and 4, we prove below the main result of this section, which shows under very general assumptions that the minimum-norm dual optimal solution is informative.

Proposition 5 (existence of informative dual optimal solution). *Consider the convex problem* (P) *under Assumption* 1 *(closedness), and assume that $f^* < \infty$ and $-\infty < q^*$. If there exists a dual optimal solution, then the dual optimal solution μ^* of minimum norm satisfies the following condition* (dCV). *Moreover, it is the only dual optimal solution that satisfies this condition.*

(dCV) *If $\mu^* \neq 0$, then there exists a sequence $\{x^k\} \subset X$ of infeasible points that satisfies*

$$f(x^k) \to q^*, \qquad g^+(x^k) \to 0, \tag{18}$$

$$\frac{q^* - f(x^k)}{\|g^+(x^k)\|} \to \|\mu^*\|, \tag{19}$$

$$\frac{g^+(x^k)}{\|g^+(x^k)\|} \to \frac{\mu^*}{\|\mu^*\|}. \tag{20}$$

Proof. Let μ^* be the dual optimal solution of minimum norm. Assume that $\mu^* \neq 0$. For $k = 1, 2, \ldots$, consider the problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in X, \qquad g_j(x) \leq \dfrac{1}{k^4}, \ k = 1, \ldots, r. \end{array}$$

By Lemma 4, for each k, the optimal value of this problem is less than or equal to q^*. Since q^* is finite (in view of the assumptions $-\infty < q^*$ and $f^* < \infty$, and the weak duality relation $q^* \le f^*$), we may select for each k a vector $\tilde{x}^k \in X$ that satisfies

$$f(\tilde{x}^k) \le q^* + \frac{1}{k^2}, \qquad g_j(\tilde{x}^k) \le \frac{1}{k^4},\ j = 1, \dots, r.$$

Consider also the problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_j(x) \le \dfrac{1}{k^4},\ j = 1, \dots, r, \\ & x \in \tilde{X}^k = X \cap \left\{ x \;\middle|\; \|x\| \le k \left(\max_{1 \le i \le k} \|\tilde{x}^i\| + 1 \right) \right\}. \end{array}$$

By the closedness assumption, f and g_j are closed and convex when restricted to X, so they are closed, convex, and coercive when restricted to $\tilde{X}^k$. Thus, the problem has an optimal solution, which we denote by $\overline{x}^k$. Note that, since $\tilde{x}^k$ belongs to the feasible solution set of this problem, we have

$$f(\overline{x}^k) \le f(\tilde{x}^k) \le q^* + \frac{1}{k^2}. \tag{21}$$

For each k, we consider the saddle function

$$L_k(x, \mu) = f(x) + \mu' g(x) - \frac{\|\mu\|^2}{2k}$$

and the set

$$X^k = \tilde{X}^k \cap \big\{ x \mid g_j(x) \le k,\ j = 1, \dots, r \big\}.$$

We note that $L_k(x, \mu)$, for fixed $\mu \ge 0$, is closed, convex, and coercive in x, when restricted to X^k, and negative definite quadratic in μ for fixed x. Hence, using the saddle point theorem (e.g., [BNO03, Proposition 2.6.9]), we can assert that L_k has a saddle point over $x \in X^k$ and $\mu \ge 0$, denoted by (x^k, μ^k).

Since L_k is quadratic in μ, the supremum of $L_k(x^k, \mu)$ over $\mu \ge 0$ is attained at

$$\mu^k = k g^+(x^k). \tag{22}$$

Similarly, the infimum in $\inf_{x \in X^k} L_k(x, \mu^k)$ is attained at x^k, implying that

$$\begin{aligned} f(x^k) + \mu^{k'} g(x^k) &= \inf_{x \in X^k} \big\{ f(x) + \mu^{k'} g(x) \big\} \\ &= \inf_{x \in X^k} \big\{ f(x) + k g^+(x^k)' g(x) \big\} \\ &\le \inf_{x \in X^k,\ g_j(x) \le \frac{1}{k^4},\, j=1,\dots,r,} \left\{ f(x) + k \sum_{j=1}^r g_j^+(x^k)' g_j(x) \right\} \\ &\le \inf_{x \in X^k,\ g_j(x) \le \frac{1}{k^4},\, j=1,\dots,r,} \left\{ f(x) + \frac{r}{k^2} \right\} \\ &= f(\overline{x}^k) + \frac{r}{k^2} \\ &\le q^* + \frac{r+1}{k^2}, \end{aligned} \tag{23}$$

where the second inequality holds in view of the fact that $x^k \in X^k$, implying that $g_j^+(x^k) \le k$, $j = 1, \dots, r$, and the third inequality follows from (21).

We also have

$$\begin{aligned} L_k(x^k, \mu^k) &= \sup_{\mu \ge 0} \inf_{x \in X^k} L_k(x, \mu) \\ &\ge \sup_{\mu \ge 0} \inf_{x \in X} L_k(x, \mu) \\ &= \sup_{\mu \ge 0} \left\{ \inf_{x \in X} \big\{ f(x) + \mu' g(x) \big\} - \frac{\|\mu\|^2}{2k} \right\} \\ &= \sup_{\mu \ge 0} \left\{ q(\mu) - \frac{\|\mu\|^2}{2k} \right\} \\ &\ge q(\mu^*) - \frac{\|\mu^*\|^2}{2k} \\ &= q^* - \frac{\|\mu^*\|^2}{2k}, \end{aligned} \tag{24}$$

where we recall that μ^* is the dual optimal solution with minimum norm.

Combining (24) and (23), we obtain

$$\begin{aligned} q^* - \frac{1}{2k}\|\mu^*\|^2 \le L_k(x^k, \mu^k) & \\ = f(x^k) + \mu^{k'} g(x^k) - \frac{1}{2k}\|\mu^k\|^2 & \\ \le q^* + \frac{r+1}{k^2} - \frac{1}{2k}\|\mu^k\|^2. & \end{aligned} \tag{25}$$

This relation shows that $\|\mu^k\|^2 \le \|\mu^*\|^2 + 2(r+1)/k$, so the sequence $\{\mu^k\}$ is bounded. Let $\overline{\mu}$ be a cluster point of $\{\mu^k\}$. Without loss of generality, we assume that the entire sequence $\{\mu^k\}$ converges to $\overline{\mu}$. We also have from (25) that

$$\lim_{k \to \infty} \big\{ f(x^k) + \mu^{k'} g(x^k) \big\} = q^*.$$

Hence, taking the limit as $k \to \infty$ in (23) yields

$$q^* \le \inf_{x \in X} \big\{ f(x) + \overline{\mu}' g(x) \big\} = q(\overline{\mu}) \le q^*.$$

Hence $\overline{\mu}$ is a dual optimal solution, and since $\|\overline{\mu}\| \le \|\mu^*\|$ (which follows by taking the limit in (25)), by using the minimum norm property of μ^*, we conclude that any cluster point $\overline{\mu}$ of μ^k must be equal to μ^*. Thus $\mu^k \to \mu^*$, and using (25), we obtain

$$\lim_{k \to \infty} k\big(L_k(x^k, \mu^k) - q^*\big) = -\frac{1}{2}\|\mu^*\|^2. \tag{26}$$

Using (22), it follows that

$$L_k(x^k, \mu^k) = \sup_{\mu \ge 0} L_k(x^k, \mu) = f(x^k) + \frac{1}{2k}\|\mu^k\|^2,$$

which combined with (26) yields

$$\lim_{k \to \infty} k\big(f(x^k) - q^*\big) = -\|\mu^*\|^2,$$

implying that $f(x^k) < q^*$ for all sufficiently large k, since $\mu^* \neq 0$. Since $\mu^k \to \mu^*$, (22) also implies that

$$\lim_{k\to\infty} kg^+(x^k) = \mu^*.$$

It follows that the sequence $\{x^k\}$ satisfies (18), (19), and (20). Moreover, Lemma 1 shows that $\{x^k\}$ satisfies (19) only when μ^* is the dual optimal solution of minimum norm. This completes the proof. □

Our next result of this section shows that Assumption 1 in Proposition 5 can in fact be relaxed. We denote by $\overline{f}$ the closure of f, i.e., the function whose epigraph is the closure of f. Similarly, for each j, we denote by $\overline{g}_j$ the closure of g_j. A key fact we use is that *replacing f and g_j by their closures does not affect the closure of the primal function, and hence also the dual function.* This is based on the following lemma on the closedness of functions generated by partial minimization.

LEMMA 6. *Consider a function $F : \Re^{n+r} \mapsto (-\infty, \infty]$ and the function $p : \Re^n \mapsto [-\infty, \infty]$ defined by*

$$p(u) = \inf_{x\in\Re^n} F(x, u).$$

Then the following hold:

(a)

$$P(\mathrm{epi}(F)) \subset \mathrm{epi}(p) \subset \mathrm{cl}(P(\mathrm{epi}(F))), \tag{27}$$

$$P(\mathrm{cl}(\mathrm{epi}(F))) \subset \mathrm{cl}(\mathrm{epi}(p)), \tag{28}$$

where $P(\cdot)$ denotes projection on the space of (u, w), i.e., $P(x, u, w) = (u, w)$.

(b) *If $\overline{F}$ is the closure of F and $\overline{p}$ is defined by*

$$\overline{p}(u) = \inf_{x\in\Re^n} \overline{F}(x, u),$$

then the closures of p and $\overline{p}$ coincide.

Proof. (a) The left-hand side of (27) follows from the definition

$$\mathrm{epi}(p) = \left\{ (u, w) \;\middle|\; \inf_{x\in\Re^n} F(x, u) \le w \right\}.$$

To show the right-hand side of (27), note that for any $(u, w) \in \mathrm{epi}(p)$ and every integer $k \ge 1$, there exists an x^k such that $(x^k, u, w + 1/k) \in \mathrm{epi}(F)$, so that $(u, w + 1/k) \in P(\mathrm{epi}(F))$ and $(u, w) \in \mathrm{cl}(P(\mathrm{epi}(F)))$.

To show (28), let $(\overline{u}, \overline{w})$ belong to $P(\mathrm{cl}(\mathrm{epi}(F)))$. Then there exists $\overline{x}$ such that $(\overline{x}, \overline{u}, \overline{w}) \in \mathrm{cl}\big(\mathrm{epi}(F)\big)$, and hence there is a sequence $(x^k, u^k, w^k) \in \mathrm{epi}(F)$ such that $x^k \to \overline{x}$, $u^k \to \overline{u}$, and $w^k \to \overline{w}$. Thus we have $p(u^k) \le F(x^k, u^k) \le w^k$, implying that $(u^k, w^k) \in \mathrm{epi}(p)$ for all k. It follows that $(\overline{u}, \overline{w}) \in \mathrm{cl}\big(\mathrm{epi}(p)\big)$.

(b) By taking closure in (27), we see that

$$\mathrm{cl}(\mathrm{epi}(p)) = \mathrm{cl}(P(\mathrm{epi}(F))), \tag{29}$$

and by replacing F with $\overline{F}$, we also have

$$\mathrm{cl}(\mathrm{epi}(\overline{p})) = \mathrm{cl}(P(\mathrm{epi}(\overline{F}))). \tag{30}$$

On the other hand, by taking closure in (28), we have

$$\mathrm{cl}(P(\mathrm{epi}(\overline{F}))) \subset \mathrm{cl}(P(\mathrm{epi}(F))),$$

which implies that

$$\mathrm{cl}(P(\mathrm{epi}(\overline{F}))) = \mathrm{cl}(P(\mathrm{epi}(F))). \tag{31}$$

By combining (29)–(31), we see that

$$\mathrm{cl}\big(\mathrm{epi}(p)\big) = \mathrm{cl}\big(\mathrm{epi}(\overline{p})\big). \qquad \square$$

Using Lemmas 1 and 6, we now prove the next main result of this section.

PROPOSITION 7 (relaxing closedness assumption in Proposition 5). *Consider the convex problem* (P), *and assume that* $f^* < \infty$, $-\infty < q^*$, *and* $\mathrm{dom}(\overline{f}) = \mathrm{dom}(\overline{g}_j)$, $j = 1, \ldots, r$. *If* μ^* *is the dual optimal solution of minimum norm, then it satisfies condition* (*dCV*) *of Proposition* 5. *Moreover, it is the only dual optimal solution that satisfies this condition.*

Proof. We apply Lemma 6 to the primal function $p(u)$, which is defined by partial minimization over $x \in \Re^n$ of the extended real-valued function

$$F(x,u) = \begin{cases} f(x) & \text{if } x \in X,\ g(x) \le u, \\ \infty & \text{otherwise.} \end{cases}$$

Note that the closure of F is

$$\overline{F}(x,u) = \begin{cases} \overline{f}(x) & \text{if } x \in \overline{X},\ \overline{g}(x) \le u, \\ \infty & \text{otherwise,} \end{cases}$$

where $\overline{g} = (\overline{g}_1, \ldots, \overline{g}_r)'$ and $\overline{X} = \mathrm{dom}(\overline{f}) = \mathrm{dom}(\overline{g}_j)$, $j = 1, \ldots, r$.[2] Thus, by Lemma 6, replacing X, f, and g with $\overline{X}$, $\overline{f}$, and $\overline{g}$ does not change the closure of the primal function, and therefore does not change the dual function.

Assume $\mu^* \neq 0$. By Proposition 5, there exists a sequence $\{x^k\} \subset \overline{X}$ of infeasible points that satisfies

$$\frac{q^* - \overline{f}(x^k)}{\|\overline{g}^+(x^k)\|} \to \|\mu^*\|, \qquad \frac{\overline{g}^+(x^k)}{\|\overline{g}^+(x^k)\|} \to \frac{\mu^*}{\|\mu^*\|}, \qquad \|\overline{g}^+(x^k)\| \to 0.$$

We will now perturb the sequence $\{x^k\}$ so that it lies in $\mathrm{ri}(X)$, while it still satisfies the preceding relations. Indeed, fix any $\overline{x} \in \mathrm{ri}(X)$. For each k, we can choose a sufficiently small $\epsilon \in (0,1)$ such that $\overline{f}(\epsilon\overline{x} + (1-\epsilon)x^k)$ and $\|\overline{g}^+(\epsilon\overline{x} + (1-\epsilon)x^k)\|$ are arbitrarily close to $\overline{f}(x^k)$ and $\|\overline{g}^+(x^k)\|$, respectively. This is possible because $\overline{f}$,

[2] Why? By definition of the closure of F, $\overline{F}(x,u) = \liminf_{(x^k,u^k)\to(x,u)} F(x^k,u^k)$. Suppose $\overline{F}(x,u) < \infty$. Then there exist $x^k \in X$ and u^k such that $(x^k,u^k) \to (x,u)$, $f(x^k) \to \overline{F}(x,u)$, and $g(x^k) \le u^k$ for all $k = 1, 2, \ldots$. Passing to the limit yields $\overline{f}(x) \le \overline{F}(x,u)$ and $\overline{g}(x) \le u$. Conversely, suppose $\overline{f}(x) < \infty$ and $\overline{g}(x) \le u$. Fix any $\overline{x} \in \mathrm{ri}(X)$, and let $x^\epsilon = (1-\epsilon)x + \epsilon\overline{x}$, $u^\epsilon = (1-\epsilon)u + \epsilon\overline{u}$, where $\overline{u} = g(\overline{x})$. Then $x^\epsilon \in \mathrm{ri}(X) = \mathrm{ri}(\overline{X})$ and $g(x^\epsilon) \le u^\epsilon$ for $\epsilon \in (0,1)$. Since $\overline{f}$ coincides with f on $\mathrm{ri}(\overline{X})$ and $\overline{f}$ is continuous along any line segment in $\overline{X}$, this implies

$$\lim_{\epsilon\to 0} f(x^\epsilon) = \lim_{\epsilon\to 0} \overline{f}(x^\epsilon) = \overline{f}(x).$$

Thus $\lim_{\epsilon\to 0} F(x^\epsilon,u^\epsilon) = \overline{f}(x)$, implying $\overline{F}(x,u) \le \overline{f}(x)$.

$\overline{g}_1, \dots, \overline{g}_r$ are closed and hence continuous along the line segment that connects x^k and $\overline{x}$. Thus, for each k, we can choose $\epsilon_k \in (0,1)$ so that the corresponding vector $\overline{x}^k = \epsilon_k \overline{x} + (1-\epsilon_k)x^k$ satisfies

$$\left| \frac{q^* - \overline{f}(x^k)}{\|\overline{g}^+(x^k)\|} - \frac{q^* - \overline{f}(\overline{x}^k)}{\|\overline{g}^+(\overline{x}^k)\|} \right| \le \frac{1}{k}, \qquad \left| \frac{\overline{g}^+(x^k)}{\|\overline{g}^+(x^k)\|} - \frac{\overline{g}^+(\overline{x}^k)}{\|\overline{g}^+(\overline{x}^k)\|} \right| \le \frac{1}{k}, \qquad \|\overline{g}^+(\overline{x}^k)\| \to 0.$$

Since $\overline{x}$ lies in $\mathrm{ri}(X) = \mathrm{ri}(\overline{X})$, every point in the open line segment that connects x^k and $\overline{x}$, including $\overline{x}^k$, lies in $\mathrm{ri}(X)$, so that $\overline{f}(\overline{x}^k) = f(\overline{x}^k)$ and $\overline{g}(\overline{x}^k) = g(\overline{x}^k)$. We thus obtain a sequence $\{\overline{x}^k\}$ in the relative interior of X satisfying

$$\frac{q^* - f(\overline{x}^k)}{\|g^+(\overline{x}^k)\|} \to \|\mu^*\|, \qquad \frac{g^+(\overline{x}^k)}{\|g^+(\overline{x}^k)\|} \to \frac{\mu^*}{\|\mu^*\|}, \qquad \|g^+(\overline{x}^k)\| \to 0.$$

The first and the third relations imply $f(\overline{x}^k) \to q^*$. Thus μ^* satisfies condition (dCV) of Proposition 5. By Lemma 1, μ^* is the only dual optimal solution that satisfies this condition. □

3.1. Fritz John conditions and constraint qualifications.

3.1. Fritz John conditions and constraint qualifications. We close this section by discussing the connection of the Fritz John conditions with classical constraint qualifications that guarantee the existence of a geometric multiplier (and hence also the existence of a dual optimal solution, which makes the analysis of the present section applicable). As mentioned earlier in this section, the classical Fritz John conditions of Proposition 3 can be used to assert the existence of a geometric multiplier when the Slater condition holds. However, Proposition 3 is insufficient to show that a geometric multiplier exists in the case of affine constraints. The following proposition strengthens the Fritz John conditions for this case, so that they suffice for the proof of the corresponding existence result. In contrast to the Kuhn–Tucker theory [Hes75], [Roc70], this does not assume (P) has an optimal solution.

PROPOSITION 8 (Fritz John conditions for affine constraints). *Consider the convex problem* (P)*, and assume that the functions $g_1, \dots, g_r$ are affine, and $f^* < \infty$. Then there exists an FJ-multiplier (μ_0^*, μ^*) satisfying the following condition:*

(CV$'$) *If $\mu^* \ne 0$, then there exists a vector $\tilde{x} \in X$ satisfying*

$$f(\tilde{x}) < f^*, \qquad {\mu^*}' g(\tilde{x}) > 0.$$

Proof. If $\inf_{x \in X} f(x) = f^*$, then $\mu_0^* = 1$ and $\mu^* = 0$ form an FJ-multiplier, and condition (CV$'$) is automatically satisfied. We will thus assume that $\inf_{x \in X} f(x) < f^*$, which also implies that f^* is finite.

Let the affine constraint function be represented as

$$g(x) = Ax - b$$

for some real matrix A and vector b. Consider the nonempty convex sets

$$C_1 = \big\{(x, w) \mid \text{there is a vector } x \in X \text{ such that } f(x) < w\big\},$$
$$C_2 = \big\{(x, f^*) \mid Ax - b \le 0\big\}.$$

Note that C_1 and C_2 are disjoint. The reason is that if $(x, f^*) \in C_1 \cap C_2$, then we must have $x \in X$, $Ax - b \le 0$, and $f(x) < f^*$, contradicting the fact that f^* is the optimal value of the problem.

Since C_2 is polyhedral, by the polyhedral proper separation theorem (see [Roc70, Theorem 20.2] or [BNO03, Proposition 3.5.1]), there exists a hyperplane that separates C_1 and C_2 and does not contain C_1, i.e., there exists a vector (ξ, μ_0^*) such that

$$\mu_0^* f^* + \xi' z \le \mu_0^* w + \xi' x \qquad \forall\ x \in X, w, z \text{ with } f(x) < w,\ Az - b \le 0, \tag{32}$$

$$\inf_{(x,w)\in C_1} \{\mu_0^* w + \xi' x\} < \sup_{(x,w)\in C_1} \{\mu_0^* w + \xi' x\}.$$

These relations imply that

$$\mu_0^* f^* + \sup_{Az-b\le 0} \xi' z \le \inf_{(x,w)\in C_1} \{\mu_0^* w + \xi' x\} < \sup_{(x,w)\in C_1} \{\mu_0^* w + \xi' x\}, \tag{33}$$

and that $\mu_0^* \ge 0$ (since w can be taken arbitrarily large in (32)).

Consider the linear program in (33):

$$\begin{array}{ll} \text{maximize} & \xi' z \\ \text{subject to} & Az - b \le 0. \end{array}$$

By (33), this program is bounded and therefore it has an optimal solution, which we denote by z^*. The dual of this program is

$$\begin{array}{ll} \text{minimize} & b' \mu \\ \text{subject to} & \xi = A' \mu, \qquad \mu \ge 0. \end{array}$$

By linear programming duality, it follows that this problem has a dual optimal solution $\mu^* \ge 0$ satisfying

$$\sup_{Az-b\le 0} \xi' z = \xi' z^* = \mu^{*'} b, \qquad \xi = A' \mu^*. \tag{34}$$

Note that μ_0^* and μ^* satisfy the nonnegativity condition (ii). Furthermore, we cannot have both $\mu_0^* = 0$ and $\mu^* = 0$, since then by (34) we would also have $\xi = 0$, and (33) would be violated. Thus, μ_0^* and μ^* also satisfy condition (iii) in the definition of an FJ-multiplier.

From (33), we have

$$\mu_0^* f^* + \sup_{Az-b\le 0} \xi' z \le \mu_0^* w + \xi' x \qquad \forall\ x \in X \text{ with } f(x) < w,$$

which together with (34) implies that

$$\mu_0^* f^* + \mu^{*'} b \le \mu_0^* w + \mu^{*'} A x \qquad \forall\ x \in X \text{ with } f(x) < w,$$

or

$$\mu_0^* f^* \le \inf_{x \in X,\, f(x) < w} \big\{\mu_0^* w + \mu^{*'} (Ax - b)\big\}. \tag{35}$$

Similarly, from (33) and (34), we have

$$\mu_0^* f^* < \sup_{x \in X,\, f(x) < w} \big\{\mu_0^* w + \mu^{*'} (Ax - b)\big\}. \tag{36}$$

Using (35), we obtain

$$\begin{aligned}\mu_0^* f^* &\leq \inf_{x\in X}\{\mu_0^* f(x) + \mu^{*'}(Ax-b)\}\\ &\leq \inf_{x\in X,\, Ax-b\leq 0}\{\mu_0^* f(x) + \mu^{*'}(Ax-b)\}\\ &\leq \inf_{x\in X,\, Ax-b\leq 0}\mu_0^* f(x)\\ &= \mu_0^* f^*.\end{aligned}$$

Hence, equality holds throughout above, which proves condition (i) in the definition of an FJ-multiplier.

We will now show that the vector μ^* also satisfies condition (CV$'$). To this end, we consider separately the cases where $\mu_0^* > 0$ and $\mu_0^* = 0$.

If $\mu_0^* > 0$, let $\tilde{x} \in X$ be such that $f(\tilde{x}) < f^*$ (based on our earlier assumption that $\inf_{x\in X} f(x) < f^*$). Then condition (i) yields

$$\mu_0^* f^* \leq \mu_0^* f(\tilde{x}) + \mu^{*\prime}(A\tilde{x} - b),$$

implying that $0 < \mu_0^*(f^* - f(\tilde{x})) \leq \mu^{*\prime}(A\tilde{x} - b)$, and showing condition (CV$'$).

If $\mu_0^* = 0$, condition (i) together with (36) yields

$$0 = \inf_{x\in X}\mu^{*\prime}(Ax-b) < \sup_{x\in X}\mu^{*\prime}(Ax-b). \tag{37}$$

The above relation implies the existence of a vector $\hat{x} \in X$ such that $\mu^{*'}(A\hat{x} - b) > 0$. Let $\overline{x} \in X$ be such that $f(\overline{x}) < f^*$, and consider a vector of the form

$$\tilde{x} = \alpha\hat{x} + (1-\alpha)\overline{x},$$

where $\alpha \in (0, 1)$. Note that $\tilde{x} \in X$ for all $\alpha \in (0, 1)$, since X is convex. From (37), we have $\mu^{*\prime}(A\overline{x} - b) \geq 0$ which combined with the inequality $\mu^{*'}(A\hat{x} - b) > 0$ implies that

$$\mu^{*\prime}(A\tilde{x} - b) = \alpha\mu^{*\prime}(A\hat{x} - b) + (1-\alpha)\mu^{*\prime}(A\overline{x} - b) > 0 \qquad \forall\ \alpha \in (0, 1). \tag{38}$$

Furthermore, since f is convex, we have

$$f(\tilde{x}) \leq \alpha f(\hat{x}) + (1-\alpha)f(\overline{x}) = f^* + \big(f(\overline{x}) - f^*\big) + \alpha\big(f(\hat{x}) - f(\overline{x})\big) \qquad \forall\ \alpha \in (0, 1).$$

Thus, for α small enough so that $\alpha\big(f(\hat{x}) - f(\overline{x})\big) < f^* - f(\overline{x})$, we have $f(\tilde{x}) < f^*$ as well as $\mu^{*\prime}(A\tilde{x} - b) > 0$ (cf. (38)). □

We now introduce the following constraint qualification, which is analogous to one introduced for nonconvex problems by Bertsekas and Ozdaglar [BeO02].

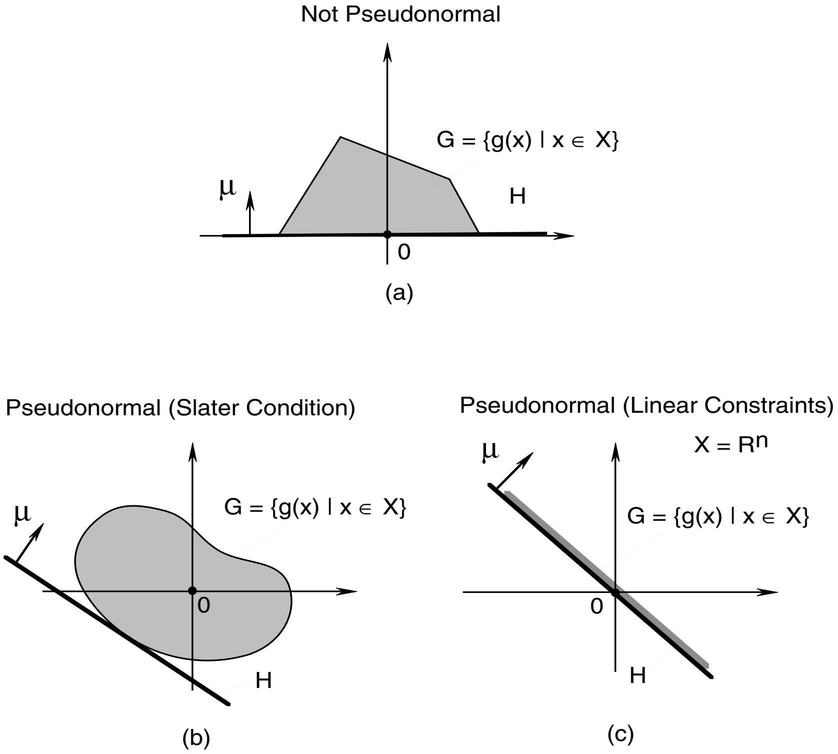
Definition 9. *The constraint set of the convex problem* (P) *is said to be pseudonormal if there does not exist a vector $\mu \geq 0$ and a vector $\tilde{x} \in X$ satisfying the following conditions:*

(i) $0 = \inf_{x\in X} \mu' g(x)$.

(ii) $\mu' g(\tilde{x}) > 0$.

To provide a geometric interpretation of pseudonormality, let us introduce the set

$$G = \big\{g(x) \mid x \in X\big\}$$



Not Pseudonormal

(a)

Pseudonormal (Slater Condition)

(b)

Pseudonormal (Linear Constraints)

(c)

FIG. 2. *Geometric multiplier interpretation of pseudonormality. Consider the set*

$$G = \{g(x) \mid x \in X\}$$

and the hyperplanes that support this set. For feasibility, G should intersect the nonpositive orthant $\{z \mid z \leq 0\}$. The first condition $[0 = \inf_{x \in X} \mu' g(x)]$ in the definition of pseudonormality means that there is a hyperplane H with normal $\mu \geq 0$, which passes through 0, supports G, and contains G in its positive halfspace (note that, as illustrated in figure (b), this cannot happen if G intersects the interior of the nonpositive orthant; cf. the Slater criterion). The second condition means that H does not fully contain G (cf. figures (a) and (c)). If the Slater criterion holds, the first condition cannot be satisfied. If the linearity criterion holds, the set G is an affine set and the second condition cannot be satisfied (this depends critically on X being an affine set rather than X being a general polyhedron).

and consider hyperplanes that support this set and pass through 0. As Figure 2 illustrates, *pseudonormality means that there is no hyperplane with a normal $\mu \geq 0$ that properly separates the sets $\{0\}$ and G, and contains G in its positive halfspace.*

It is evident (see also Figure 2) that pseudonormality holds under the Slater condition, i.e., if there exists an $\bar{x} \in X$ such that $g(\bar{x}) < 0$. Proposition 8 also shows that if $f^* < \infty$, the constraint functions $g_1, \dots, g_r$ are affine, and the constraint set is pseudonormal, then there exists a geometric multiplier satisfying the special condition (CV') of Proposition 8. As illustrated also in Figure 2, the constraint set is pseudonormal if X is an affine set and g_j, $j = 1, \dots, r$, are affine functions. In conclusion, *if $f^* < \infty$, and either the Slater condition holds, or X and $g_1, \dots, g_r$ are affine, then the constraint set is pseudonormal, and a geometric multiplier is guaranteed to exist.* Since in this case there is no duality gap, Proposition 7 guarantees the existence of a geometric multiplier (the one of minimum norm) that satisfies the corresponding (CV) condition and sensitivity properties.

Finally, consider the question of pseudonormality and existence of geometric multipliers in the case where X is the intersection of a polyhedral set and a convex set C, and there exists a feasible solution that belongs to the relative interior of C. Then, the constraint set need not be pseudonormal, as Figure 2(a) illustrates. However, it is pseudonormal in the extended representation (i.e., when the affine inequalities that represent the polyhedral part are lumped with the remaining affine inequality constraints), and it follows that there exists a geometric multiplier in the extended representation. From this, it follows that there exists a geometric multiplier in the original representation as well (see Exercise 6.2 of [BNO03]).

4. Fritz John conditions when there is no optimal solution. In the preceding sections, we studied sensitivity properties of the geometric multiplier or dual optimal solution of minimum norm in the case where there exists a primal optimal solution or a dual optimal solution. In this section and the next section, we allow the problem to have neither a primal nor a dual optimal solution, and we develop several analogous results.

The Fritz John conditions of Propositions 3 and 8 are weaker than Proposition 2 in that they do not include conditions analogous to condition (CV). Unfortunately, such a condition does not hold in the absence of additional assumptions, as can be seen from the following example.

Example 1. Consider the one-dimensional problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g(x) = x \leq 0,\ x \in X = \{x \mid x \geq 0\}, \end{array}$$

where

$$f(x) = \begin{cases} -1 & \text{if } x > 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x < 0. \end{cases}$$

Then f is convex over X, and the assumptions of Propositions 3 and 8 are satisfied. Indeed, each FJ-multiplier must have the form $\mu_0^* = 0$ and $\mu^* > 0$ (cf. Figure 3). However, here we have $f^* = 0$, and for all x with $g(x) > 0$, we have $x > 0$ and $f(x) = -1$. Thus, there is no sequence $\{x^k\} \subset X$ satisfying (2)–(4).

The following proposition imposes the stronger closedness assumption in order to derive an enhanced set of Fritz John conditions analogous to those in Proposition 2. The proof uses ideas that are similar to the ones of the proof of Proposition 2, but is more complicated because an optimal solution of (P) may not exist. In particular, we approximate X by a sequence of expanding bounded convex subsets and we work with an optimal solution of the corresponding problem.

Proposition 10 (enhanced Fritz John conditions). *Consider the convex problem* (P) *under Assumption* 1 *(closedness), and assume that $f^* < \infty$. Then there exists an FJ-multiplier (μ_0^*, μ^*) satisfying the following condition* (CV). *Moreover, if $\mu_0^* \neq 0$, then μ^*/μ_0^* must be the geometric multiplier of minimum norm.*

(CV) *If $\mu^* \neq 0$, then there exists a sequence $\{x^k\} \subset X$ of infeasible points that satisfies* (2), (3), *and* (4).

Proof. If $f(x) \geq f^*$ for all $x \in X$, then $\mu_0^* = 1$ and $\mu^* = 0$ form an FJ-multiplier, and condition (CV) is satisfied. Moreover, (CV) (in particular, (3)) is satisfied by only $\mu^* = 0$. We will thus assume that there exists some $\overline{x} \in X$ such that $f(\overline{x}) < f^*$.

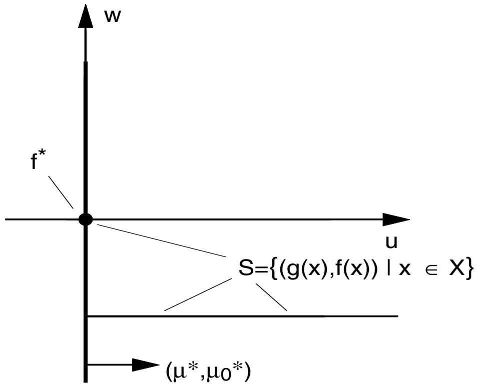


FIG. 3. *Illustration of the set* $S = \{(g(x), f(x)) \mid x \in X\}$ *in Example* 1. *Even though* $\mu^* > 0$, *there is no sequence* $\{x^k\} \subset X$ *such that* $g(x^k) > 0$ *for all* k, *and* $f(x^k) \to f^*$.

In this case, f^* is finite. Consider the problem

$$\text{(39)} \qquad \begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in X^k, \qquad g(x) \le 0, \end{array}$$

where

$$X^k = X \cap \{x \mid \|x\| \le \beta k\}, \qquad k = 1, 2, \dots ,$$

and β is a scalar that is large enough so that for all k, the constraint set $\{x \in X^k \mid g(x) \le 0\}$ is nonempty. Since f and g_j are closed and convex when restricted to X, they are closed, convex, and coercive when restricted to X^k. Hence, problem (39) has an optimal solution, which we denote by $\overline{x}^k$. Since this is a more constrained problem than the original, we have $f^* \le f(\overline{x}^k)$ and $f(\overline{x}^k) \downarrow f^*$ as $k \to \infty$. Let

$$\gamma^k = f(\overline{x}^k) - f^*.$$

Note that if $\gamma^k = 0$ for some k, then $\overline{x}^k$ is an optimal solution for problem (P), and the result follows from Proposition 2 on enhanced Fritz John conditions for convex problems with an optimal solution. Therefore, we assume that $\gamma^k > 0$ for all k.

For positive integers k and positive scalars m, we consider the saddle function

$$L_{k,m}(x, \xi) = f(x) + \frac{(\gamma^k)^2}{4k^2}\|x - \overline{x}^k\|^2 + \xi' g(x) - \frac{\|\xi\|^2}{2m}.$$

We note that $L_{k,m}(x, \xi)$, viewed as a function from X^k to $\Re$, for fixed $\xi \ge 0$, is closed, convex, and coercive, in view of the closedness assumption. Furthermore, $L_{k,m}(x, \xi)$ is negative definite quadratic in ξ for fixed x. Hence, we can use the saddle point theorem (e.g., [BNO03, Proposition 2.6.9]) to assert that $L_{k,m}$ has a saddle point over $x \in X^k$ and $\xi \ge 0$, which we denote by $(x^{k,m}, \xi^{k,m})$.

We now derive several properties of the saddle points $(x^{k,m}, \xi^{k,m})$, which set the stage for the main argument. The first of these properties is

$$f(x^{k,m}) \le L_{k,m}(x^{k,m}, \xi^{k,m}) \le f(\overline{x}^k),$$

which is shown in the next paragraph.

The infimum of $L_{k,m}(x, \xi^{k,m})$ over $x \in X^k$ is attained at $x^{k,m}$, implying that

$$\begin{aligned} & f(x^{k,m}) + \frac{(\gamma^k)^2}{4k^2}\|x^{k,m} - \overline{x}^k\|^2 + \xi^{k,m\prime} g(x^{k,m}) \\ &\quad = \inf_{x \in X^k} \left\{ f(x) + \frac{(\gamma^k)^2}{4k^2}\|x - \overline{x}^k\|^2 + \xi^{k,m\prime} g(x) \right\} \\ &\quad \le \inf_{x \in X^k,\, g(x) \le 0} \left\{ f(x) + \frac{(\gamma^k)^2}{4k^2}\|x - \overline{x}^k\|^2 + \xi^{k,m\prime} g(x) \right\} \\ &\quad \le \inf_{x \in X^k,\, g(x) \le 0} \left\{ f(x) + \frac{(\gamma^k)^2}{4k^2}\|x - \overline{x}^k\|^2 \right\} \\ &\quad = f(\overline{x}^k). \end{aligned} \tag{40}$$

Hence, we have

$$\begin{aligned} L_{k,m}(x^{k,m}, \xi^{k,m}) &= f(x^{k,m}) + \frac{(\gamma^k)^2}{4k^2}\|x^{k,m} - \overline{x}^k\|^2 + \xi^{k,m\prime} g(x^{k,m}) - \frac{1}{2m}\|\xi^{k,m}\|^2 \\ &\le f(x^{k,m}) + \frac{(\gamma^k)^2}{4k^2}\|x^{k,m} - \overline{x}^k\|^2 + \xi^{k,m\prime} g(x^{k,m}) \\ &\le f(\overline{x}^k). \end{aligned} \tag{41}$$

Since $L_{k,m}$ is quadratic in ξ, the supremum of $L_{k,m}(x^{k,m}, \xi)$ over $\xi \ge 0$ is attained at

$$\xi^{k,m} = m g^+(x^{k,m}). \tag{42}$$

This implies that

$$\begin{aligned} L_{k,m}(x^{k,m}, \xi^{k,m}) &= f(x^{k,m}) + \frac{(\gamma^k)^2}{4k^2}\|x^{k,m} - \overline{x}^k\|^2 + \frac{m}{2}\|g^+(x^{k,m})\|^2 \\ &\ge f(x^{k,m}). \end{aligned} \tag{43}$$

We next show another property of the saddle points $(x^{k,m}, \xi^{k,m})$, namely, that for each k, we have

$$\lim_{m \to \infty} f(x^{k,m}) = f(\overline{x}^k) = f^* + \gamma^k. \tag{44}$$

For a fixed k and any sequence of integers m that tends to ∞, consider the corresponding sequence $\{x^{k,m}\}$. From (41) and (43), we see that $\{x^{k,m}\}$ belongs to the set $\{x \in X^k \mid f(x) \le f(\overline{x}^k)\}$, which is compact, since f is closed. Hence, $\{x^{k,m}\}$ has a cluster point, denoted by $\hat{x}^k$, which belongs to $\{x \in X^k \mid f(x) \le f(\overline{x}^k)\}$. By passing to a subsequence if necessary, we can assume without loss of generality that $\{x^{k,m}\}$ converges to $\hat{x}^k$. We claim that $\hat{x}^k$ is feasible for problem (39), i.e., $\hat{x}^k \in X^k$ and $g(\hat{x}^k) \le 0$. Indeed, the sequence $\{f(x^{k,m})\}$ is bounded from below by $\inf_{x \in X^k} f(x)$, which is finite by Weierstrass's theorem since f is closed and coercive when restricted to X^k. Also, for each k, $L_{k,m}(x^{k,m}, \xi^{k,m})$ is bounded from above by $f(\overline{x}^k)$ (cf. (41)), so (43) implies that

$$\limsup_{m \to \infty} g_j(x^{k,m}) \le 0 \qquad \forall\ j = 1, \ldots, r.$$

Therefore, by using the closedness of g_j, we obtain $g(\hat{x}^k) \le 0$, implying that $\hat{x}^k$ is a feasible solution of problem (39). Thus, $f(\hat{x}^k) \ge f(\overline{x}^k)$. Using (41) and (43) together with the closedness of f, we also have

$$f(\hat{x}^k) \le \liminf_{m \to \infty} f(x^{k,m}) \le \limsup_{m \to \infty} f(x^{k,m}) \le f(\overline{x}^k),$$

thereby showing (44).

The next step in the proof is given in the following lemma.

LEMMA 11. *For all sufficiently large k, and for all scalars $m \leq 1/\sqrt{\gamma^k}$, we have*

$$f(x^{k,m}) \leq f^* - \frac{\gamma^k}{2}. \tag{45}$$

Furthermore, there exists a scalar $m_k \geq 1/\sqrt{\gamma^k}$ such that

$$f(x^{k,m_k}) = f^* - \frac{\gamma^k}{2}. \tag{46}$$

Proof. Let $\gamma = f^* - f(\overline{x})$, where $\overline{x}$ was defined earlier as the vector in X such that $f(\overline{x}) < f^*$. For sufficiently large k, we have $\overline{x} \in X^k$ and $\gamma^k < \gamma$. Consider the vector

$$z^k = \left(1 - \frac{2\gamma^k}{\gamma^k + \gamma}\right)\overline{x}^k + \frac{2\gamma^k}{\gamma^k + \gamma}\overline{x},$$

which belongs to X^k for sufficiently large k (by the convexity of X^k and the fact that $2\gamma^k/(\gamma^k + \gamma) < 1$). By the convexity of f, we have

$$\begin{aligned} f(z^k) &\leq \left(1 - \frac{2\gamma^k}{\gamma^k + \gamma}\right) f(\overline{x}^k) + \frac{2\gamma^k}{\gamma^k + \gamma} f(\overline{x}) \\ &= \left(1 - \frac{2\gamma^k}{\gamma^k + \gamma}\right)(f^* + \gamma^k) + \frac{2\gamma^k}{\gamma^k + \gamma}(f^* - \gamma) \\ &= f^* - \gamma^k. \end{aligned} \tag{47}$$

Similarly, by the convexity of g_j, we have

$$g_j(z^k) \leq \left(1 - \frac{2\gamma^k}{\gamma^k + \gamma}\right) g_j(\overline{x}^k) + \frac{2\gamma^k}{\gamma^k + \gamma} g_j(\overline{x}) \leq \frac{2\gamma^k}{\gamma^k + \gamma} g_j(\overline{x}). \tag{48}$$

Using (43), we obtain

$$\begin{aligned} f(x^{k,m}) &\leq L_{k,m}(x^{k,m}, \xi^{k,m}) \\ &= \inf_{x \in X^k} \sup_{\xi \geq 0} L_{k,m}(x, \xi) \\ &= \inf_{x \in X^k} \left\{ f(x) + \frac{(\gamma^k)^2}{4k^2} \|x - \bar{x}^k\|^2 + \frac{m}{2} \|g^+(x)\|^2 \right\} \\ &\leq f(x) + (\beta\gamma^k)^2 + \frac{m}{2}\|g^+(x)\|^2 \qquad \forall\, x \in X^k, \end{aligned}$$

where in the last inequality we also use the definition of X^k so that $\|x - \bar{x}^k\| \leq 2\beta k$ for all $x \in X^k$. Substituting $x = z^k$ in the preceding relation, and using (47) and (48), we see that for large k,

$$f(x^{k,m}) \leq f^* - \gamma^k + (\beta\gamma^k)^2 + \frac{2m(\gamma^k)^2}{(\gamma^k + \gamma)^2}\|g^+(\overline{x})\|^2.$$

Since $\gamma^k \to 0$, this implies that for sufficiently large k and for all scalars $m \leq 1/\sqrt{\gamma^k}$, we have

$$f(x^{k,m}) \leq f^* - \frac{\gamma^k}{2},$$

i.e., (45) holds.

We next show that there exists a scalar $m_k \geq 1/\sqrt{\gamma^k}$ such that (46) holds. In the process, we show that, for fixed k, $L_{k,m}(x^{k,m}, \xi^{k,m})$ changes continuously with m, i.e., for all $\overline{m} > 0$, we have $L_{k,m}(x^{k,m}, \xi^{k,m}) \to L_{k,\overline{m}}(x^{k,\overline{m}}, \xi^{k,\overline{m}})$ as $m \to \overline{m}$. (By this we mean that, for every sequence $\{m^t\}$ that converges to $\overline{m}$, the corresponding sequence $L_{k,m^t}(x^{k,m^t}, \xi^{k,m^t})$ converges to $L_{k,\overline{m}}(x^{k,\overline{m}}, \xi^{k,\overline{m}})$.) Denote

$$f^k(x) = f(x) + \frac{(\gamma^k)^2}{4k^2}\|x - \overline{x}^k\|^2.$$

From (43), we have

$$L_{k,m}(x^{k,m}, \xi^{k,m}) = \bar{f}(x^{k,m}) + \frac{m}{2}\|g^+(x^{k,m})\|^2 = \inf_{x \in X^k}\left\{\bar{f}(x) + \frac{m}{2}\|g^+(x)\|^2\right\},$$

so that for all $m \geq \overline{m}$, we obtain

$$\begin{aligned} L_{k,\overline{m}}(x^{k,\overline{m}}, \xi^{k,\overline{m}}) &= f^k(x^{k,\overline{m}}) + \frac{\overline{m}}{2}\|g^+(x^{k,\overline{m}})\|^2 \\ &\leq f^k(x^{k,m}) + \frac{\overline{m}}{2}\|g^+(x^{k,m})\|^2 \\ &\leq f^k(x^{k,m}) + \frac{m}{2}\|g^+(x^{k,m})\|^2 \\ &\leq f^k(x^{k,\overline{m}}) + \frac{m}{2}\|g^+(x^{k,\overline{m}})\|^2. \end{aligned}$$

It follows that $L_{k,m}(x^{k,m}, \xi^{k,m}) \to L_{k,\overline{m}}(x^{k,\overline{m}}, \xi^{k,\overline{m}})$ as $m \downarrow \overline{m}$. Similarly, we have for all $m \leq \overline{m}$

$$\begin{aligned} f^k(x^{k,\overline{m}}) + \frac{m}{2}\|g^+(x^{k,\overline{m}})\|^2 &\leq f^k(x^{k,\overline{m}}) + \frac{\overline{m}}{2}\|g^+(x^{k,\overline{m}})\|^2 \\ &\leq f^k(x^{k,m}) + \frac{\overline{m}}{2}\|g^+(x^{k,m})\|^2 \\ &= f^k(x^{k,m}) + \frac{m}{2}\|g^+(x^{k,m})\|^2 + \frac{\overline{m} - m}{2}\|g^+(x^{k,m})\|^2 \\ &\leq f^k(x^{k,\overline{m}}) + \frac{m}{2}\|g^+(x^{k,\overline{m}})\|^2 + \frac{\overline{m} - m}{2}\|g^+(x^{k,m})\|^2. \end{aligned}$$

For each k, $f(x^{k,m})$ is bounded from below by $\inf_{x \in X^k} f(x)$, which is finite by Weierstrass's theorem since f is closed and coercive when restricted to X^k. Since, by (41) and (43),

$$f(x^{k,m}) + \frac{m}{2}\|g^+(x^{k,m})\|^2 \leq f(\overline{x}^k),$$

we see that $m\|g^+(x^{k,m})\|^2$ is bounded from above as $m \uparrow \overline{m} > 0$, so that $(\overline{m} - m)\|g^+(x^{k,m})\|^2 \to 0$. Therefore, we have from the preceding relation that $L_{k,m}(x^{k,m}, \xi^{k,m}) \to L_{k,\overline{m}}(x^{k,\overline{m}}, \xi^{k,\overline{m}})$ as $m \uparrow \overline{m}$, which shows that $L_{k,m}(x^{k,m}, \xi^{k,m})$ changes continuously with m.

Next, we show that, for fixed k, $x^{k,m} \to x^{k,\overline{m}}$ as $m \to \overline{m}$. Since, for each k, $x^{k,m}$ belongs to the compact set $\{x \in X^k \mid f(x) \leq f(\overline{x}^k)\}$, it has a cluster point as $m \to \overline{m}$. Let $\hat{x}$ be a cluster point of $x^{k,m}$. Using the continuity of $L_{k,m}(x^{k,m}, \xi^{k,m})$

in m, and the closedness of f^k and g_j, we obtain

$$\begin{aligned} L_{k,\overline{m}}(x^{k,\overline{m}}, \xi^{k,\overline{m}}) &= \lim_{m\to\bar{m}} L_{k,m}(x^{k,m}, \xi^{k,m}) \\ &= \lim_{m\to\overline{m}} \left\{ f^k(x^{k,m}) + \frac{m}{2}\|g^+(x^{k,m})\|^2 \right\} \\ &\geq f^k(\hat{x}) + \frac{\overline{m}}{2}\|g^+(\hat{x})\|^2 \\ &\geq \inf_{x\in X^k} \left\{ f^k(x) + \frac{\overline{m}}{2}\|g^+(x)\|^2 \right\} \\ &= L_{k,\overline{m}}(x^{k,\overline{m}}, \xi^{k,\overline{m}}). \end{aligned}$$

This shows that $\hat{x}$ attains the infimum of $f^k(x) + \frac{\overline{m}}{2}\|g^+(x)\|^2$ over $x \in X^k$. Since this function is strictly convex, it has a unique optimal solution, showing that $\hat{x} = x^{k,\overline{m}}$.

Finally, we show that $f(x^{k,m}) \to f(x^{k,\overline{m}})$ as $m \to \overline{m}$. Since f is lower semicontinuous at $x^{k,\overline{m}}$, we have $f(x^{k,\overline{m}}) \leq \liminf_{m\to\overline{m}} f(x^{k,m})$. Thus it suffices to show that $f(x^{k,\overline{m}}) \geq \limsup_{m\to\overline{m}} f(x^{k,m})$. Assume that $f(x^{k,\overline{m}}) < \limsup_{m\to\overline{m}} f(x^{k,m})$. Using the continuity of $L_{k,m}(x^{k,m}, \xi^{k,m})$ in m and the fact that $x^{k,m} \to x^{k,\overline{m}}$ as $m \to \overline{m}$, we have

$$\begin{aligned} f^k(x^{k,\overline{m}}) + \liminf_{m\to\overline{m}} \|g^+(x^{k,m})\|^2 &< \limsup_{m\to\overline{m}} L_{k,m}(x^{k,m}, \xi^{k,m}) \\ &= L_{k,\overline{m}}(x^{k,\overline{m}}, \xi^{k,\overline{m}}) \\ &= f^k(x^{k,\overline{m}}) + \|g^+(x^{k,\overline{m}})\|^2. \end{aligned}$$

This contradicts the lower semicontinuity of g_j, so that $f(x^{k,\overline{m}}) \geq \limsup_{m\to\overline{m}} f(x^{k,m})$. Thus $f(x^{k,m})$ is continuous in m.

From (44), (45), and the continuity of $f(x^{k,m})$ in m, we see that there exists some scalar $m_k \geq 1/\sqrt{\gamma^k}$ such that (46) holds. □

We are now ready to construct FJ-multipliers with the desired properties. By combining (46), (41), and (43) (for $m = m_k$), together with the facts that $f(\overline{x}^k) \to f^*$ and $\gamma^k \to 0$ as $k \to \infty$, we obtain

$$(49) \qquad \lim_{k\to\infty} \left(f(x^{k,m_k}) - f^* + \frac{(\gamma^k)^2}{4k^2}\|x^{k,m_k} - \overline{x}^k\|^2 + \xi^{k,m_k'} g(x^{k,m_k}) \right) = 0.$$

Denote

$$(50) \qquad \delta^k = \sqrt{1 + \|\xi^{k,m_k}\|^2}, \qquad \mu_0^k = \frac{1}{\delta^k}, \qquad \mu^k = \frac{\xi^{k,m_k}}{\delta^k}.$$

Since δ^k is bounded from below by 1, (49) yields

$$(51) \qquad \lim_{k\to\infty} \left(\mu_0^k f(x^{k,m_k}) - \mu_0^k f^* + \frac{(\gamma^k)^2}{4k^2\delta^k}\|x^{k,m_k} - \overline{x}^k\|^2 + \mu^{k'} g(x^{k,m_k}) \right) = 0.$$

Substituting $m = m_k$ in the first relation of (40) and dividing by δ^k, we obtain

$$\begin{aligned} &\mu_0^k f(x^{k,m_k}) + \frac{(\gamma^k)^2}{4k^2\delta^k}\|x^{k,m_k} - \overline{x}^k\|^2 + \mu^{k'} g(x^{k,m_k}) \\ &\qquad \leq \mu_0^k f(x) + \mu^{k'} g(x) + \frac{(\beta\gamma^k)^2}{\delta^k} \qquad \forall\, x \in X^k, \end{aligned}$$

where we also use the fact that $\|x-\bar{x}^k\| \leq 2\beta k$ for all $x \in X^k$ (cf. the definition of X^k). Since the sequence $\{(\mu_0^k, \mu^k)\}$ is bounded, it has a cluster point, denoted by (μ_0^*, μ^*), which satisfies conditions (ii) and (iii) in the definition of an FJ-multiplier. Without loss of generality, we will assume that the entire sequence $\{(\mu_0^k, \mu^k)\}$ converges to (μ_0^*, μ^*). For any $x \in X$, we have $x \in X^k$ for all k sufficiently large. Taking the limit as $k \to \infty$ in the preceding relation and using (51) and $\gamma^k \to 0$ yield

$$\mu_0^* f^* \leq \mu_0^* f(x) + \mu^{*\prime} g(x) \qquad \forall\, x \in X,$$

which implies that

$$\begin{aligned} \mu_0^* f^* &\leq \inf_{x \in X} \{\mu_0^* f(x) + \mu^{*\prime} g(x)\} \\ &\leq \inf_{x \in X,\, g(x) \leq 0} \{\mu_0^* f(x) + \mu^{*\prime} g(x)\} \\ &\leq \inf_{x \in X,\, g(x) \leq 0} \mu_0^* f(x) \\ &= \mu_0^* f^*. \end{aligned}$$

Thus we have

$$\mu_0^* f^* = \inf_{x \in X} \{\mu_0^* f(x) + \mu^{*\prime} g(x)\},$$

so that μ_0^*, μ^* satisfy condition (i) in the definition of an FJ-multiplier. Note that the existence of $\overline{x} \in X$ such that $f(\overline{x}) < f^*$, together with condition (i), implies that $\mu^* \neq 0$.

Finally, we establish condition (CV). Using (42) and (50) and the fact that $\mu^k \to \mu^*$, we obtain

$$\frac{g^+(x^{k,m_k})}{\|g^+(x^{k,m_k})\|} = \frac{\mu^{k,m_k}}{\|\mu^{k,m_k}\|} \to \frac{\mu^*}{\|\mu^*\|}.$$

We have from (46) and $\gamma^k \to 0$ that $f(x^{k,m_k}) \to f^*$. We also have from (41), (43) with $m = m_k$, and (46) that

$$\frac{m_k}{2}\|g^+(x^{k,m_k})\|^2 \leq f(\bar{x}^k) - f(x^{k,m_k}) = \frac{3}{2}\gamma^k,$$

where the equality uses (42) and (50). Since $\gamma^k \to 0$ and $m_k \geq 1/\sqrt{\gamma^k} \to \infty$, this yields $g^+(x^{k,m_k}) \to 0$. Moreover, combining the above inequality with (46) yields

$$\text{(52)} \qquad \frac{f^* - f(x^{k,m_k})}{\|g^+(x^{k,m_k})\|} = \frac{\gamma^k}{2\|g^+(x^{k,m_k})\|} \geq \frac{m_k\|g^+(x^{k,m_k})\|}{6} = \frac{\|\mu^{k,m_k}\|}{6\mu_0^{k,m_k}}.$$

If $\mu_0^* = 0$, then $\mu_0^{k,m_k} \to 0$, and so (52) together with $\|\mu^{k,m_k}\| \to \|\mu^*\| > 0$ yields

$$\frac{f^* - f(x^{k,m_k})}{\|g^+(x^{k,m_k})\|} \to \infty.$$

It follows that the sequence $\{x^{k,m_k}\}$ satisfies condition (CV) of the proposition. If $\mu_0^* \neq 0$, then μ^*/μ_0^* is a geometric multiplier and $f^* = q^*$, so that μ^*/μ_0^* is also a dual optimal solution. Thus the set of dual optimal solutions is nonempty and coincides with the set of geometric multipliers. Then, the vector $(1, \bar{\mu})$, where $\bar{\mu}$ is the dual optimal solution of minimum norm, is an FJ-multiplier and, by Proposition 5 and the fact that $f^* = q^*$, it satisfies condition (CV) and is the only dual optimal solution that satisfies this condition. This completes the proof. □

5. Dual Fritz John conditions when there is no optimal solution. The FJ-multipliers of Propositions 3, 8, and 10 define a hyperplane with normal (μ^*, μ_0^*) that supports the set of constraint-cost pairs

$$M = \big\{(u, w) \mid \text{ there exists } x \in X \text{ such that } \ g(x) \le u,\ f(x) \le w\big\}$$

at $(0, f^*)$. On the other hand, it is possible to construct a hyperplane that supports the set M at the point $(0, q^*)$, where q^* is the dual optimal value, while asserting the existence of a sequence that satisfies a condition analogous to condition (CV) of Proposition 10. This is the subject of the next proposition. Its proof uses Lemmas 1 and 4.

In analogy with an FJ-multiplier, we consider a scalar μ_0^* and a vector $\mu^* = (\mu_1^*, \dots, \mu_r^*)'$, satisfying the following conditions:

(i) $\mu_0^* q^* = \inf_{x\in X}\big\{\mu_0^* f(x) + \mu^{*\prime} g(x)\big\}$.

(ii) $\mu_j^* \ge 0$ for all $j = 0, 1, \dots, r$.

(iii) $\mu_0^*, \mu_1^*, \dots, \mu_r^*$ are not all equal to 0.

We call such a pair (μ_0^*, μ^*) a *dual FJ-multiplier.* If $\mu_0^* \neq 0$, then μ^*/μ_0^* is a dual optimal solution; otherwise $\mu_0^* = 0$ and $\mu^* \neq 0$.

PROPOSITION 12 (enhanced dual Fritz John conditions). *Consider the convex problem* (P) *under Assumption* 1 (*closedness*)*, and assume that $f^* < \infty$ and $-\infty < q^*$. Then there exists a dual FJ-multiplier (μ_0^*, μ^*) satisfying the following condition* (dCV)*. Moreover, if $\mu_0^* \neq 0$, then μ^*/μ_0^* must be the dual optimal solution of minimum norm.*

(dCV) *If $\mu^* \neq 0$, then there exists a sequence $\{x^k\} \subset X$ of infeasible points that satisfies*

$$f(x^k) \to q^*, \qquad g^+(x^k) \to 0, \tag{53}$$

$$\frac{q^* - f(x^k)}{\|g^+(x^k)\|} \to \begin{cases} \|\mu^*\|/\mu_0^* & \textit{if } \mu_0^* \neq 0, \\ \infty & \textit{if } \mu_0^* = 0, \end{cases} \tag{54}$$

$$\frac{g^+(x^k)}{\|g^+(x^k)\|} \to \frac{\mu^*}{\|\mu^*\|}. \tag{55}$$

Proof. Since by assumption we have $-\infty < q^*$ and $f^* < \infty$, it follows from the weak duality relation $q^* \le f^*$ that both q^* and f^* are finite. For $k = 1, 2, \dots$, consider the problem

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && x \in X,\ g_j(x) \le \frac{1}{k^4}, \quad j = 1, \dots, r. \end{aligned}$$

By Lemma 4, for each k, the optimal value of this problem is less than or equal to q^*. Then, for each k, there exists a vector $\tilde{x}^k \in X$ that satisfies

$$f(\tilde{x}^k) \le q^* + \frac{1}{k^2}, \qquad g_j(\tilde{x}^k) \le \frac{1}{k^4}, \quad j = 1, \dots, r.$$

Consider also the problem

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && g_j(x) \le \frac{1}{k^2}, \quad j = 1, \dots, r, \\ & && x \in \tilde{X}^k = X \cap \{x \mid \|x\| \le k\left(\max\nolimits_{1\le i\le k} \|\tilde{x}^i\| + 1\right)\}. \end{aligned} \tag{56}$$

Since f and g_j are closed and convex when restricted to X, they are closed, convex, and coercive when restricted to $\tilde{X}^k$. Hence, problem (56) has an optimal solution, which we denote by $\overline{x}^k$. Note that since $\tilde{x}^k$ belongs to the feasible solution set of this problem, we have

$$f(\overline{x}^k) \leq f(\tilde{x}^k) \leq q^* + \frac{1}{k^2}. \tag{57}$$

For each k, we consider the saddle function

$$L_k(x, \xi) = f(x) + \xi' g(x) - \frac{\|\xi\|^2}{2k}$$

and the set

$$X^k = \tilde{X}^k \cap \{x \mid g_j(x) \leq k,\ j = 1, \dots, r\}. \tag{58}$$

We note that $L_k(x, \xi)$, for fixed $\xi \geq 0$, is closed, convex, and coercive in x, when restricted to X^k, and negative definite quadratic in ξ for fixed x. Hence, using the saddle point theorem (e.g., [BNO03, Proposition 2.6.9]), we can assert that L_k has a saddle point over $x \in X^k$ and $\xi \geq 0$, denoted by (x^k, ξ^k).

Since L_k is quadratic in ξ, the supremum of $L_k(x^k, \xi)$ over $\xi \geq 0$ is attained at

$$\xi^k = k g^+(x^k). \tag{59}$$

Similarly, the infimum of $L_k(x, \xi^k)$ over $x \in X^k$ is attained at x^k, implying that

$$\begin{aligned}
f(x^k) + \xi^{k'} g(x^k) &= \inf_{x \in X^k} \{f(x) + \xi^{k'} g(x)\} \\
&= \inf_{x \in X^k} \{f(x) + k g^+(x^k)' g(x)\} \\
&\leq \inf_{x \in X^k,\ g_j(x) \leq \frac{1}{k^4},\, j=1,\dots,r,} \left\{ f(x) + k \sum_{j=1}^{r} g_j^+(x^k)' g_j(x) \right\} \\
&\leq \inf_{x \in X^k,\ g_j(x) \leq \frac{1}{k^4},\, j=1,\dots,r,} \left\{ f(x) + \frac{r}{k^2} \right\} \\
&= f(\overline{x}^k) + \frac{r}{k^2} \\
&\leq q^* + \frac{r+1}{k^2},
\end{aligned} \tag{60}$$

where the second inequality follows using the fact $g_j^+(x^k) \leq k$, $j = 1, \dots, r$ (cf. (58)), and the third inequality follows from (57).

Since q^* is finite, we may select a nonnegative sequence $\{\zeta^k\}$ such that

$$q(\zeta^k) \to q^*, \qquad \frac{\|\zeta^k\|^2}{2k} \to 0. \tag{61}$$

(For example, we can take ζ^k to be any maximizer of $q(\zeta)$ subject to $\zeta \geq 0$ and

$\|\zeta\| \leq k^{1/3}$.) Then, we have for all k

$$\begin{aligned} L_k(x^k,\xi^k) &= \sup_{\xi\geq 0}\inf_{x\in X^k} L_k(x,\xi) \\ &\geq \sup_{\xi\geq 0}\inf_{x\in X} L_k(x,\xi) \\ &= \sup_{\xi\geq 0}\left\{\inf_{x\in X}\big\{f(x)+\xi' g(x)\big\} - \frac{\|\xi\|^2}{2k}\right\} \\ &= \sup_{\xi\geq 0}\left\{q(\xi) - \frac{\|\xi\|^2}{2k}\right\} \\ &\geq q(\zeta^k) - \frac{\|\zeta^k\|^2}{2k}. \end{aligned} \tag{62}$$

Combining (62) and (60), we obtain

$$\begin{aligned} q(\zeta^k) - \frac{\|\zeta^k\|^2}{2k} &\leq L_k(x^k,\xi^k) \\ &= f(x^k) + \xi^{k'} g(x^k) - \frac{\|\xi^k\|^2}{2k} \\ &\leq f(x^k) + \xi^{k'} g(x^k) \\ &\leq q^* + \frac{r+1}{k^2}. \end{aligned} \tag{63}$$

Taking the limit in the preceding relation and using (61), we obtain

$$\lim_{k\to\infty}\big\{f(x^k) - q^* + \xi^{k'} g(x^k)\big\} = 0. \tag{64}$$

Denote

$$\delta^k = \sqrt{1+\|\xi^k\|^2}, \qquad \mu_0^k = \frac{1}{\delta^k}, \qquad \mu^k = \frac{\xi^k}{\delta^k}. \tag{65}$$

Since δ^k is bounded from below by 1, (64) yields

$$\lim_{k\to\infty}\{\mu_0^k\big(f(x^k) - q^*\big) + \mu^{k'} g(x^k)\} = 0. \tag{66}$$

Dividing both sides of the first relation in (60) by δ^k, we get

$$\mu_0^k f(x^k) + \mu^{k'} g(x^k) \leq \mu_0^k f(x) + \mu^{k'} g(x) \qquad \forall\, x \in X^k.$$

Since the sequence $\big\{(\mu_0^k,\mu^k)\big\}$ is bounded, it has a cluster point (μ_0^*,μ^*). This cluster point satisfies conditions (ii) and (iii) of the proposition. Without loss of generality, we assume that the entire sequence converges. For any $x \in X$, we have $x \in X^k$ for all k sufficiently large. Taking the limit as $k \to \infty$ in the preceding relation and using (66) yield

$$\mu_0^* q^* \leq \mu_0^* f(x) + \mu^{*'} g_j(x) \qquad \forall\, x \in X.$$

We consider separately the two cases, $\mu_0^* > 0$ and $\mu_0^* = 0$, in the above relation to show that (μ_0^*,μ^*) satisfy condition (i) of the proposition. Indeed, if $\mu_0^* > 0$, by dividing with μ_0^*, we have

$$q^* \leq \inf_{x\in X}\left\{f(x) + \frac{\mu^{*'}}{\mu_0^*} g(x)\right\} = q\left(\frac{\mu^*}{\mu_0^*}\right) \leq q^*.$$

Similarly, if $\mu_0^* = 0$, it can be seen that

$$0 = \inf_{x \in X} \mu^{*'} g(x)$$

(since $f^* < \infty$, so that there exists an $x \in X$ such that $g(x) \le 0$ and $\mu^{*\prime} g(x) \le 0$). Hence, in both cases, we have

$$\mu_0^* q^* = \inf_{x \in X} \{\mu_0^* f(x) + \mu^{*'} g(x)\},$$

thus showing condition (i) in the definition of a dual FJ-multiplier.

If $\mu^* = 0$, then $\mu_0^* \neq 0$, (dCV) is automatically satisfied, and $\mu^*/\mu_0^* = 0$ has minimum norm. Assume now that $\mu^* \neq 0$. Using (59), (65), and the fact that $\mu^k \to \mu^*$, we obtain

$$\frac{g^+(x^k)}{\|g^+(x^k)\|} = \frac{\mu^k}{\|\mu^k\|} \to \frac{\mu^*}{\|\mu^*\|}.$$

This proves (55). Also, we have from (63) that

$$k\bigl(f(x^k) - q^*\bigr) + \xi^{k'} k g(x^k) \le \frac{r+1}{k} \qquad \forall\, k = 1, 2, \ldots .$$

Using (59), this yields

$$k\bigl(f(x^k) - q^*\bigr) + \|\xi^k\|^2 \le \frac{r+1}{k}.$$

Dividing both sides by $\|\xi^k\| = k\|g^+(x^k)\|$ and using (65) yield

$$\text{(67)} \qquad \frac{q^* - f(x^k)}{\|g^+(x^k)\|} \ge \|\xi^k\| - \frac{r+1}{k\|\xi^k\|} = \frac{\|\mu^k\|}{\mu_0^k} - \frac{r+1}{k\|\mu^k\|/\mu_0^k}.$$

If $\mu_0^* = 0$, then $\mu_0^k \to 0$, and so (67) together with $\|\mu^k\| \to \|\mu^*\| > 0$ yields

$$\frac{q^* - f(x^k)}{\|g^+(x^k)\|} \to \infty.$$

If $\mu_0^* \neq 0$, then (67) together with $\mu_0^k \to \mu_0^*$ and $\|\mu^k\| \to \|\mu^*\|$ yields

$$\liminf_{k \to \infty} \frac{q^* - f(x^k)}{\|g^+(x^k)\|} \ge \frac{\|\mu^*\|}{\mu_0^*}.$$

Since μ^*/μ_0^* is a dual optimal solution, Lemma 1 shows that in fact μ^*/μ_0^* is of minimum norm and the inequality holds with equality.

We finally show that $f(x^k) \to q^*$ and $g^+(x^k) \to 0$. By (63) and (61), we have

$$\text{(68)} \qquad \lim_{k \to \infty} \frac{\|\xi^k\|^2}{2k} = 0.$$

By (59), we have

$$\xi^{k\prime} g(x^k) = \frac{1}{k}\|\xi^k\|^2,$$

and so using also (63) and (61), we obtain

$$\lim_{k\to\infty} f(x^k) + \frac{\|\xi^k\|^2}{2k} = q^*,$$

which together with (68) shows that $f(x^k) \to q^*$. Moreover, (68) and (59) imply that

$$\lim_{k\to\infty} k\|g^+(x^k)\|^2 = 0,$$

showing that $g^+(x^k) \to 0$. Therefore, the sequence $\{x^k\}$ satisfies condition (dCV) of the proposition, completing the proof. □

Note that the proof of Proposition 12 is similar to the proof of Proposition 2. The idea is to generate saddle points of the function

$$L_k(x,\xi) = f(x) + \xi' g(x) - \frac{\|\xi\|^2}{2k}$$

over $x \in X^k$ (cf. (58)) and $\xi \geq 0$. It can be shown that

$$L_k(x^k,\xi^k) = \inf_{u\in\Re^r} \left\{ p^k(u) + \frac{k}{2}\|u^+\|^2 \right\},$$

where $p^k(u)$ is the optimal value of the problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g(x) \leq u,\ x \in X^k \end{array}$$

(see the discussion following the proof of Proposition 2). For each k, the value $L_k(x^k,\xi^k)$ can be visualized geometrically as in Figure 1. However, here the rate at which X^k approaches X is chosen high enough so that $L_k(x^k,\xi^k)$ converges to q^* as $k \to \infty$ (cf. (63)), and not to f^*, as in the proof of Propositions 2 or 10.

As a final remark, it appears that the closedness assumption in Proposition 12 can be relaxed analogously as in Proposition 7 by using Lemmas 1 and 6.

REFERENCES

[BNO03] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar, *Convex Analysis and Optimization*, Athena Scientific, Belmont, MA, 2003.

[BeO02] D. P. Bertsekas and A. E. Ozdaglar, *Pseudonormality and a Lagrange multiplier theory for constrained optimization*, J. Optim. Theory Appl., 114 (2002), pp. 287–343.

[Ber99] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, MA, 1999.

[Hes75] M. R. Hestenes, *Optimization Theory: The Finite Dimensional Case*, Wiley, New York, 1975.

[Joh48] F. John, *Extremum problems with inequalities as subsidiary conditions*, in Studies and Essays: Courant Anniversary Volume, K. O. Friedrichs, O. E. Neugebauer, and J. J. Stoker, eds., Wiley-Interscience, New York, 1948, pp. 187–204.

[Kar39] W. Karush, *Minima of functions of several variables with inequalities*, M.Sc. Thesis, Department of Mathematics, University of Chicago, Chicago, IL, 1939.

[Kuh91] H. W. Kuhn, *Nonlinear programming: A historical note*, in History of Mathematical Programming, J. K. Lenstra, A. H. G. Rinnooy Kan, and A. Schrijver, eds., North–Holland, Amsterdam, 1991, pp. 77–96.

[Lue79] D. G. Luenberger, *Optimization by Vector Space Methods*, Wiley, New York, 1979.

[Roc70] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[Roc93] R. T. Rockafellar, *Lagrange multipliers and optimality*, SIAM Rev., 35 (1993), pp. 183–238.

[Sla50] M. Slater, *Lagrange Multipliers Revisited: A Contribution to Nonlinear Programming*, Cowles Commission Discussion Paper, Math. 403, November, 1950.

SIAM J. Optim.
Vol. 16, No. 3, pp. 798–825

NONLINEAR INEXACT UZAWA ALGORITHMS FOR LINEAR AND NONLINEAR SADDLE-POINT PROBLEMS*

QIYA HU† AND JUN ZOU‡

Abstract. This paper proposes some nonlinear Uzawa methods for solving linear and nonlinear saddle-point problems. A nonlinear inexact Uzawa algorithm is first introduced for linear saddle-point problems. Two different PCG techniques are allowed in the inner and outer iterations of the algorithm. This algorithm is then extended for a class of nonlinear saddle-point problems arising from some convex optimization problems with linear constraints. For this extension, some PCG method used in the inner iteration needs to be carefully constructed so that it converges in a certain energy norm instead of the usual l^2-norm. It is shown that the new algorithm converges under some practical conditions and there is no need for any a priori estimates on the minimal and maximal eigenvalues of the two local preconditioned systems involved. The two new methods perform more efficiently than the existing methods in the cases where no good preconditioners are available for the Schur complements.

Key words. linear and nonlinear saddle-point problems, inexact Uzawa algorithm, preconditioning

AMS subject classifications. 65F10, 65N22

DOI. 10.1137/S1052623403428683

1. Introduction.

This paper is mainly concerned with the construction of efficient nonlinear inexact Uzawa algorithms for solving the nonlinear saddle-point system

$$\left\{ \begin{array}{rcl} F(x) + B\,y &=& f, \\ B^t\,x &=& g, \end{array} \right. \tag{1.1}$$

where B is an $n \times m$ matrix with full column rank ($m \leq n$), and $F : R^n \to R^n$ is a nonlinear vector-valued function, not necessarily differentiable.

The nonlinear saddle-point system of form (1.1) arises frequently in augmented Lagrangian formulations of inverse problems [16], electromagnetic Maxwell equations [13], [15], and nonlinear optimizations, for example, of the form (cf. [12], [22], [41])

$$\left\{ \begin{array}{l} \min\limits_{x \in R^n} \{J(x) - (f, x)\} \\ \text{s.t.} \quad B^t x = g, \end{array} \right. \tag{1.2}$$

where $J(x)$ is the function satisfying $\nabla J(x) = F(x)$.

When $F(x)$ is linear, for example, $F(x) = Ax$ with A being an $n \times n$ symmetric positive definite matrix, system (1.1) reduces to the well-known (linear) saddle-point problem

$$\left\{ \begin{array}{rcl} A\,x + B\,y &=& f, \\ B^t\,x &=& g. \end{array} \right. \tag{1.3}$$

*Received by the editors May 24, 2003; accepted for publication (in revised form) June 28, 2005; published electronically January 12, 2006.

http://www.siam.org/journals/siopt/16-3/42868.html

†Institute of Computational Mathematics and Scientific Engineering Computing, Academy of Mathematics and System Sciences, The Chinese Academy of Sciences, Beijing 100080, China (hqy@lsec.cc.ac.cn). The work of this author was supported by National Natural Science Foundation of China grant 10371129, National Basic Research Program of China grant 2005CB321702, and the Key Project of Natural Science Foundation of China G10531080.

‡Department of Mathematics, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong (zou@math.cuhk.edu.hk). The work of this author was substantially supported by Hong Kong RGC grants (Projects 403403 and 404105).

As we shall see, the Schur complement matrix

$$K = B^t A^{-1} B \tag{1.4}$$

associated with system (1.3), and its preconditioners, play an essential role in solving the saddle-point problem.

During the past decade, there has been a growing interest in preconditioned iterative methods for solving the indefinite saddle-point system of equations like (1.3); see [4], [10], [11], [18], [20], [37], [39]. The standard Uzawa-type method [2], [3] and the minimal residual (MINRES) method are the most popular iterative methods for solving (1.3). Let $\hat{A}$ and $\hat{K}$ be two positive definite matrices, which are assumed to be the preconditioners for the matrices A and K, respectively. Then it is known that the convergence rates of both the standard Uzawa-type method and the MINRES method depend on the condition numbers $\mathrm{cond}(\hat{A}^{-1}A)$ and $\mathrm{cond}(\hat{K}^{-1}K)$ of the two local preconditioned systems, and they are much less efficient when one of the two condition numbers is relatively larger than the other. To effectively deal with the case where $\mathrm{cond}(\hat{A}^{-1}A)$ is relatively larger than $\mathrm{con}(\hat{K}^{-1}K)$, a nonlinear inexact Uzawa algorithm was proposed in [10], in which the inner iteration uses a (nonlinear) iterative method to replace the action of A^{-1}. Recently we have introduced two new algorithms (Algorithms 3.1 and 4.1 in [29]) to improve the existing algorithms and convergence results. These two algorithms were, respectively, designed to effectively treat two different cases: (a) $\mathrm{cond}(\hat{A}^{-1}A) \gg \mathrm{cond}(\hat{K}^{-1}K)$; (b) $\mathrm{cond}(\hat{K}^{-1}K) \gg \mathrm{cond}(\hat{A}^{-1}A)$. It was shown that Algorithm 3.1 in [29] is efficient for case (a), but Algorithm 4.1 there may not always be efficient for case (b), as there is one condition (see (4.2) in [29]) which may not be easily guaranteed in applications.

The purpose of this paper is twofold. To better understand the difference between linear and nonlinear saddle-point problems, we first propose some improved version of Algorithm 4.1 in [29] for solving linear system (1.3). As we shall see, the new algorithm is always convergent without any assumptions on the spectra of the preconditioned systems $\widehat{K}^{-1}K$ and $\widehat{A}^{-1}A$. This seems to be an important advantage of the new algorithm over the existing iterative methods for saddle-point problems. Then we extend this improved algorithm to effectively solve nonlinear saddle-point problems of form (1.1), which is assumed to arise from some convex minimization problems with linear constraints.

To our knowledge, there have been very few investigations into the rate of convergence for preconditioned iterative methods for nonlinear saddle-point systems. Al-Baali and Fletcher studied in [1] rates of convergence of preconditioned nonlinear conjugate gradient (CG) methods for unconstrained optimizations, when the preconditioning matrix is taken to be the exact Hessian matrix at each iteration. In [14], Chen gave a deep analysis on rates of convergence of inexact Uzawa methods for nonlinear saddle-point systems, when the exact Hessian matrix at each iteration is used in the preconditioner for the Schur complement. Most existing analyses are carried out in the standard l^2-norm, which may not be so natural and accurate for many problems from applications.

In this paper, we shall make an effort to study convergence rates of inexact Uzawa algorithms in the *energy-norm* when the exact Hessian matrix in the Schur complement is replaced by some *inexact preconditioner* at each iteration. Due to the nonlinearity of the saddle-point system, the conditioning of the preconditioned Schur complement may become much worse than that of the preconditioned Hessian matrix at each iteration. In this case, a special nonlinear preconditioning process is first introduced to improve the conditioning of the preconditioned Schur complement. Further, a preconditioned nonlinear CG method is introduced in the inner iteration for the nonlinear system related to $F(\cdot)$ at each iteration. To ensure the convergence of the global inexact Uzawa algorithm, the preconditioned nonlinear CG method needs to be carefully constructed so that it converges in a certain energy norm instead of the usual l^2-norm. As we shall see, the new algorithm is always convergent without any assumptions on the spectra of the preconditioned Schur complement systems and Hessian matrices. More important, all the tolerance parameters involved in the inner iterations of the inexact Uzawa algorithm can be taken to be some fixed constants independent of the iterations, for example, 1/2 or 1/3. This appears to be an important advantage of the new algorithm over the existing ones.

Although quite different from what we are doing here, we mention another interesting and popular approach widely used in the optimization community. This approach intends to solve a nonlinear equality-constrained minimization problem by sequential quadratic programming in which successively quadratic subproblems are solved. Each quadratic subproblem amounts to solving a linear Karush–Kuhn–Tucker (KKT) saddle-point system. Global preconditioners for the resulting KKT coefficient matrices have been widely studied, and maintain the block structures of the original KKT matrices; see [5], [6], [7], [23], [32], and the references therein.

The rest of this paper is organized as follows. First, in section 2 we propose an improved variant of Algorithm 4.1 studied in [29] for linear saddle-point problems. The algorithm is then extended for nonlinear saddle-point problems in section 3, and its rate of convergence is also analyzed under some weak smoothness assumptions on the nonlinear functions $F(\cdot)$. Finally, in section 4 we apply two new algorithms proposed in sections 2 and 3 to solve an algebraic system of nonlinear saddle-point problem and a linear saddle-point problem arising from the domain decomposition method with Lagrange multiplier.

2. Nonlinear inexact Uzawa algorithms for linear saddle-point problems. In this section, we shall propose an improved variant of Algorithm 4.1 from [29] for solving the linear saddle-point problem (1.3) and study its convergence. This improved algorithm will be extended in section 3 to solve the nonlinear saddle-point system (1.1). To do so, we need to introduce some notation. R^l will mean the usual l-dimensional Euclidean space. For any $l \times l$ positive definite matrix G, $\|x\|_G$ will represent the G-induced norm, namely $\|x\|_G = (Gx,\ x)^{1/2}$ for all $x \in R^l$. To describe the nonlinear inexact Uzawa algorithm, we introduce a nonlinear mapping $\Psi_A : R^n \to R^n$ such that for any given $\xi \in R^n$, $\Psi_A(\xi)$ is an "approximation" to the solution φ of the linear system

$$A\varphi = \xi. \tag{2.1}$$

The following assumption was often made on the accuracy of the approximation (e.g., see (4.2) in [10]):

$$\|\Psi_A(\xi) - A^{-1}\xi\|_A \le \delta\, \|A^{-1}\xi\|_A \quad \forall \xi \in R^n \tag{2.2}$$

for some $\delta \in (0, 1)$. Assumption (2.2) is natural and can be satisfied, for example, by the approximate inverse generated by the preconditioned conjugate gradient (PCG) iteration or by one sweep of a multigrid method with conjugate gradient smoothing [10].

We first recall an algorithm from [29] for solving the linear saddle-point problem (1.3).

Algorithm 2.1 (nonlinear inexact Uzawa-steepest descent). *Given* $\{x_0, y_0\} \in R^n \times R^m$*, the sequence of pairs* $\{x_i, y_i\} \in R^n \times R^m$ *is defined for* $i = 1, 2, \ldots,$ *by the following.*

Step 1. *Compute* $f_i = f - (Ax_i + By_i)$ *and* $\Psi_A(f_i)$*; update*

$$x_{i+1} = x_i + \Psi_A(f_i). \tag{2.3}$$

Step 2. *Compute* $g_i = B^t x_{i+1} - g$ *and* $d_i = \hat{K}^{-1} g_i$*. Then compute the relaxation parameter*

$$\tau_i = \begin{cases} \frac{(g_i, d_i)}{(\Psi_A(Bd_i), Bd_i)} & \text{for} \quad g_i \neq 0; \\ 1 & \text{for} \quad g_i = 0. \end{cases} \tag{2.4}$$

Update

$$y_{i+1} = y_i + \frac{1}{2}\,\tau_i\, d_i. \tag{2.5}$$

To study the convergence of Algorithm 2.1, we assume Ψ_A satisfies

$$\|A^{-1} f_i - \Psi_A(f_i)\|_A \le \delta_f\, \|A^{-1} f_i\|_A, \tag{2.6}$$

$$\|A^{-1} Bd_i - \Psi_A(Bd_i)\|_A \le \delta_d\, \|A^{-1} Bd_i\|_A \tag{2.7}$$

for two positive constants $\delta_f < 1$ and $\delta_d < 1$. We remark that one can simply take δ_f and δ_d to be the constant δ in (2.2). But the introduction of these two different constants enables us to see how the rate of convergence depends more explicitly on the accuracies of the nonlinear inner iterations in (2.3) and (2.5).

To measure the convergence rate of Algorithm 2.1 more accurately, an appropriate norm is very crucial. For each element v from the product space $R^n \times R^m$, we will write it as $v = \{v_1, v_2\}$, where $v_1 \in R^n$ and $v_2 \in R^m$. Then, as we did in [28], [29], we shall use the norm

$$|||v||| = (\|v_1\|^2_{A^{-1}} + \|v_2\|^2_K)^{\frac{1}{2}} \quad \forall\, v = \{v_1, v_2\} \in R^n \times R^m. \tag{2.8}$$

Finally, we introduce three error vectors $e_i^x \in R^n$, $e_i^y \in R^m$, and $E_i \in R^n \times R^m$:

$$e_i^x = x - x_i, \quad e_i^y = y - y_i, \quad E_i = \{\sqrt{\delta} f_i,\, e_i^y\}, \quad i = 0, 1, 2, \ldots,$$

and two parameters

$$\hat{\kappa} = \text{cond}(\hat{K}^{-1} K), \quad \hat{\beta} = \sqrt{1 - \frac{4\hat{\kappa}(1 - 2\delta_d)}{(1 + \hat{\kappa})^2 (1 - \delta_d)^2}}; \tag{2.9}$$

then we have the following estimates on the rate of convergence of Algorithm 2.1 in [29].

Lemma 2.1. *Assume that* (2.6) *and* (2.7) *are satisfied with the parameters* $\delta_f < \frac{1}{3}$ *and* $\delta_d < \frac{1}{2}$*; then Algorithm* 2.1 *converges. Moreover, the following estimate holds:*

$$|||E_{i+1}||| \le \hat{\rho}\, |||E_i|||, \quad i = 0, 1, 2, \ldots. \tag{2.10}$$

Also, (2.10) *implies for* $i = 1, 2, 3, \ldots$ *that*

$$\|e_i^x\|_A \le (\sqrt{1 + 4\delta_f} + \hat{\rho})\hat{\rho}^{i-1} |||E_0|||, \quad \|e_i^y\|_K \le \hat{\rho}^i\, |||E_0|||, \tag{2.11}$$

where the rate of convergence $\hat{\rho} < 1$ *and can be estimated by*

$$\hat{\rho} = \begin{cases} \sqrt{\delta_f + \delta_f^2} + \delta_f & \text{for} \quad 0 < \frac{1+\hat{\beta}}{2} \leq \frac{4\delta_f}{1+\delta_f}; \\ 1 - \frac{1}{4}(1-\hat{\beta})(1+\delta_f) & \text{for} \quad \frac{4\delta_f}{1+\delta_f} < \frac{1+\hat{\beta}}{2} < 1. \end{cases} \tag{2.12}$$

Remark 2.1. Algorithm 2.1 converges when a general preconditioner $\hat{K}$ is used for the Schur complement system $K = B^t A^{-1} B$ and a general nonlinear iteration Ψ_A is used for solving $A\varphi = \xi$ involved in the inner iteration. However, the steepest descent method converges with a reasonable rate only when a good preconditioner is available for the Schur complement system, namely, $\hat{\kappa} = \text{cond}(\hat{K}^{-1}K)$ is not large. This is the case when the saddle-point problem arises, for example, from the Stokes problem [39]. Without such a good preconditioner the method may converge with a very slow rate. Particularly, Algorithm 2.1 may be much less effective when $\text{cond}(\hat{K}^{-1}K) \gg \text{cond}(\hat{A}^{-1}A)$.

Another algorithm (Algorithm 4.1) was proposed in [29] that combines the nonlinear inexact Uzawa algorithm with the CG method, in an effort to accelerate the nonlinear inexact Uzawa algorithm when $\text{cond}(\hat{K}^{-1}K) \gg \text{cond}(\hat{A}^{-1}A)$. This is the case when the saddle-point problems arise from the domain decomposition method with Lagrange multiplier [27], [34], or from the Lagrange multiplier formulations for optimization problems [25] and the parameter identification [16], [33]. But the algorithm still does not seem satisfactory, as its convergence can be guaranteed only under some restriction; see (4.2) in [29]. Next, we propose an improved variant of Algorithm 4.1 in [29].

Let $H = B^t \hat{A}^{-1} B$. Consider the equation

$$H\psi = g_i, \tag{2.13}$$

where $g_i = B^t x_{i+1} - g$ comes from Algorithm 2.1. We apply the PCG method with the preconditioner $\hat{K}$ to solve system (2.13) and let $\Psi_H(g_i)$ be the approximation generated by this iteration. Assume that the approximation satisfies

$$\|\Psi_H(g_i) - H^{-1}g_i\|_H \leq \delta_g \, \|H^{-1}g_i\|_H \tag{2.14}$$

for some $\delta_g \in (0,1)$. For the approximation $d_i = \Psi_H(g_i)$, we introduce a relaxation parameter $\overline{\tau}_i$ such that the error

$$\|\overline{\tau}_i \, d_i - K^{-1}g_i\|_K^2$$

is minimized. If $d_i \neq 0$, the direct calculation gives

$$\overline{\tau}_i = \frac{(g_i, d_i)}{(Kd_i, d_i)} = \frac{(g_i, d_i)}{(A^{-1}Bd_i, Bd_i)}.$$

But the action of A^{-1} is usually very expensive, and thus will be replaced by the action of Ψ_A:

$$\tau_i = \frac{(g_i, d_i)}{(\Psi_A(Bd_i), Bd_i)} \approx \overline{\tau}_i. \tag{2.15}$$

With this parameter τ_i, we propose the following new algorithm.

Algorithm 2.2 (nonlinear inexact Uzawa with mixed iteration). *Given* $\{x_0, y_0\} \in R^n \times R^m$, *the sequence of pairs* $\{x_i, y_i\} \in R^n \times R^m$ *is defined for* $i = 1, 2, \ldots,$ *as follows.*

Step 1. *Compute* $f_i = f - (Ax_i + By_i)$ *and* $\Psi_A(f_i)$*; update*

$$x_{i+1} = x_i + \Psi_A(f_i). \tag{2.16}$$

Step 2. *Compute* $g_i = B^t x_{i+1} - g$ *and* $d_i = \Psi_H(g_i)$*. Then compute the parameter* τ_i*:*

$$\tau_i = \begin{cases} \frac{(g_i, d_i)}{(\Psi_A(Bd_i), Bd_i)} & \textit{if} \quad d_i \neq 0; \\ 1 & \textit{if} \quad d_i = 0 \end{cases} \tag{2.17}$$

and update

$$y_{i+1} = y_i + \frac{1}{2}\tau_i \, d_i. \tag{2.18}$$

Remark 2.2. Clearly when both f_i and g_i vanish, the vectors x_i and y_i are the exact solution of (1.3). Thus Algorithm 2.2 terminates.

Next we shall analyze the convergence of Algorithm 2.2. Let $\kappa^* = \operatorname{cond}(\hat{A}^{-1}A)$ and $\kappa = \kappa^*(1+\delta_g)/(1-\delta_g)$. It follows from (2.14) that there is a symmetric and positive definite matrix $\hat{Q}_i$ (see Lemma 9 in [4]) such that $\hat{Q}_i^{-1} g_i = \Psi_H(g_i)$ and all eigenvalues of the matrix $\hat{Q}_i^{-1}H$ are in the interval $[1-\delta_g, 1+\delta_g]$. Using this property, one can directly check that

$$\operatorname{cond}(\hat{Q}_i^{-1}K) \leq \kappa = \frac{1+\delta_g}{1-\delta_g}\operatorname{cond}(\hat{A}^{-1}A).$$

This relation tells us the *actual effect* of introducing of approximation Ψ_H: when $\operatorname{cond}(\hat{K}^{-1}K) \gg \operatorname{cond}(\hat{A}^{-1}A)$, the effect of $\Psi_H(g_i)$ $(= \hat{Q}_i^{-1}g_i)$ amounts to generating a new preconditioner $\hat{Q}_i$ such that $\operatorname{cond}(\hat{Q}_i^{-1}K)$ is much more improved than $\operatorname{cond}(\hat{K}^{-1}K)$ and has about the same magnitude as $\operatorname{cond}(\hat{A}^{-1}A)$, e.g., less than three times $\operatorname{cond}(\hat{A}^{-1}A)$ when we take $\delta_g = \frac{1}{2}$.

Let δ_f and δ_d be two parameters in (2.6) and (2.7), respectively, with f_i and d_i given in Algorithm 2.2, and define

$$\beta = \sqrt{1 - \frac{4\kappa(1-2\delta_d)}{(1+\kappa)^2(1-\delta_d)^2}}; \tag{2.19}$$

then Algorithm 2.2 can be viewed as a variant of Algorithm 2.1 with $\hat{K}$ replaced by $\hat{Q}_i$. The following theorem follows from Lemma 2.1.

Theorem 2.2. *Assume that* (2.6) *and* (2.7) *are satisfied with* $\delta_f < \frac{1}{3}$ *and* $\delta_d < \frac{1}{2}$*; then Algorithm* 2.2 *converges. Moreover, the following estimate holds:*

$$|||E_{i+1}||| \leq \rho \, |||E_i|||, \quad i = 0, 1 \ldots, \tag{2.20}$$

which implies for $i = 1, 2, \ldots$ *that*

$$\|e_i^x\|_A \leq (\sqrt{1+4\delta_f} + \rho)\rho^{i-1}|||E_0|||, \quad \|e_i^y\|_K \leq \rho^i |||E_0|||, \tag{2.21}$$

where the rate of convergence $\rho(<1)$ *can be estimated by*

$$\rho = \begin{cases} \sqrt{\delta_f + \delta_f^2} + \delta_f & \text{for} \quad 0 < \frac{1+\beta}{2} \leq \frac{4\delta_f}{1+\delta_f}; \\ 1 - \frac{1}{4}(1-\beta)(1+\delta_f) & \text{for} \quad \frac{4\delta_f}{1+\delta_f} < \frac{1+\beta}{2} < 1. \end{cases} \tag{2.22}$$

Remark 2.3. We see from Theorem 2.2 that the convergence of Algorithm 2.2 is independent of the spectrum of the preconditioned Schur complement $\hat{K}^{-1}K$, and the convergence rate of this new algorithm depends only on the condition number κ^*, not on $\mathrm{cond}(\hat{K}^{-1}K)$. In contrast to Algorithm 2.1, Algorithm 2.2 should be very efficient for the case when $\mathrm{cond}(\hat{K}^{-1}K) \gg \mathrm{cond}(\hat{A}^{-1}A)$. This seems to be an important advantage of the new algorithm over the existing iterative methods for saddle-point problems. The coefficient 1/2 in (2.18) is obtained by the worst case $\delta_g \to 1^-$ (refer to [29]). In applications, the parameter δ_g is much less than 1, so we can choose a larger parameter than 1/2 in (2.18), e.g., 7/10.

3. Nonlinear inexact Uzawa algorithms for nonlinear saddle-point problems.

In this section, we discuss how to effectively extend the new Algorithm 2.2 proposed in section 2 for the linear saddle-point problem (1.3) to solve the nonlinear saddle-point system (1.1), which is assumed to arise from some convex minimization problems with linear constraints, e.g., of the form

$$\begin{cases} \min\limits_{x\in R^n} \{J(x) - (f,x)\} \\ \ \text{s.t.}\ \ B^t x = g, \end{cases} \tag{3.1}$$

where $J(x)$ is the function satisfying $\nabla J(x) = F(x)$.

3.1. Notation and assumptions.

We start with a few smoothness descriptions on the nonlinear mapping $F : R^n \to R^n$ in (1.1) and recall some existing results from [18] and [36], which will be used in the subsequent analysis.

As standard assumptions for nonlinear systems (cf. [1], [14]), we assume that F is Lipschitzian and strongly monotone with modulus μ, i.e.,

$$(F(\xi) - F(\eta), \xi - \eta) \geq \mu\, \|\xi - \eta\|^2 \quad \forall\ \xi, \eta \in R^n. \tag{3.2}$$

By Rademacher's theorem [18], the Lipschitzian property of F implies that F is differentiable almost everywhere. Let D_F be the set of points where F is differentiable, and let $\nabla F(\xi)$ be the gradient of F at $\xi \in D_F$. Then at any point $x \in R^n$, we introduce a set $\partial_s F(x)$:

$$\partial_s F(x) = \left\{ \lim_{\substack{\xi \to x \\ \xi \in D_F}} \nabla F(\xi) \right\}.$$

With this set, we can define a generalized Jacobian of F at x in the sense of Clarke [18] by

$$\partial F(x) = \mathrm{co}\, \partial_s F(x),$$

where $\mathrm{co}\, \partial_s F(x)$ is the convex hull of the set.

It is known (cf. [18]) that if F is locally Lipschitzian, then the following generalized mean-value theorem holds: for any $\xi,\ \eta \in R^n$,

$$F(\xi) - F(\eta) \in \mathrm{co}\, \partial F(\overline{\xi\eta})(\xi - \eta), \tag{3.3}$$

where $\overline{\xi\eta}$ is the line segment between ξ and η, and $\mathrm{co}\, \partial F(\overline{\xi\eta}) = \mathrm{co}\{V \in \partial F(\zeta),\ \zeta \in \overline{\xi\eta}\}$.

A nice consequence (cf. [36]) of the strong monotone property (3.2) is that all matrices from $\partial F(\eta)$ for any $\eta \in R^n$ are positive definite, and the following holds for any $V \in \partial F(\eta)$:

$$(V\xi,\, \xi) \geq \mu\, (\xi,\, \xi) \quad \forall\, \xi \in R^n. \tag{3.4}$$

As in [14], we do not assume F is differentiable everywhere but that it is semismooth on R^n in the sense that for any $\xi \in R^n$ there is a positive definite matrix A_ξ such that

$$\lim_{\substack{\alpha\to 0\\ A_\xi\in\partial_s F(\xi+\alpha)}} \frac{\|F(\xi+\alpha)-F(\xi)-A_\xi\alpha\|}{\|\alpha\|}=0. \tag{3.5}$$

Nonlinear saddle-point problems with nondifferentiable but semismooth vector-valued functions F arise from some convex optimizations and numerical solutions of certain nonlinear partial differential equations; we refer to [14] for such examples.

3.2. Properties of F in terms of its generalized Jacobian. Note that all the descriptions in section 3.1 of the smoothness of the nonlinear mapping $F : R^n \to R^n$ are in terms of the l^2-norm. As we will see later, it is more accurate to interpret these smoothness properties in terms of the so-called energy-norm, that is, the induced norm by the generalized Jacobian of F, especially by the generalized Jacobian A_x of F at x, where $\{x, y\} \in R^n \times R^m$ is the exact solution of system (1.1). This will be the task of this section. For the sake of simplicity, we shall write A_x as A below.

First, directly from (3.4) and (3.5), we know that if F is semismooth on R^n, then for any $\xi \in R^n$ there is a positive definite matrix A_ξ such that

$$\lim_{\substack{\alpha\to 0\\ A_\xi\in\partial_s F(\xi+\alpha)}} \frac{\|F(\xi+\alpha)-F(\xi)-A_\xi\alpha\|_{A^{-1}}}{\|\alpha\|_A}=0. \tag{3.6}$$

Next, by the strictly monotone and Lipschitzian property of F we immediately know there are two positive constants c_0 and C_0, which will be frequently needed later, such that

$$c_0\,\|\xi-\eta\|_A^2 \le (F(\xi)-F(\eta),\xi-\eta) \quad \forall\,\xi,\ \eta\in R^n, \tag{3.7}$$

$$\|F(\xi)-F(\eta)\|_{A^{-1}}^2 \le C_0\,\|\xi-\eta\|_A^2 \quad \forall\,\xi,\ \eta\in R^n. \tag{3.8}$$

The use of constants c_0 and C_0 is more reasonable than the use of the corresponding constants in the sense of the l^2-norm. For instance, when $F(x)$ is linear, say $F(x) = Ax$, then $c_0 = C_0 = 1$, but the corresponding constants in the sense of the l^2-norm depend on the smallest and largest eigenvalues of A.

Starting now, we shall often use $S_V(\xi, r)$ to denote a ball in R^n which is centered at point ξ, with radius r measured in the $\|\cdot\|_V$-norm. The next lemma gives two further properties of the nonlinear vector-valued function F.

LEMMA 3.1. *There are two positive constants $c_1 \le 1$ and $C_1 \ge 1$ depending only on constants c_0 and C_0 such that*

$$(F(\zeta)-F(\xi),\zeta-\xi) \le C_1(A_\eta(\zeta-\xi),\zeta-\xi) \quad \forall\zeta,\ \xi,\ \eta\in R^n, \tag{3.9}$$

$$\|F(\zeta)-F(\xi)\|_{A_\eta^{-1}} \ge c_1\,\|\zeta-\xi\|_{A_\eta} \qquad \forall\zeta,\ \xi,\ \eta\in R^n. \tag{3.10}$$

Proof. We first consider (3.9). It follows from (3.8) that for any $\zeta,\ \xi \in R^n$,

$$(F(\zeta)-F(\xi),\zeta-\xi) \le \|F(\zeta)-F(\xi)\|_{A^{-1}}\,\|\zeta-\xi\|_A \le \sqrt{C_0}\|\zeta-\xi\|_A^2. \tag{3.11}$$

But for any $\eta \in R^n$, by (3.6) there exists a positive number $\overline{r} = \overline{r}(\eta)$ such that

$$\|F(\eta+\alpha)-F(\eta)-A_\eta\alpha\|_{A^{-1}} \le \frac{c_0}{2}\|\alpha\|_A \quad \forall\ \alpha\in S_A(0,\overline{r}). \tag{3.12}$$

This, together with (3.7), leads to

$$\begin{aligned}\|\alpha\|_A^2 \le \frac{1}{c_0}(F(\eta+\alpha)-F(\eta),\alpha) &= \frac{1}{c_0}\{(F(\eta+\alpha)-F(\eta)-A_\eta\alpha,\alpha)+(A_\eta\alpha,\alpha)\}\\ &\le \frac{1}{c_0}\{\|F(\eta+\alpha)-F(\eta)-A_\eta\alpha\|_{A^{-1}}\,\|\alpha\|_A+(A_\eta\alpha,\alpha)\}\\ &\le \frac{1}{2}\|\alpha\|_A^2+\frac{1}{c_0}(A_\eta\alpha,\alpha).\end{aligned}$$

So we have

$$\|\alpha\|_A^2 \le \frac{2}{c_0}(A_\eta\alpha,\alpha) \quad \forall \alpha\in S_A(0,\bar r). \tag{3.13}$$

Noting that inequality (3.13) is invariant with respect to any constant scaling of α, (3.9) follows readily from (3.11) and (3.13) with $C_1 = 2\sqrt{C_0}/c_0$, or $C_1 = 1$ if $\sqrt{C_0}/c_0 < 1/2$.

Now we consider (3.10). By (3.12) and (3.8), we derive

$$\begin{aligned}(A_\eta\alpha,\alpha) &\le \|A_\eta\alpha\|_{A^{-1}}\,\|\alpha\|_A\\ &\le (\|A_\eta\alpha - F(\eta+\alpha)+F(\eta)\|_{A^{-1}} + \|F(\eta+\alpha)-F(\eta)\|_{A^{-1}})\,\|\alpha\|_A\\ &\le \left(\frac{c_0}{2}+\sqrt{C_0}\right)\|\alpha\|_A^2 \quad \forall\alpha\in S_A(0,\bar r), \qquad (3.14)\end{aligned}$$

which is invariant up to any constant scaling of α, while by (3.7) we have

$$\|\alpha\|_A^2 \le \frac{1}{c_0}(F(\eta+\alpha)-F(\eta),\alpha) \le \frac{1}{c_0}\|F(\eta+\alpha)-F(\eta)\|_{A_\eta^{-1}}\,\|\alpha\|_{A_\eta}. \tag{3.15}$$

Then (3.10) follows immediately from (3.14) and (3.15), with $c_1 = 2c_0/(c_0+2\sqrt{C_0})$ or $c_1 = 1$ if $\sqrt{C_0}/c_0 < 1/2$. □

We end this section by assuming some sort of Lipschitzian property on the generalized Jacobian of F: there exists a positive constant L such that for any two vectors $\xi,\eta\in R^n$,

$$\|A^{-\frac{1}{2}}(V_\xi - V_\eta)A^{-\frac{1}{2}}\| \le L\,\|\xi-\eta\|_A \quad \forall V_\xi\in\partial F(\xi),\ V_\eta\in\partial F(\eta). \tag{3.16}$$

3.3. Algorithms and their convergence. We are now going to extend the new Algorithm 2.2 in section 2 for the linear saddle-point problem (1.3) to solve the nonlinear saddle-point problem (1.1). As we have observed, an essential improvement of this new algorithm over the existing ones (cf. [10]) lies in the fact that its convergence is guaranteed and its rate of convergence can be estimated by assuming only constant upper bounds for the error reduction factors ($\delta_f < 1/3$ and $\delta_d < 1/2$) in the nonlinear inner iterations. These conditions may be easily satisfied, for example, by the approximate inverse generated by the PCG iteration with the preconditioner $\hat{A}$. Such loose requirements come as the consequence of the choice of a particular norm $|||\cdot|||$; see Remark 2.1 in [29]. In order to preserve this good feature in the current nonlinear saddle-point system, one should use a norm similar to $|||\cdot|||$ involving the matrix $A = A_x$. But unlike the linear saddle-point problem, the matrix A_x is not available now since it involves the x-component of the exact solution $\{x,y\}$ to system (1.1). This fact brings in one of the major difficulties of nonlinear systems and can be regarded as the main distinction between the linear and nonlinear saddle-point problems.

Now, we discuss how to extend Algorithm 2.2 of section 2 to solve the nonlinear saddle-point problem (1.1). Let $\{x_i, y_i\} \in R^n \times R^m$ be the ith iterate, and set

$$f_i = f - F(x_i) - By_i.$$

Let x_{i+1} be an approximate solution of the nonlinear equation

$$F(\xi) = f - By_i \tag{3.17}$$

such that the residual $\varepsilon_i = F(x_{i+1}) - (f - By_i)$ satisfies

$$\|\varepsilon_i\|_{A^{-1}} \le \delta_0 \, \|f_i\|_{A^{-1}} \tag{3.18}$$

with $0 \le \delta_0 < 1$. In general, the approximation x_{i+1} can be obtained by some iterative method with x_i as a natural initial guess. This will be discussed in detail in section 3.4.

Let $A_i = A_{x_i}$ be a positive definite matrix as defined in (3.5), and let $\hat{A}_i$ be a (positive definite) preconditioner of A_i; then we obtain an exact Schur complement at x_i and its approximation:

$$K_i = B^t A_i^{-1} B, \quad H_i = B^t \hat{A}_i^{-1} B.$$

Let $\hat{K}_i$ be a preconditioner for H_i, and set $g_i = B^t x_{i+1} - g$. Similarly to the introduction of the mapping Ψ_H in (2.14), we define a nonlinear mapping $\Psi_{H_i} : R^m \to R^m$ such that

$$\|\Psi_{H_i}(g_i) - H_i^{-1} g_i\|_{H_i} \le \delta_g \, \|H_i^{-1} g_i\|_{H_i} \tag{3.19}$$

for some $\delta_g \in (0, 1)$. Let $d_i = \Psi_{H_i}(g_i)$; then we introduce a nonlinear solver $\Psi_{A_{i+1}} : R^n \to R^n$ satisfying

$$\|\Psi_{A_{i+1}}(Bd_i) - A_{i+1}^{-1} Bd_i\|_{A_{i+1}} \le \gamma \, \|A_{i+1}^{-1} Bd_i\|_{A_{i+1}} \tag{3.20}$$

for some $\gamma \in [0, 1)$. As we observed in the linear saddle-point case, a relaxation parameter $\overline{\tau}_i$ (see (2.15)) is important to ensure the convergence of our new algorithm:

$$\overline{\tau}_i = \frac{(g_i, \, d_i)}{(\Psi_A(Bd_i), Bd_i)} \quad \text{for} \quad d_i \ne 0.$$

Unfortunately, the matrix $A = A_x$ is no longer available for the current nonlinear problem (1.1). One alternative is to use the approximation A_{i+1} of A; this leads to the following new choice of the relaxation parameter τ_i:

$$\tau_i = \frac{(g_i, \, d_i)}{(\Psi_{A_{i+1}}(Bd_i), Bd_i)} \quad \text{for} \quad d_i \ne 0. \tag{3.21}$$

With this parameter and the motivation of Algorithm 2.2 for linear saddle-point problems, we propose the following algorithm for solving the nonlinear saddle-point system (1.1).

ALGORITHM 3.1 (nonlinear inexact Uzawa with mixed iteration). *Given $\{x_0, y_0\} \in R^n \times R^m$, the sequence $\{x_i, y_i\} \in R^n \times R^m$ is defined for $i = 1, 2, \ldots$ as follows:*

Step 1. Compute x_{i+1} such that

$$F(x_{i+1}) = f - By_i + \varepsilon_i.$$

Step 2. *Compute* $g_i = B^t x_{i+1} - g$ *and* $d_i = \Psi_{H_i}(g_i)$. *Then compute the parameter* τ_i*:*

$$\tau_i = \begin{cases} \frac{(g_i,\, d_i)}{(\Psi_{A_{i+1}}(Bd_i),\, Bd_i)} & \text{if} \quad d_i \neq 0; \\ 1 & \text{if} \quad d_i = 0 \end{cases}$$

and update

$$y_{i+1} = y_i + \frac{1}{2}\tau_i\, d_i. \tag{3.22}$$

To understand and more accurately describe the convergence of this new algorithm, we need to introduce a few more parameters. First, by (3.6) we know that for any parameter $\omega \in (0,\ 1)$, there is a positive number r_ω such that

$$\|F(x+\alpha) - F(x) - Ax\|_{A^{-1}} \leq \omega\, \|\alpha\|_A \quad \forall \alpha \in S_A(0, r_\omega). \tag{3.23}$$

Then we introduce a constant δ_d that is the minimal positive number satisfying

$$\|\Psi_{A_{i+1}}(Bd_i) - A^{-1}Bd_i\|_A \leq \delta_d\, \|A^{-1}Bd_i\|_A. \tag{3.24}$$

Now set $r_\gamma = (1-2\gamma)/(6L)$ for any positive parameter $\gamma < \frac{1}{2}$, and

$$\kappa_i = \mathrm{cond}(\hat{A}_i^{-1}A_i), \quad \kappa_0 = \frac{1+\delta_g}{1-\delta_g}\max_i \kappa_i, \quad \beta_0 = \sqrt{1 - \frac{4\kappa_0(1-2\delta_d)}{(1+\kappa_0)^2(1-\delta_d)^2}}, \tag{3.25}$$

we will see $\delta_d < \frac{1}{2}$ when $x_{i+1} \in S_A(x, r_\gamma)$ (Lemma 3.7), and hence we have $0 < \beta_0 < 1$.

The following few parameters will be used to describe the convergence region and convergence rate of Algorithm 3.1:

$$\rho_0 = \begin{cases} \sqrt{\delta_0 + \delta_0^2} + \delta_0 & \text{for} \quad 0 < \frac{1+\beta_0}{2} \leq \frac{4\delta_0}{1+\delta_0}, \\ 1 - \frac{1}{4}(1-\beta_0)(1+\delta_0) & \text{for} \quad \frac{4\delta_0}{1+\delta_0} < \frac{1+\beta_0}{2} < 1 \end{cases} \tag{3.26}$$

and

$$\rho_\omega = \frac{\omega\sqrt{1+\delta_0}(1+\sqrt{\delta_0})}{c_0}, \quad \omega_0 = \frac{c_0(1-\rho_0)}{\sqrt{1+\delta_0}(1+\sqrt{\delta_0})}, \quad r_\omega^* = \frac{c_0}{(1+\sqrt{\delta_0})}\min\{r_\omega,\ r_\gamma\}. \tag{3.27}$$

Our final preparation is to choose an appropriate norm in which the convergence can be ensured and well measured. As for the linear saddle-point problem, we define $|||\cdot|||$ to be the same as in (2.8), but with $A = A_x$ and $K = B^tA^{-1}B$ here.

Now we are ready to state our main results of this section about the convergence of Algorithm 3.1, whose proof will be provided in section 3.5.

Theorem 3.2. *Let* $\omega_0 < 1$ *be a fixed positive constant,* $\omega \in (0, \omega_0)$ *be a given parameter, and* r_ω *be the maximal positive number such that estimate* (3.23) *is satisfied. Assume that* (3.20) *holds with* $\gamma < \frac{1}{2}$ *and that the initial guess* $\{x_0,\ y_0\}$ *satisfies* $|||E_0||| \leq r_\omega^*$. *If the tolerance* ε_i *in Step* 1 *satisfies* (3.18) *with* $\delta_0 < \frac{1}{3}$, *then Algorithm* 3.1 *converges, and the rate of convergence can be estimated by*

$$|||E_{i+1}||| \leq \rho_0^*\, |||E_i|||, \quad i = 0, 1, 2, \ldots, \tag{3.28}$$

where $\rho_0^* = \rho_0 + \rho_\omega < 1$, *and*

$$E_i = \{\sqrt{\delta_0}f_i, e_i^y\}, \quad f_i = f - F(x_i) - By_i, \quad e_i^y = y - y_i.$$

Remark 3.1. From Theorem 3.2 we see that the preconditioner $\hat{K}_i$ for H_i can be chosen in Algorithm 3.1 without any particular restriction, and the approximation accuracy parameters δ_0 and δ_d are independent of any other parameter, unlike in most existing algorithms. The rate of convergence ρ_0^* depends only on δ_d and β_0, and does not depend directly on $\text{cond}(\hat{K}_i^{-1}K)$; therefore no proper scalings of the preconditioner $\hat{K}_i$ are required as in the existing inexact Uzawa algorithms.

Remark 3.2. In Theorem 3.2, the initial guess $\{x_0, y_0\}$ is required to lie within a small neighborhood of $\{x, y\}$. Care must be taken for the choice of such initial guesses. In applications, the initial guess may be obtained using a globally convergent algorithm for (1.1) with one or two iterations, for example, the simple perturbation method as described below.

Given a small positive number μ, approximate (1.1) by the perturbed system

$$F(x) + By = f, \quad B^t x - \mu\, y = g. \tag{3.29}$$

Expressing y in terms of x from the second equation and then substituting it into the first equation, we obtain

$$F(x) + \frac{1}{\mu} BB^t x = f + \frac{1}{\mu} Bg. \tag{3.30}$$

One can solve this nonlinear equation using some classical iterative methods, for instance, the steepest descent method, which is known to have slow convergence but usually converges very fast at the first few iterations. Once an approximation of x is available, the approximation of y can be obtained directly from the second equation in (3.29). As this process is used to generate only an initial guess, the perturbation parameter μ need not be too small, e.g., one may take $\mu = 0.1$.

3.4. Solution of system (3.17). One major task in Algorithm 3.1 is to find some effective way to compute x_{i+1} such that the tolerance requirement (3.18) is satisfied with $\delta_0 < \frac{1}{3}$. In this subsection we will propose an iterative algorithm for computing x_{i+1} which meets the requirement.

Let x_i^* be the exact solution of (3.17); then we have $f - By_i = F(x_i^*)$, and (3.18) can be written as

$$\|F(x_{i+1}) - F(x_i^*)\|_{A^{-1}} \le \delta_0 \|F(x_i) - F(x_i^*)\|_{A^{-1}}. \tag{3.31}$$

When nonlinear equation (3.17) is solved by an iterative method with the initial guess x_i, condition (3.31) should come from the convergence results in the underlying norm. Unfortunately, this conclusion is not straightforward here since the convergences of *most iterative methods* for nonlinear equations are analyzed in *the l^2-norm* (cf. [30], [31], [36]), not in the "energy-norm" as required in (3.31).

Let G_i be a functional defined by

$$G_i(\xi) = J(\xi) + (By_i, \xi) - (f, \xi),$$

where $J(x)$ is the functional in (3.1) satisfying $\nabla J(x) = F(x)$. Then (3.17) amounts to the following minimization problem: Find $x_i^* \in R^n$ such that

$$G_i(x_i^*) = \min_{\xi \in R^n} G_i(\xi). \tag{3.32}$$

Next, we propose a PCG-type method to solve this minimization problem.

Algorithm 3.2 (PCG-type method for solving (3.32)). *Set*

$$x_i^{(0)} = x_i, \quad p_i^{(0)} = -\hat{A}_i^{-1}\nabla G_i(x_i),$$

then generate a sequence $\{x_i^{(k)}\}_{k=1}^{\infty}$ *as follows.*

Step 1. *Compute the parameter* $\tau_i^{(k-1)}$ *such that*

$$G_i(x_i^{(k-1)} + \tau_i^{(k-1)} p_i^{(k-1)}) = \min_{\tau} G_i(x_i^{(k-1)} + \tau\, p_i^{(k-1)}). \tag{3.33}$$

Update

$$x_i^{(k)} = x_i^{(k-1)} + \tau_i^{(k-1)} p_i^{(k-1)}. \tag{3.34}$$

Step 2. *Compute*

$$\theta_i^{(k)} = -(\hat{A}_i^{-1}\nabla G_i(x_i^{(k)}),\, A_{x_i^{(k)}} p_i^{(k-1)})/(A_{x_i^{(k)}} p_i^{(k-1)},\ p_i^{(k-1)}). \tag{3.35}$$

Compute

$$p_i^{(k)} = -\hat{A}_i^{-1}\nabla G_i(x_i^{(k)}) - \theta_i^{(k)} p_i^{(k-1)}. \tag{3.36}$$

Before analyzing the convergence of Algorithm 3.2, let us introduce a few useful constants. For the sake of simplicity, we shall write $A_i^{(k)} = A_{x_i^{(k)}}$ below. First, we can easily see the existence of the two constants $C_i^{(k)}$ and $c_i^{(k)}$ from Lemma 3.1 and by noting the relation

$$\nabla G_i(\xi + \alpha) - \nabla G_i(\xi) = \nabla J(\xi + \alpha) - \nabla J(\xi) = F(\xi + \alpha) - F(\xi).$$

The first constant $C_i^{(k)} \geq 1$ is the smallest positive number satisfying

$$(\nabla G_i(\xi + \alpha) - \nabla G_i(\xi), \alpha) \leq C_i^{(k)} (A_i^{(k)} \alpha, \alpha) \tag{3.37}$$

for $\xi = x_i^{(k)}$ and $\alpha = s\, p_i^{(k)}$ with all $s > 0$, also for $\xi = x_i^*$ and $\alpha = t(x_i^{(k)} - x_i^*)$ with all $t \in (0,1)$; the second constant $c_i^{(k)} \leq 1$ is the largest positive number satisfying

$$c_i^{(k)} \|x_i^{(k)} - x_i^*\|_{A_i^{(k)}} \leq \|\nabla G_i(x_i^{(k)}) - \nabla G_i(x_i^*)\|_{(A_i^{(k)})^{-1}}\,. \tag{3.38}$$

It is obvious that for a less accurate estimate one may simply take the above two constants $C_i^{(k)}$ and $c_i^{(k)}$ to be the constants C_1 and c_1 from (3.9) and (3.10), respectively.

Now define

$$\kappa_i^{(k)} = \mathrm{cond}(\hat{A}_i^{-\frac{1}{2}} A_i^{(k)} \hat{A}_i^{-\frac{1}{2}}), \quad \rho_i^{(k)} = 1 - \left(\frac{c_i^{(k)}}{C_i^{(k)}}\right)^2 \frac{4\,\kappa_i^{(k)}}{(\kappa_i^{(k)} + 1)^2}.$$

Clearly, we see that $\rho_i^{(k)}$ lies in the range $0 < \rho_i^{(k)} < 1$. The following estimate holds on the rate of convergence with Algorithm 3.2.

Lemma 3.3. *Let* x_i^* *be the exact solution of* (3.32), *and let the sequence* $\{x_i^{(k)}\}_{k=1}^{\infty}$ *be generated by Algorithm* 3.2. *Then the following estimate holds:*

$$G_i(x_i^{(k+1)}) - G_i(x_i^*) \leq \rho_i^{(k)}\, (G_i(x_i^{(k)}) - G_i(x_i^*)), \quad k = 0, 1 \ldots. \tag{3.39}$$

If we set $x_{i+1} = x_i^{(k_0)}$ *for some positive integer* k_0*, then*

$$G_i(x_{i+1}) - G_i(x_i^*) \le \left(\prod_{k=1}^{k_0} \rho_i^{(k-1)}\right)(G_i(x_i^{(k)}) - G_i(x_i^*)). \tag{3.40}$$

Proof. Using the generalized mean-value theorem and (3.34), we have for any $\tau > 0$,

$$\begin{aligned} &G_i(x_i^{(k)} + \tau p_i^{(k)}) - G_i(x_i^{(k)}) \\ &= \int_0^1 \Big(\nabla G_i(x_i^{(k)} + t\,\tau p_i^{(k)}),\, \tau p_i^{(k)}\Big)dt \\ &= \tau\Big(\nabla G_i(x_i^{(k)}),\, p_i^{(k)}\Big) + \int_0^1 \Big(\nabla G_i(x_i^{(k)} + t\,\tau p_i^{(k)}) - \nabla G_i(x_i^{(k)}),\, \tau p_i^{(k)}\Big)dt. \end{aligned} \tag{3.41}$$

But it follows from (3.37) with $\xi = x_i^{(k)}$ and $\alpha = t\,\tau p_i^{(k)}$ that

$$\Big(\nabla G_i(x_i^{(k)} + t\,\tau p_i^{(k)}) - \nabla G_i(x_i^{(k)}),\, \tau p_i^{(k)}\Big) \le C_i^{(k)} t\,\tau^2 (A_i^{(k)} p_i^{(k)},\, p_i^{(k)}).$$

Plugging this into (3.41) yields

$$G_i(x_i^{(k)} + \tau p_i^{(k)}) - G_i(x_i^{(k)}) \le \tau(\nabla G_i(x_i^{(k)}),\ p_i^{(k)}) + \frac{1}{2}C_i^{(k)}\tau^2(A_i^{(k)}p_i^{(k)},\ p_i^{(k)}). \tag{3.42}$$

Noting that the parameter $\tau_i^{(k)}$ defining $x_i^{(k+1)}$ satisfies (3.33), we obtain

$$G_i(x_i^{(k+1)}) - G_i(x_i^{(k)}) \le -\frac{(\nabla G_i(x_i^{(k)}),\ p_i^{(k)})^2}{2C_i^{(k)}(A_i^{(k)}p_i^{(k)},\ p_i^{(k)})} \tag{3.43}$$

if we take in (3.42) that

$$\tau = -\frac{(\nabla G_i(x_i^{(k)}),\ p_i^{(k)})}{C_i^{(k)}(A_i^{(k)}p_i^{(k)},\ p_i^{(k)})}.$$

To further estimate the fraction in (3.43), we first know from (3.33) that

$$(\nabla G_i(x_i^{(k)}),\ p_i^{(k-1)}) = 0. \tag{3.44}$$

Using this, and making the scalar product of both sides of (3.36) with $\nabla G_i(x_i^{(k)})$, we derive

$$(\nabla G_i(x_i^{(k)}),\ p_i^{(k)}) = -\|\hat{A}_i^{-\frac{1}{2}}\nabla G_i(x_i^{(k)})\|^2. \tag{3.45}$$

On the other hand, by direct computing using (3.36) and (3.35) we get

$$\begin{aligned} (A_i^{(k)}p_i^{(k)},\ p_i^{(k)}) &= \|\hat{A}_i^{-1}\nabla G_i(x_i^{(k)})\|^2_{A_i^{(k)}} - \frac{(\hat{A}_i^{-1}\nabla G_i(x_i^{(k)}),\ A_i^{(k)}p_i^{(k-1)})^2}{\|p_i^{(k-1)}\|^2_{A_i^{(k)}}} \\ &\le \|\hat{A}_i^{-1}\nabla G_i(x_i^{(k)})\|^2_{A_i^{(k)}}. \end{aligned} \tag{3.46}$$

Now it follows from (3.45), (3.46), (3.43), and a well-known matrix-eigenvalue inequality (see (3.1) in [29]) that

$$G_i(x_i^{(k+1)}) - G_i(x_i^{(k)}) \leq -\frac{\|\hat{A}_i^{-\frac{1}{2}}\nabla G_i(x_i^{(k)})\|^4}{2C_i^{(k)}\|\hat{A}_i^{-1}\nabla G_i(x_i^{(k)})\|^2_{A_i^{(k)}}}$$

$$\leq -\frac{1}{2C_i^{(k)}}\,\frac{4\kappa_i^{(k)}}{(\kappa_i^{(k)}+1)^2}\|\nabla G_i(x_i^{(k)})\|^2_{(A_i^{(k)})^{-1}}. \tag{3.47}$$

But noting $\nabla G_i(x_i^*) = 0$, we deduce from (3.38) that

$$\|\nabla G_i(x_i^{(k)})\|^2_{(A_i^{(k)})^{-1}} = \|\nabla G_i(x_i^{(k)}) - \nabla G_i(x_i^*)\|^2_{(A_i^{(k)})-1} \geq (c_i^{(k)})^2\,\|x_i^{(k)} - x_i^*\|^2_{A_i^{(k)}}.$$

This, along with (3.47), implies

$$G_i(x_i^{(k+1)}) - G_i(x_i^{(k)}) \leq -\frac{(c_i^{(k)})^2}{2C_i^{(k)}}\,\frac{4\kappa_i^{(k)}}{(\kappa_i^{(k)}+1)^2}\|x_i^{(k)} - x_i^*\|^2_{A_i}. \tag{3.48}$$

Furthermore, by the generalized mean-value theorem and the fact that $\nabla G_i(x_i^*) = 0$ again, we can write

$$G_i(x_i^{(k)}) - G_i(x_i^*) = \int_0^1 \Big(\nabla G_i(x_i^* + t(x_i^{(k)} - x_i^*)) - \nabla G_i(x_i^*),\, x_i^{(k)} - x_i^*\Big)dt.$$

Taking $\xi = x_i^*$ and $\alpha = t\,(x_i^{(k)} - x_i^*)$ in (3.37), we come to

$$G_i(x_i^{(k)}) - G_i(x_i^*) \leq \frac{C_i^{(k)}}{2}\|x_i^{(k)} - x_i^*\|^2_{A_i^{(k)}}.$$

Combining this with (3.48) leads to

$$G_i(x_i^{(k+1)}) - G_i(x_i^{(k)}) \leq -\left(\frac{c_i^{(k)}}{C_i^{(k)}}\right)^2 \frac{4\kappa_i^{(k)}}{(\kappa_i^{(k)}+1)^2}(G(x_i^{(k)}) - G(x_i^*)).$$

Therefore

$$G_i(x_i^{(k+1)}) - G_i(x_i^*) \leq \left(1 - \left(\frac{c_i^{(k)}}{C_i^{(k)}}\right)^2 \frac{4\kappa_i^{(k)}}{(\kappa_i^{(k)}+1)^2}\right)(G(x_i^{(k)}) - G(x_i^*)),$$

which proves the desired result. □

Remark 3.3. Algorithm 3.2 is always convergent, so we have $x_i^{(k)} \to x_i^*$ and $p_i^{(k)} \to 0$ as $k \to +\infty$. Then by (3.6) one can verify that $C_i^{(k)} \to 1$ and $c_i^{(k)} \to 1$ as $k \to +\infty$. This implies

$$\lim_{k\to\infty} \rho_i^{(k)} = 1 - \frac{4\kappa_i}{(\kappa_i+1)^2} = \left(\frac{1-\kappa_i}{1+\kappa_i}\right)^2 \quad \text{with} \quad \kappa_i = \text{cond}(\hat{A}_i^{-1}A_i).$$

Remark 3.4. The second inequality in (3.46) cannot become an equality except that

$$(\hat{A}_i^{-1}\nabla G_i(x_i^{(k)}),\, A_i^{(k)}p_i^{(k-1)}) = 0.$$

But this orthogonality does not hold, since we have only $(\nabla G_i(x_i^{(k)}),\, p_i^{(k-1)}) = 0$ but $\hat{A}_i \neq A_i^{(k)}$ in general. So (3.46) should be a strict inequality, and in turn the actual rate of convergence of Algorithm 3.2 is faster than that described by (3.39), i.e., the convergence rate of the steepest descent method (cf. [35]). Also, we remark that the exact line search is assumed in Algorithm 3.2. For the cases with inexact line searches, the relaxation parameter $\theta_i^{(k)}$ needs to be corrected somehow, and the discussion is much more technical (cf. [1], [19], [40]).

With the convergence rate estimate given by Lemma 3.3, the next lemma discusses how to meet the tolerance condition (3.18).

LEMMA 3.4. *For a given pair* $\{x_i, y_i\}$ *in* $R^n \times R^m$*, let* $\{x_i^{(k)}\}_{k=1}^{\infty}$ *be a sequence generated by Algorithm* 3.2 *for the minimization problem* (3.32)*. Then for any* $\delta_0 \in (0,1)$*, there exists an integer* k_0 *depending on* δ_0 *such that with* $x_{i+1} = x_i^{(k_0)}$*, the residual* $\varepsilon_i = F(x_{i+1}) - (f - By_i)$ *satisfies the tolerance condition* (3.18)*.*

Proof. Let x_i^* be the minimizer in (3.32) and the solution to (3.17). It follows from (3.8) that

$$\|\varepsilon_i\|_{A^{-1}}^2 = \|F(x_{i+1}) - F(x_i^*)\|_{A^{-1}}^2 \leq C_0 \|x_{i+1} - x_i^*\|_A^2. \tag{3.49}$$

By the mean-value theorem and the fact that $\nabla G_i(x_i^*) = 0$, we can write

$$\begin{aligned} G_i(x_{i+1}) - G_i(x_i^*) &= \int_0^1 (\nabla G_i(x_i^* + t(x_{i+1} - x_i^*)),\, x_{i+1} - x_i^*)dt \\ &= \int_0^1 (\nabla G_i(x_i^* + t(x_{i+1} - x_i^*)) - \nabla G_i(x_i^*),\, x_{i+1} - x_i^*)dt \\ &= \int_0^1 (F(x_i^* + t(x_{i+1} - x_i^*)) - F(x_i^*),\, x_{i+1} - x_i^*)dt. \end{aligned}$$

This, along with (3.7), leads to

$$G_i(x_{i+1}) - G_i(x_i^*) \geq c_0 \int_0^1 t\, \|x_{i+1} - x_i^*\|_A^2 dt = \frac{c_0}{2} \|x_{i+1} - x_i^*\|_A^2;$$

combining it with (3.49) and (3.39), we have

$$\|\varepsilon_i\|_{A^{-1}}^2 \leq \frac{2C_0}{c_0}(G_i(x_{i+1}) - G_i(x_i^*)) \leq \frac{2C_0\, \hat{\delta}_i}{c_0}(G_i(x_i) - G_i(x_i^*)) \tag{3.50}$$

with $\hat{\delta}_i = \prod_{k=1}^{k_0} \rho_i^{(k-1)}$. On the other hand, using the mean-value theorem and (3.8), we see

$$\begin{aligned} G_i(x_i) - G_i(x_i^*) &= \int_0^1 (\nabla G_i(x_i^* + t(x_i - x_i^*)),\, x_i - x_i^*)dt \\ &= \int_0^1 (F(x_i^* + t(x_i - x_i^*)) - F(x_i^*),\, x_i - x_i^*)dt \\ &\leq \frac{C_0}{2} \|x_i - x_i^*\|_A^2. \end{aligned} \tag{3.51}$$

But it follows from (3.7) that

$$\|x_i - x_i^*\|_A^2 \leq \frac{1}{c_0}(F(x_i) - F(x_i^*), x_i - x_i^*) \leq \frac{1}{c_0} \|F(x_i) - F(x_i^*)\|_{A^{-1}}\, \|x_i - x_i^*\|_A,$$

which implies that

$$\|x_i - x_i^*\|_A \le \frac{1}{c_0}\|F(x_i) - F(x_i^*)\|_{A^{-1}}.$$

This combined with (3.51) gives

$$G_i(x_i) - G_i(x_i^*) \le \frac{C_0}{2c_0^2}\|F(x_i) - F(x_i^*)\|^2_{A^{-1}}.$$

Now we obtain from (3.50) that

$$\|\varepsilon_i\|^2_{A^{-1}} \le \frac{\hat{\delta}_i C_0^2}{c_0^3}\|F(x_i) - F(x_i^*)\|^2_{A^{-1}} = \frac{\hat{\delta}_i C_0^2}{c_0^3}\|f_i\|^2_{A^{-1}},$$

which leads to the satisfaction of (3.18) when k_0 is chosen such that $\hat{\delta}_i \le \frac{c_0^3\delta_0^2}{C_0^2}$. □

3.5. An analysis on the convergence of Algorithm 3.1. We are now ready to study the convergence of Algorithm 3.1 and show Theorem 3.2. For this purpose, we need a few auxiliary lemmas. We remark that all notation below will be the same as in subsection 3.3. But for the sake of convenience, let us recall some frequently used notation here: $\{x, y\}$ is the exact solution to (1.1), A_ξ is a positive definite matrix defined by (3.6) at any given point $\xi \in R^n$, but with A_x simply denoted as A and A_{x_i} as A_i. The following are some error, or residual, vector quantities:

$$E_i = \{\sqrt{\delta_0}f_i, e_i^y\}, \quad f_i = f - F(x_i) - By_i, \quad e_i^x = x - x_i, \quad e_i^y = y - y_i.$$

The first lemma below gives conditions on the current approximation pair $\{x_i, y_i\}$ to ensure x_{i+1} lies in a specified neighborhood of x.

LEMMA 3.5. *Let r_ω be a fixed positive number, and let the pair $\{x_i, y_i\}$ be given such that $|||E_i||| \le \frac{c_0}{(1+\sqrt{\delta_0})}r_\omega$. If $x_{i+1} \in R^n$ is generated such that condition* (3.18) *holds for the residual $\varepsilon_i = F(x_{i+1}) - (f - By_i)$, then $x_{i+1} \in S_A(x, r_\omega)$.*

Proof. We know from the first equation of (1.1) that $f = F(x) + By$; hence

$$F(x_{i+1}) - F(x) = B(y - y_i) + \varepsilon_i. \tag{3.52}$$

But it follows from (3.7) that

$$\begin{aligned} c_0\|x_{i+1} - x\|_A^2 &\le (F(x_{i+1}) - F(x), x_{i+1} - x) \\ &= (B(y - y_i) + \varepsilon_i, x_{i+1} - x) \\ &\le \|B(y - y_i) + \varepsilon_i\|_{A^{-1}}\,\|x_{i+1} - x\|_A, \end{aligned}$$

which implies

$$\|x_{i+1} - x\|_A \le \frac{1}{c_0}\|B(y - y_i) + \varepsilon_i\|_{A^{-1}} \le \frac{1}{c_0}(\|B(y - y_i)\|_{A^{-1}} + \|\varepsilon_i\|_{A^{-1}}).$$

This, along with (3.18), leads to

$$\begin{aligned} \|x_{i+1} - x\|_A &\le \frac{1}{c_0}(\|B(y - y_i)\|_{A^{-1}} + \delta_0\|f_i\|_{A^{-1}}) \\ &= \frac{1}{c_0}(\|e_i^y\|_K + \sqrt{\delta_0}\|\sqrt{\delta_0}f_i\|_{A^{-1}}) \\ &\le \frac{(1 + \sqrt{\delta_0})}{c_0}|||E_i|||, \end{aligned} \tag{3.53}$$

which proves the desired result. □

The next lemma demonstrates that the matrix $A = A_x$ will be close to A_{i+1} as long as x_{i+1} stays close to x.

LEMMA 3.6. *For any given positive $\varepsilon < 1$, and any $x_{i+1} \in S_A(x, \varepsilon/(2L))$, we have*

$$\|(A - A_{i+1})\alpha\|_{A^{-1}} \leq \varepsilon \, \|\alpha\|_A \quad \forall \alpha \in R^n. \tag{3.54}$$

So all the eigenvalues of the matrix $A^{-1}A_{i+1}$ lie in the interval $[1-\varepsilon, 1+\varepsilon]$, and

$$\|\alpha\|_A \leq \frac{\|\alpha\|_{A_{i+1}}}{\sqrt{1-\varepsilon}} \quad \forall \alpha \in R^n. \tag{3.55}$$

Proof. Clearly (3.54) is invariant with respect to any constant scaling of α. Therefore it suffices to show that there is a number $\alpha_0 > 0$ such that (3.54) holds for all $\alpha \in R^n$ satisfying $\|\alpha\|_A \leq \alpha_0$. To show this, we rewrite $(A - A_{i+1})\alpha$ as

$$\begin{aligned}(A - A_{i+1})\alpha = {} & [A\alpha - (F(x+\alpha) - F(x))] + [F(x_{i+1}+\alpha) - F(x_{i+1}) - A_{i+1}\alpha] \\ & + [F(x+\alpha) - F(x) - (F(x_{i+1}+\alpha) - F(x_{i+1}))],\end{aligned}$$

then by the triangle inequality we have

$$\begin{aligned}\|(A - A_{i+1})\alpha\|_{A^{-1}} \leq {} & \|A\alpha - (F(x+\alpha) - F(x))\|_{A^{-1}} \\ & + \|F(x_{i+1}+\alpha) - F(x_{i+1}) - A_{i+1}\alpha\|_{A^{-1}} \\ & + \|F(x+\alpha) - F(x) - (F(x_{i+1}+\alpha) - F(x_{i+1}))\|_{A^{-1}}.\end{aligned} \tag{3.56}$$

But for any given positive $\varepsilon < 1$, we know from (3.6) that there is a positive number α_0 such that the following two estimates hold for any $\alpha \in R^n$ satisfying $\|\alpha\|_A \leq \alpha_0$,

$$\|A\alpha - (F(x+\alpha) - F(x))\|_{A^{-1}} \leq \frac{\varepsilon}{4} \, \|\alpha\|_A, \tag{3.57}$$

$$\|F(x_{i+1}+\alpha) - F(x_{i+1}) - A_{i+1}\alpha\|_{A^{-1}} \leq \frac{\varepsilon}{4} \, \|\alpha\|_A. \tag{3.58}$$

Let $G(\xi) = F(\xi+\alpha) - F(\xi)$. Clearly, the Lipschitzian property of F implies the same property for G. Thus by (3.3) there is a matrix $V \in \operatorname{co} \partial G(\overline{x_{i+1}x})$ such that

$$F(x+\alpha) - F(x) - (F(x_{i+1}+\alpha) - F(x_{i+1})) = G(x) - G(x_{i+1}) = V(x - x_{i+1});$$

this gives

$$\|F(x+\alpha) - F(x) - (F(x_{i+1}+\alpha) - F(x_{i+1}))\|_{A^{-1}} \leq \|A^{-\frac{1}{2}}VA^{-\frac{1}{2}}\| \, \|x - x_{i+1}\|_A. \tag{3.59}$$

As $G(\xi) = F(\xi+\alpha) - F(\xi)$, we have

$$\operatorname{co} \partial G(\overline{x_{i+1}x}) = \operatorname{co} \partial F(\alpha + \overline{x_{i+1}x}) - \operatorname{co} \partial F(\overline{x_{i+1}x}),$$

so it follows from (3.16) that

$$\|A^{-\frac{1}{2}}VA^{-\frac{1}{2}}\| \leq L \, \|\alpha\|_A.$$

This with (3.59) yields

$$\|F(x+\alpha) - F(x) - (F(x_{i+1}+\alpha) - F(x_{i+1}))\|_{A^{-1}} \leq L\|x - x_{i+1}\|_A \, \|\alpha\|_A.$$

Thus for any $x_{i+1} \in S_A(x, \varepsilon/(2L))$, we have

$$\|F(x+\alpha) - F(x) - (F(x_{i+1}+\alpha) - F(x_{i+1}))\|_{A^{-1}} \le \frac{\varepsilon}{2}\,\|\alpha\|_A;$$

this, along with (3.57) and (3.58), proves (3.54).

Now by writing (3.54) as

$$\|(I - A^{-1}A_{i+1})\alpha\|_A \le \varepsilon\,\|\alpha\|_A,$$

we see that the eigenvalues of $A^{-1}A_{i+1}$ must lie in the interval $[1-\varepsilon, 1+\varepsilon]$. This fact further implies

$$(A(A^{-1}A_{i+1})\alpha, \alpha) \ge (1-\varepsilon)(A\alpha, \alpha) \quad \forall\, \alpha \in R^n;$$

therefore,

$$\begin{aligned}(A\alpha, \alpha) &= (A_{i+1}\alpha, \alpha) + (A\alpha, \alpha) - (A(A^{-1}A_{i+1})\alpha, \alpha)\\ &\le (A_{i+1}\alpha, \alpha) + \varepsilon\,(A\alpha, \alpha),\end{aligned}$$

which leads to estimate (3.55). □

Remark 3.5. We see from the proof of Lemma 3.6 that estimate (3.54) is a direct consequence of (3.5) or (3.6). If F is smooth, then inequality (3.54) amounts to the condition that the gradient of F is Lipschitzian.

LEMMA 3.7. *For a given parameter $\gamma < \frac{1}{2}$, assume $x_{i+1} \in S_A(x, \varepsilon/(2L))$ with $\varepsilon < (1-2\gamma)/3$, and* (3.20) *is satisfied. Then* (3.24) *holds with $\delta_d < \frac{1}{2}$.*

Proof. For simplicity, we write $b = Bd_i$ and $\Psi(b) = \Psi_{A_{i+1}}(Bd_i)$. Then it suffices to verify

$$\|\Psi(b) - A^{-1}b\|_A \le \frac{1}{2}\|A^{-1}b\|_A \tag{3.60}$$

under the condition

$$\|\Psi(b) - A_{i+1}^{-1}b\|_{A_{i+1}} \le \gamma\,\|A_{i+1}^{-1}b\|_{A_{i+1}}. \tag{3.61}$$

To see this, first by the triangle inequality,

$$\|\Psi(b) - A^{-1}b\|_A \le \|\Psi(b) - A_{i+1}^{-1}b\|_A + \|A_{i+1}^{-1}b - A^{-1}b\|_A. \tag{3.62}$$

Using (3.55) and (3.61) above we derive

$$\|\Psi(b) - A_{i+1}^{-1}b\|_A \le \frac{\|\Psi(b) - A_{i+1}^{-1}b\|_{A_{i+1}}}{\sqrt{1-\varepsilon}} \le \frac{\gamma}{\sqrt{1-\varepsilon}}\|A_{i+1}^{-1}b\|_{A_{i+1}}. \tag{3.63}$$

On the other hand, it follows from (3.54) and (3.55) that

$$\|A_{i+1}^{-1}b - A^{-1}b\|_A = \|(A - A_{i+1})A_{i+1}^{-1}b\|_{A^{-1}} \le \varepsilon\,\|A_{i+1}^{-1}b\|_A \le \frac{\varepsilon}{\sqrt{1-\varepsilon}}\|A_{i+1}^{-1}b\|_{A_{i+1}}.$$

Substituting this and (3.63) into (3.62), and using Lemma 3.6 again, leads to

$$\begin{aligned}\|\Psi(b) - A^{-1}b\|_A &\le \frac{\gamma+\varepsilon}{\sqrt{1-\varepsilon}}\|A_{i+1}^{-1}b\|_{A_{i+1}} = \frac{\gamma+\varepsilon}{\sqrt{1-\varepsilon}}\|b\|_{A_{i+1}^{-1}}\\ &\le \frac{\gamma+\varepsilon}{1-\varepsilon}\|b\|_{A^{-1}} = \frac{\gamma+\varepsilon}{1-\varepsilon}\|A^{-1}b\|_A.\end{aligned} \tag{3.64}$$

This proves (3.60) by noting $\frac{\gamma+\varepsilon}{1-\varepsilon} < \frac{1}{2}$ when $\varepsilon < \frac{1-2\gamma}{3}$. □

Recalling that g_i, τ_i, and $\Psi_{H_i}(g_i)$ are the quantities used in Algorithm 3.1, that the parameter β_0 is defined in (3.25), and that $K = B^tA^{-1}B$ is the exact Schur complement with $A = A_x$, using (3.24) ensured by Lemma 3.7 we can derive the following lemma, which is basically the same as Lemma 3.1 in [29] (but with different notation).

LEMMA 3.8. *For a given parameter $\gamma < \frac{1}{2}$, assume $x_{i+1} \in S_A(x, \varepsilon/(2L))$ with $\varepsilon < (1-2\gamma)/3$, and* (3.20) *is satisfied. Then there is a symmetric and positive definite matrix Q_i such that*

(i) $Q_i^{-1} g_i = \frac{1}{2}\,\tau_i\,\Psi_{H_i}(g_i)$;

(ii) *all eigenvalues of the matrix $Q_i^{-1}K$ are in the interval* $[(1-\beta_0)/2, 1]$.

The following lemma is basically Lemma 3.5 in [28], with slight modifications.

LEMMA 3.9. *Let N be an $n\times n$ symmetric and positive semidefinite matrix, and let $\mathcal{F}(N)$ be a block matrix given by*

$$\mathcal{F}(N) = \begin{pmatrix} -\delta_0(I+N) & -\sqrt{\delta_0}N \\ -\sqrt{\delta_0}N & (I-N) \end{pmatrix}.$$

If all positive eigenvalues of N lie in the interval $[1-\frac{1+\beta_0}{2}, 1]$, then we have $\|\mathcal{F}(N)\| \le \rho_0$, where $\rho_0 < 1$ is defined in (3.26).

Proof of Theorem 3.2. With the previous technical preparations, we are now ready to demonstrate Theorem 3.1. First, we recall that ρ_0, ρ_ω, and r_ω^* are three parameters defined in (3.26) and (3.27). Then for Theorem 3.2 it suffices to prove that $\rho_0^* = \rho_0 + \rho_\omega < 1$, and the following relations hold for $i = 0, 1, 2, \ldots$:

$$|||E_{i+1}||| \le \rho_0^*|||E_i||| < |||E_i||| \le r_\omega^*. \tag{3.65}$$

We shall achieve this by induction. We start with the verification of this for $i = 0$. To do so, we shall first derive an error propagation equation. We know from (3.52) ($i = 0$) that

$$F(x_1) - F(x) = Be_0^y + \varepsilon_0. \tag{3.66}$$

But by the assumption of Theorem 3.2 on the initial guess $\{x_0, y_0\}$ and the definition of r_ω^*, we know $|||E_0||| \le r_\omega^* = c_0\hat{r}_\omega/(1+\sqrt{\delta_0})$ with $\hat{r}_\omega = \min\{r_\omega, r_\gamma\}$, so $x_1 \in S_A(x, \hat{r}_\omega)$ by Lemma 3.5, which implies $x_1 \in S_A(x, r_\gamma) \cap S_A(x, r_\omega)$. This enables us to apply Lemma 3.8(i) and Algorithm 3.1 to write

$$y_1 = y_0 + Q_0^{-1}B^t(x_1 - x). \tag{3.67}$$

With (3.66), we can further deduce

$$\begin{aligned} &A^{\frac{1}{2}}(x_1 - x) \\ &\quad= A^{-\frac{1}{2}}(F(x_1) - F(x)) - A^{-\frac{1}{2}}[F(x_1) - F(x) - A(x_1 - x)] \\ &\quad= A^{-\frac{1}{2}}(Be_0^y + \varepsilon_0) - \varphi_1, \end{aligned} \tag{3.68}$$

where $\varphi_1 = A^{-\frac{1}{2}}[F(x_1) - F(x) - A(x_1 - x)]$. Setting $N_0 = A^{-\frac{1}{2}}BQ_0^{-1}B^tA^{-\frac{1}{2}}$, we obtain from (3.67) and (3.68) that

$$\begin{aligned} A^{-\frac{1}{2}}Be_1^y &= A^{-\frac{1}{2}}Be_0^y - N_0[A^{-\frac{1}{2}}(Be_0^y + \varepsilon_0) - \varphi_1] \\ &= (I - N_0)A^{-\frac{1}{2}}Be_0^y - N_0A^{-\frac{1}{2}}\varepsilon_0 + N_0\varphi_1. \end{aligned} \tag{3.69}$$

Multiplying (3.66) by $A^{-\frac{1}{2}}$, we have

$$A^{-\frac{1}{2}}(F(x)-F(x_1)) = -A^{-\frac{1}{2}}Be_0^y - A^{-\frac{1}{2}}\varepsilon_0. \tag{3.70}$$

Then using the fact that

$$f_1 = f - F(x_1) - By_1 = F(x) - F(x_1) + Be_1^y,$$

we derive from (3.69) and (3.70) that

$$A^{-\frac{1}{2}}f_1 = -(I+N_0)A^{-\frac{1}{2}}\varepsilon_0 - N_0A^{-\frac{1}{2}}Be_0^y + N_0\varphi_1. \tag{3.71}$$

Now by defining for $k = 0, 1, 2, \ldots,$

$$E_k^y = A^{-\frac{1}{2}}Be_k^y, \quad E_k^{xy} = \sqrt{\delta_0}A^{-\frac{1}{2}}f_k, \quad e^{\varepsilon_k} = \sqrt{\delta_0}^{-1}A^{-\frac{1}{2}}\varepsilon_k,$$

we come to the following propagation equation using (3.69) and multiplying (3.71) by $\sqrt{\delta_0}$:

$$\begin{pmatrix} E_1^{xy} \\ E_1^y \end{pmatrix} = \mathcal{F}(N_0)\begin{pmatrix} e^{\varepsilon_0} \\ E_0^y \end{pmatrix} + \begin{pmatrix} \sqrt{\delta_0}N_0\,\varphi_1 \\ N_0\,\varphi_1 \end{pmatrix}. \tag{3.72}$$

We know from Lemma 3.8(ii) that all the positive eigenvalues of the matrix N_0 lie in the interval $[1-\frac{1+\beta_0}{2}, 1]$; thus $\|\mathcal{F}(N_0)\| \le \rho_0$ by Lemma 3.9. Then with the assumption of Theorem 3.2 on the tolerance ε_i for $i = 0, 1, 2, \ldots,$ we know (3.18) is satisfied, leading to

$$\|e^{\varepsilon_0}\| \le \|\sqrt{\delta_0}A^{-\frac{1}{2}}f_0\| = \|E_0^{xy}\|.$$

By this, with the definition of the norm $|||\cdot|||$, we see

$$\|e^{\varepsilon_0}\|^2 + \|E_0^y\|^2 \le |||E_0|||^2.$$

Using this and the bound $\|\mathcal{F}(N_0)\| \le \rho_0$, we derive from (3.72) that

$$|||E_1||| \le \rho_0\,|||E_0||| + \sqrt{1+\delta_0}\,\|\varphi_1\|. \tag{3.73}$$

Noting $x_1 \in S_A(x, r_\omega)$, it follows from (3.23), (3.53), and the definition of ρ_ω in (3.27) that

$$\begin{aligned}\sqrt{1+\delta_0}\,\|\varphi_1\| &= \sqrt{1+\delta_0}\,\|F(x_1)-F(x)-A(x_1-x)\|_{A^{-1}} \\ &\le \omega\sqrt{1+\delta_0}\,\|x_1-x\|_A \\ &\le \frac{\omega\sqrt{1+\delta_0}(1+\sqrt{\delta_0})}{c_0}|||E_0||| = \rho_\omega|||E_0|||.\end{aligned}$$

Then we know from (3.73) that

$$|||E_1||| \le (\rho_0+\rho_\omega)|||E_0||| = \rho_0^*|||E_0|||.$$

Noting the fact that ω is taken from the range $(0, \omega_0)$, we have

$$\rho_\omega = \frac{\omega\sqrt{1+\delta_0}(1+\sqrt{\delta_0})}{c_0} < \frac{\omega_0\sqrt{1+\delta_0}(1+\sqrt{\delta_0})}{c_0}.$$

This, with the definition of ω_0 in (3.27), shows $\rho_\omega < 1 - \rho_0$, so $\rho_0^* = \rho_\omega + \rho_0 < 1$, and

$$|||E_1||| \le \rho_0^* |||E_0||| < |||E_0||| \le r_\omega^*,$$

which verifies (3.65) for $i = 0$.

Now we assume (3.65) holds for $i = k - 1$ with any integer $k > 1$; then in exactly the same manner as for deriving the error propagation equation (3.72), we have

$$\begin{pmatrix} E_{k+1}^{xy} \\ E_{k+1}^{y} \end{pmatrix} = \mathcal{F}(N_k) \begin{pmatrix} e^{\varepsilon_k} \\ E_k^y \end{pmatrix} + \begin{pmatrix} \sqrt{\delta_0} N_k \, \varphi_{k+1} \\ N_k \, \varphi_{k+1} \end{pmatrix} \tag{3.74}$$

where $\varphi_{k+1} = A^{-\frac{1}{2}}[F(x_{k+1}) - F(x) - A(x_{k+1} - x)]$ and $N_k = A^{-\frac{1}{2}} B Q_k^{-1} B^t A^{-\frac{1}{2}}$. With this relation, one can follow exactly the same proof as for $i = 0$ above to verify that (3.65) holds for $i = k$. This completes the proof of (3.65) by induction. □

4. Numerical experiments.

In this section, we shall apply two new algorithms proposed in sections 2 and 3, Algorithms 2.2 and 3.1, and some other existing algorithms, to solve a linear saddle-point problem arising from a domain decomposition method with a Lagrange multiplier and a nonlinear saddle-point problem.

4.1. A linear saddle-point problem arising from a domain decomposition method with a Lagrange multiplier.

Domain decomposition methods with Lagrange multipliers have become popular in solving second order elliptic problems; see, for example, [8], [27], [34], and the references therein. This method allows nonmatching grids to be used in different subdomains, with Lagrange multipliers introduced to preserve necessary interface continuities between local solutions from neighboring subdomains. A domain decomposition method with a Lagrange multiplier results in a saddle-point system with respect to the primal variable and the Lagrange multiplier. Two different approaches are often used to solve the resulting saddle-point system: the first one eliminates the primal variable in the system and forms an interface equation for the multiplier, then solves the interface equation by a PCG method [26]; the second directly solves the saddle-point system by some preconditioned iterative method [27], [34]. We shall compare the efficiency of these two different approaches.

Consider the model elliptic problem

$$-\nabla \cdot (a \nabla u) = f \quad \text{in} \quad \Omega; \qquad u = g \quad \text{on} \quad \partial\Omega, \tag{4.1}$$

where Ω is a three-dimensional rectangular domain $\Omega = [0, 2] \times [0, 1]^2$. We decompose Ω into two subdomains Ω_1 and Ω_2: $\Omega_1 = [0, 1]^3$, $\Omega_2 = [1, 2] \times [0, 1]^2$, and then triangulate each subdomain Ω_k $(k = 1, 2)$ into smaller cubic elements, each with edges of equal length h_k. We remark that the two triangulations in Ω_1 and Ω_2, denoted $\mathcal{T}^{h_1}$ and $\mathcal{T}^{h_2}$, respectively, are not required to match on the interface $\Gamma = \overline{\Omega}_1 \cap \overline{\Omega}_2$. By $\mathcal{N}_{h_k}$ we denote the set of vertices of all elements in the triangulation of Ω_k, and $\Gamma_{h_k} = \Gamma \cap \mathcal{N}_{h_k}$ for $k = 1, 2$.

On each Ω_k, we define $V^h(\Omega_k) \subset H^1(\Omega_k)$ to be the standard Q_1 finite element space [17], [24], associated with the triangulation $\mathcal{T}^{h_k}$, and

$$V_g^h(\Omega_k) = \Big\{ v \in V^h(\Omega_k); \quad v(x_i) = g(x_i) \quad \forall\, x_i \in \mathcal{N}_{h_k} \cap (\partial\Omega_k \backslash \Gamma) \Big\},$$

$$V_g^h(\Omega) = \Big\{ v = \{v_1,\, v_2\} \in V_g^h(\Omega_1) \times V_g^h(\Omega_2); \quad v_1(x_i) = v_2(x_i) \quad \forall\, x_i \in \Gamma_{h_1} \Big\}.$$

Now the finite element approximation of the elliptic problem (4.1) can be formulated as follows: Find $\{u_{h_1}, u_{h_2}\} \in V_g^h(\Omega)$ such that

$$\sum_{k=1}^{2}(a\nabla u_{h_k}, \nabla v_k)_{\Omega_k} = \sum_{k=1}^{2}(f, v_k)_{\Omega_k} \quad \forall\, v = \{v_1, v_2\} \in V_0^h(\Omega). \tag{4.2}$$

By introducing a discrete Lagrange multiplier χ to remove the constraints on the interface as required in the finite element space $V_g^h(\Omega)$, system (4.2) can be written as the algebraic saddle-point system [27]

$$\begin{pmatrix} A_1 & 0 & B_1 \\ 0 & A_2 & B_2 \\ B_1^t & B_2^t & 0 \end{pmatrix} \begin{pmatrix} U_1 \\ U_2 \\ \chi \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ d \end{pmatrix}, \tag{4.3}$$

where A_1 and A_2 are the stiffness matrices associated with the bilinear form $(a\nabla\cdot, \nabla\cdot)_{\Omega_k}$ under the nodal basis of $V_0^{h_k}(\Omega_k)$, and U_k corresponds to the nodal values of u_{h_k} in $\Omega_k \cup \Gamma_{h_k}$.

By eliminating the variables U_1 and U_2 in (4.3), we obtain the interface equation [26]

$$K\chi = b \tag{4.4}$$

with

$$K = \sum_{k=1}^{2} B_k^t A_k^{-1} B_k, \quad b = \sum_{k=1}^{2} B_k^t A_k^{-1} b_k - d.$$

We note that the Schur complement K is a dense matrix, possibly of large size. The direct solver for system (4.4) is very expensive. Instead, we will consider the following three iterative methods for solving (4.3).

M1. Solve the interface equation (4.4) by the CG method. Recall that we are mainly interested in the case where no good preconditioners are available for the Schur complement, so CG is used here instead of PCG, although some effective preconditioners are available for the current Schur complement; see, e.g., [9], [21]. In fact, this main interest prompts us to choose the worst preconditioner, the identity, for the Schur complement K involved in all three algorithms we shall test.

M2. Solve the saddle-point system (4.3) directly by the well-known preconditioned MINRES method [39]. We will take an (algebraic) multigrid preconditioner (cf. [42]) to be the preconditioner $\hat{A}$ for the first 2×2 block matrix in the coefficient matrix of (4.3) and the identity matrix to be the preconditioner $\hat{K}$ for the Schur complement K.

M3. Solve the saddle-point system (4.3) directly by Algorithm 2.2 of section 2. The preconditioners $\hat{A}$ and $\hat{K}$ are taken to be the same as in **M2**. The approximation $\Psi_A(\phi)$ is taken to be $\hat{A}^{-1}\phi$ for any ϕ, while the approximation $\Psi_H(g_i)$ is generated by two PCG iterations for solving (2.13). (Note that PCG is the same as CG here since $\hat{K}$ is taken to be the identity.)

Table 4.1 shows the numerical results with M1, M2, and M3, where the coefficient $a(x, y, z)$ in (4.1) is taken to be $a(x, y, z) = 1 + xyz$, and functions f and g are taken so that the exact solution of (4.1) is $u(x, y, z) = (x + y + z)^{\frac{1}{5}}$. Considering the singularity

Table 4.1
Number of iterations and CPU time (in seconds).

h_1	M1		M2		M3 ($\theta = \frac{1}{2}$)		M3 ($\theta = 0.7$)	
	iter	CPU	iter	CPU	iter	CPU	iter	CPU
1/8	10	0.14	20	0.53	10	0.52	8	0.43
1/16	15	16.6	33	8.5	11	6.6	10	6.0
1/32	21	1956.9	58	158.5	13	89.9	12	86.3

of the solution at the origin, we take $h_1 = h_2/2$. In all the experiments, the initial guesses for the three iterative methods are taken to be zero, and the outer iterations terminate when the relative error of the residual reaches 10^{-5}. We mentioned in Remark 2.3 that the coefficient 1/2 in (2.18) can be replaced by a larger number than 1/2, which is now tested in Table 4.1 with two different choices of this coefficient, denoted by a parameter θ.

We remark that the actions of the Schur complement K involve the actions of the local solvers A_k^{-1}. These actions must be very accurate, since our target is to solve system (4.3) or (4.4). One may use both direct solvers or iterative solvers to realize the actions of these local solvers. But if iterative solvers are used, the stopping criterion should be up to a very high accuracy. So we have chosen in our experiments to realize these local solvers by the standard direct solver, a banded sparse version of the Gauss elimination.

From Table 4.1 we see that the new method (M3), Algorithm 2.2, outperforms M1 and M2 essentially, and this appears to be more evident when the discrete system becomes larger.

4.2. An algebraic nonlinear saddle-point problem. We now consider an algebraic nonlinear saddle-point problem and compare the convergence of the new Algorithm 3.1 of section 3 and the well-known augmented-Uzawa method, which has been widely used for nonlinear saddle-point systems.

Let I_m be the $m \times m$ identity matrix, and let T_m an $m \times m$ matrix with entries given by

$$t_{ij} = \begin{cases} 1 & \text{if } \ |i-j| = 1; \\ 0 & \text{otherwise.} \end{cases}$$

For $n = 2m$, we define an $n \times n$ symmetric positive definite matrix M and an $n \times m$ matrix B with full rank as follows:

$$M = \begin{pmatrix} \frac{5}{2}I_m - \frac{1}{4}T_m & -I_m \\ -I_m & \frac{5}{2}I_m - \frac{1}{4}T_m \end{pmatrix}, \quad B = (0, \ \ 2I_m - T_m)^t.$$

The smallest and largest eigenvalues of M are given by [14]

$$\begin{aligned} \lambda_1 &= 4\sin^2\frac{m}{2(n+m)}\pi + \sin^2\frac{1}{2(m+1)}\pi = 1 + \sin^2\frac{1}{2(m+1)}\pi, \\ \lambda_n &= 4\sin^2\frac{n}{2(n+m)}\pi + \sin^2\frac{m}{2(m+1)}\pi = 3 + \sin^2\frac{m}{2(m+1)}\pi. \end{aligned}$$

Now we define the nonlinear mapping F as

$$F(\xi) = M\xi + \frac{1}{5}\left(\frac{\xi_1}{1+\xi_1^2}, \frac{\xi_2}{1+\xi_2^2}, \ldots, \frac{\xi_n}{1+\xi_n^2}\right)^t \quad \forall\, \xi = (\xi_1 \ \xi_2 \cdots \xi_n)^t \in R^n. \tag{4.5}$$

One can verify that F is strongly monotone and Lipschitz continuous. Moreover, we have

$$\nabla F(\xi) = M + \frac{1}{5}\mathrm{diag}\left(\frac{1-\xi_1^2}{(1+\xi_1^2)^2}, \frac{1-\xi_2^2}{(1+\xi_2^2)^2}, \ldots, \frac{1-\xi_n^2}{(1+\xi_n^2)^2}\right),$$

which implies

$$\frac{4}{5}\|\zeta\|^2 \le (\nabla F(\xi)\zeta, \zeta) \le \frac{21}{5}\|\zeta\|^2 \quad \forall \zeta,\ \xi \in R^n. \tag{4.6}$$

Let $A_i = \nabla F(x_i)$. If we choose $\hat{A}_i = I_n$, we have $\mathrm{cond}(\hat{A}_i^{-1}A_i) \le 21/4$. Noting that the Schur complement $K_i = B^t A_i^{-1} B$ is a dense matrix without any special structure, it is difficult to find a reasonable preconditioner $\hat{K}_i$ for K_i. Therefore we will take the worst preconditioner $\hat{K}_i = I_m$.

The functional $J(\xi)$ satisfying $\nabla J(\xi) = F(\xi)$ can be written as

$$J(\xi) = \frac{1}{2}(M\xi, \xi) + \frac{1}{10}\sum_{l=1}^{n} \ln(1+\xi_l^2). \tag{4.7}$$

One can see that $J(\xi)$ is uniformly convex. This enables us to realize the nonlinear inner iteration in Algorithm 3.1 for finding x_{i+1} by Algorithm 3.2.

The right-hand side functions f and g in (1.1) are generated using system (1.1) when the exact solution is taken to be

$$x = (1, 1, \ldots, 1)^t, \quad y = \left(1, \frac{1}{2}, \ldots, \frac{1}{m}\right)^t.$$

We will compare the new nonlinear inexact Uzawa algorithm (Algorithm 3.1) with the well-known augmented-Uzawa method (see [38, pp. 234–244]). The augmented-Uzawa method converges, provided that the inner iteration involved is accurate enough, and the augmented parameter r is taken to be sufficiently large (or the initial guess is very close to the exact solution).

The initial guess x_0 for both Algorithm 3.1 and the augmented-Uzawa method will be taken to be the approximation generated by three steps of the steepest descent method for solving the perturbation system (3.30) with $\mu = 1/10$ and the zero initial guess. With x_0 available, y_0 is then determined from the second equation of (3.29). The iterations of Algorithm 3.1 and the augmented-Uzawa method terminate when

$$\varepsilon = \left\{\frac{\|f - F(x_i) - By_i\|^2 + \|g - B^t x_i\|^2}{\|f\|^2 + \|g\|^2}\right\}^{\frac{1}{2}} \le 10^{-5}. \tag{4.8}$$

We first apply Algorithm 3.1 for the nonlinear saddle-point problem (1.1) with the data described as above. There are three stopping parameters δ_0, δ_g, and γ involved in the inner iterations. For the convenience of numerical tests, we will replace the norms used in (3.18), (3.19), and (3.20) by the l^2-norms of the relative errors of the residuals, and take $\delta_0 = 1/4$ and $\gamma = 1/4$, but a few different choices for δ_g. The convergence results of Algorithm 3.1 are summarized in Table 4.2.

We then apply the augmented-Uzawa method for the nonlinear saddle-point problem (1.1) with the data described as above. Each nonlinear inner iteration involved in the augmented-Uzawa algorithm is realized by 30 or 40 iteration of Algorithm 3.2—a

TABLE 4.2
Number of iterations and CPU time (in seconds) with Algorithm 3.1.

$(n,\ m)$	(100, 50)		(200, 100)		(400, 200)		(800, 400)	
	iter	CPU	iter	CPU	iter	CPU	iter	CPU
$\delta_g = 1/4$	32	2.1	30	9.5	30	104.9	28	676.6
$\delta_g = 1/6$	29	2.2	31	14.4	29	87.3	27	635.6
$\delta_g = 1/8$	29	1.8	28	11.0	28	77.4	27	633.8

TABLE 4.3
*Number of iterations and CPU time with augmented-Uzawa algorithm (*30 *inner iterations).*

$(n,\ m)$	(100, 50)		(200, 100)		(400, 200)		(800, 400)	
	iter	CPU	iter	CPU	iter	CPU	iter	CPU
$r = 10$	255	85.5	180	267.8	139	987.7	106	7375.1
$r = 50$	62	20.8	49	73.1	40	284.9	31	2160.8
$r = 100$	64	21.6	61	90.8	51	361.8	40	2786.6

TABLE 4.4
*Number of iterations and CPU time with augmented-Uzawa algorithm (*40 *inner iteration).*

$(n,\ m)$	(100, 50)		(200, 100)		(400, 200)		(800, 400)	
	iter	CPU	iter	CPU	iter	CPU	iter	CPU
$r = 10$	255	115.5	179	388.1	138	1311.1	106	10553.7
$r = 50$	53	24.1	38	82.6	28	266.7	22	2195.9
$r = 100$	44	19.9	41	89.0	30	285.6	24	2406.2

TABLE 4.5
Number of iterations and CPU time with augmented-Uzawa algorithm ($\tilde{\varepsilon} \leq 10^{-2}$).

$(n,\ m)$	(100, 50)		(200, 100)		(400, 200)		(800, 400)	
	iter	CPU	iter	CPU	iter	CPU	iter	CPU
$r = 10$	255	97.0	179	354.8	138	1342.1	106	10269.6
$r = 50$	52	62.5	37	198.7	28	880.8	22	6249.0
$r = 100$	27	34.8	18	114.7	13	331.9	10	2356.4

preconditioned CG-type method. The numerical results are summarized in Table 4.3 (30 inner iterations) and Table 4.4 (40 inner iterations).

In order to better understand the effect of the inner iterations, we have also implemented stopping the inner iterations by the standard stopping criterion, which is set to stop the inner iterations when the relative residual is less than $\tilde{\varepsilon}$. The numerical results are summarized in Table 4.5.

Tables 4.3–4.5 indicate that both the convergence rate and the CPU time of the augmented-Uzawa algorithm depend strongly on the augmented parameter r. But there is still no theory about the selection of a reasonable or optimal parameter r.

One can see from Tables 4.2–4.5 that Algorithm 3.1 is evidently more efficient than the augmented-Uzawa method (even if the optimal parameter r may be found). For the augmented-Uzawa method, one can clearly see that it converges faster with more inner iterations, which, however, does not necessarily lead to less CPU times; see the figures in Table 4.3 and 4.4 with $r = 50$. When the inner iterations are set to be too accurate, the total CPU time may be much longer; compare the figures in Tables 4.4 and 4.5 with $r = 50$. But how to set the stopping criterion for the inner iterations of the augmented-Uzawa method is rather difficult and often quite problem dependent.

Acknowledgments. The authors wish to thank two anonymous referees for many constructive comments which led to great improvement in the results and the presentation of the paper.

REFERENCES

[1] M. Al-Baali and R. Fletcher, *On the order of convergence of preconditioned nonlinear conjugate gradient methods*, SIAM J. Sci. Comput., 17 (1996), pp. 658–665.

[2] K. Arrow, L. Hurwicz, and H. Uzawa, *Studies in Linear and Nonlinear Programming*, Stanford University Press, Stanford, CA, 1958.

[3] O. Axelsson, *Numerical algorithms for indefinite problems*, in Elliptic Problem Solvers, Academic Press, New York, 1984, pp. 219–232.

[4] R. Bank, B. Welfert, and H. Yserentant, *A class of iterative methods for solving saddle point problems*, Numer. Math., 56 (1990), pp. 645–666.

[5] A. Battermann and M. Heinkenschloss, *Preconditioners for Karush–Kuhn–Tucker matrices arising in the optimal control of distributed systems*, in Control and Estimation of Distributed Parameter Systems (Vorau, 1996), W. Desch, F. Kappel, and K. Kunisch, eds., Birkhäuser, Basel, 1998, pp. 15–32.

[6] A. Battermann and E. W. Sachs, *Block preconditioner for KKT systems in PDE-governed optimal control problems*, in Fast Solution of Discretized Optimization Problems, R. H. Hoppe, K.-H. Hoffmann, and V. Schulz, eds., Birkhäuser, Basel, 2001, pp. 1–18.

[7] A. Battermann and E. W. Sachs, *An Indefinite Preconditioner for KKT Systems Arising in Optimal Control Problems*, Technical Report, University Trier, Trier, Germany, 2004.

[8] F. Belgacem, *The mortar finite element method with Lagrange multipliers*, Numer. Math., 84 (1999), pp. 173–197.

[9] M. Bhardwaj, D. Day, C. Farhat, M. Lesoinne, K. Pierson, and D. Rixen, *Application of the FETI method to ASCI problems: Scalability results on one thousand processors and discussion of highly heterogeneous problems*, Internat. J. Numer. Methods Engrg., 47 (2000), pp. 513–535.

[10] J. H. Bramble, J. E. Pasciak, and A. T. Vassilev, *Analysis of the inexact Uzawa algorithm for saddle point problems*, SIAM J. Numer. Anal., 34 (1997), pp. 1072–1092.

[11] J. Bramble and J. Pasciak, *A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems*, Math. Comp., 50 (1988), pp. 1–18.

[12] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, New York, 1991.

[13] K. H. Chan, K. Zhang, J. Zou, and G. Schubert, *A non-linear, 3-D spherical α^2 dynamo using a finite element method*, Phys. Earth Planetary Int., 128 (2001), pp. 35–50.

[14] X. Chen, *Global and superlinear convergence of inexact Uzawa methods for saddle point problems with nondifferentiable mappings*, SIAM J. Numer. Anal., 35 (1998), pp. 1130–1148.

[15] Z. Chen, Q. Du, and J. Zou, *Finite element methods with matching and nonmatching meshes for Maxwell equations with discontinuous coefficients*, SIAM J. Numer. Anal., 37 (2000), pp. 1542–1570.

[16] Z. Chen and J. Zou, *An augmented Lagrangian method for identifying discontinuous parameters in elliptic systems*, SIAM J. Control Optim., 37 (1999), pp. 892–910.

[17] P. Ciarlet, *Basic error estimates for elliptic problems*, in Handbook of Numerical Analysis, Volume II, P. Ciarlet and J.-L. Lions, eds., North-Holland, Amsterdam, 1991, pp. 17–351.

[18] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley, New York, 1983.

[19] Y. Dai and Y. Yuan, *Nonlinear Conjugate Gradient Methods*, Shanghai Scientific and Technical Publishers, China, 2000 (in Chinese).

[20] H. C. Elman and G. H. Golub, *Inexact and preconditioned Uzawa algorithms for saddle point problems*, SIAM J. Numer. Anal., 31 (1994), pp. 1645–1661.

[21] C. Farhat and F.-X. Roux, *Implicit parallel processing in structural mechanics*, Comput. Mech. Adv., 2 (1994), pp. 1–124.

[22] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., John Wiley, Chichester, 1987.

[23] P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders, *Preconditioners for indefinite systems arising in optimization*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 292–311.

[24] V. Girault and P.-A. Raviart, *Finite Element Methods for Navier–Stokes Equations*, Springer-Verlag, Berlin, 1986.

[25] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, SIAM, Philadelphia, 1989.

[26] Q. Hu, G. Liang, and J. Liu, *A preconditioner for three dimensional domain decomposition methods with Lagrange multipliers*, J. Syst. Sci. Complex., 16 (2003), pp. 513–526.

[27] Q. Hu, Z. Shi, and D. Yu, *Efficient solvers for saddle-point problems arising from domain decompositions with Lagrange multipliers*, SIAM J. Numer. Anal., 42 (2004), pp. 905–933.

[28] Q. Hu and J. Zou, *An iterative method with variable relaxation parameters for saddle-point problems*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 317–338.

[29] Q. Hu and J. Zou, *Two new variants of nonlinear inexact Uzawa algorithms for saddle-point problems*, Numer. Math., 93 (2002), pp. 333–359.

[30] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, 1995.

[31] C. T. Kelley, *Iterative Methods for Optimization*, SIAM, Philadelphia, 1999.

[32] C. Keller, N. I. M. Gould, and A. J. Wathen, *Constraint preconditioning for indefinite linear systems*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1300–1317.

[33] Y. L. Keung and J. Zou, *An efficient linear solver for nonlinear parameter identification problems*, SIAM J. Sci. Comput., 22 (2000), pp. 1511–1526.

[34] A. Klawonn and O. B. Widlund, *A domain decomposition method with Lagrange multiplier and inexact solvers for linear elasticity*, SIAM J. Sci. Comput., 22 (2000), pp. 1199–1219.

[35] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, 1999.

[36] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Programming, 58 (1993), pp. 353–368.

[37] W. Queck, *The convergence factor of preconditioned algorithms of the Arrow–Hurwicz type*, SIAM J. Numer. Anal., 26 (1989), pp. 1016–1030.

[38] B. T. Polyak, *Introduction to Optimization*, Optimization Software, New York, 1987.

[39] T. Rusten and R. Winther, *A preconditioned iterative method for saddlepoint problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 887–904.

[40] J. Stoer and Y. Yuan, *A subspace study on conjugate algorithms*, Z. Angew. Math. Mech., 75 (1995), pp. 69–77.

[41] P. Tseng, *Applications of a splitting algorithm to decomposition in convex programming and variational inequalities*, SIAM J. Control Optim., 29 (1991), pp. 119–138.

[42] J. M. P. Vanek and M. Brezina, *Convergence of algebraic multigrid based on smoothed aggregation*, Numer. Math., 88 (2001), pp. 559–579.

SIAM J. Optim.
Vol. 16, No. 3, pp. 826–853

OPTIMAL PRICING POLICIES FOR PUBLIC TRANSPORTATION NETWORKS[*]

GIUSEPPE BUTTAZZO[†], ALDO PRATELLI[‡], AND EUGENE STEPANOV[†]

Abstract. In this paper we study the problem of finding an optimal pricing policy for the use of the public transportation network in a given populated area. The transportation network, modeled by a Borel set $\Sigma \subset \mathbb{R}^n$ of finite length, the densities of the population and of the services (or workplaces), modeled by the respective finite Borel measures φ_0 and φ_1, and the effective cost $A(t)$ for a citizen to cover a distance t without the use of the transportation network are assumed to be given. The pricing policy to be found is then a cost $B(t)$ to cover a distance t with the use of the transportation network (i.e., the "price of the ticket for a distance t"), and it has to provide an equilibrium between the needs of the population (hence minimizing the total cost of transportation of the population to the services/workplaces) and that of the owner of the transportation network (hence maximizing the total income of the latter). We present a model for such a choice and discuss the existence as well as some qualitative properties of the resulting optimal pricing policies.

Key words. optimal transportation, transportation network, Monge–Kantorovich problem, optimal pricing, Nash equilibrium

AMS subject classifications. 49Q20, 49N60, 49J35, 49N90

DOI. 10.1137/040619831

1. Introduction.

Suppose that a set $\Sigma \subset \Omega$ represents a public transportation network (say, a railway) in a city modeled by a convex set $\Omega \subset \mathbb{R}^n$, $n \geq 2$. If a path $\theta \subset \Omega$ connecting $x \in \Omega$ to $y \in \Omega$ is interpreted as the itinerary of a passenger moving from x to y, then $\mathcal{H}^1(\theta \setminus \Sigma)$ and $\mathcal{H}^1(\theta \cap \Sigma)$, where $\mathcal{H}^1$ stands for the one-dimensional Hausdorff measure, can be naturally interpreted as the "effective lengths" of the part of the itinerary which the passenger covers, respectively, outside of and along the public transportation network. Let a function A: $\mathbb{R}^+ \to \bar{\mathbb{R}}^+$ represent the effective cost (in terms of time, money, and/or fatigue) for a single citizen of traveling without using the public transportation network (i.e., "by own means"), while B: $\mathbb{R}^+ \to \bar{\mathbb{R}}^+$ represents the analogous cost of traveling with the use of such a network (i.e., "by train"). The former clearly characterizes the pattern of behavior of the population which we consider to be known, while the latter is a pricing policy on the use of the transportation network imposed by its owner. Under this assumption each single citizen moving along a path θ will choose the most convenient distance $L(B,\theta) \leq \mathcal{H}^1(\theta \cap \Sigma)$ to cover with the use of the transportation network (i.e., "by train"), and thus will cover $\mathcal{H}^1(\theta) - L(B,\theta)$ without using the transportation network (i.e., "by own means"). Then the number

$$\delta(B,\theta) := A(\mathcal{H}^1(\theta) - L(B,\theta)) + B(L(B,\theta))$$

represents the total cost for a citizen of traveling along θ. Suppose the densities of the residents φ_0 and of the services (or workplaces) φ_1 are given. The residents obviously travel to reach the services or workplaces and are free to use or not to use

[*] Received by the editors November 29, 2004; accepted for publication (in revised form) June 27, 2005; published electronically January 12, 2006.
http://www.siam.org/journals/siopt/16-3/61983.html

[†] Dipartimento di Matematica, Univeristà di Pisa, via Buonarroti 2, 56127 Pisa, Italy (buttazzo@dm.unipi.it, e.stepanov@sns.it).

[‡] Scuola Normale Superiore, Piazza dei Cavalieri 7, 56126 Pisa, Italy (a.pratelli@sns.it).

the transportation network. The choice of the way of transportation of the residents can be naturally described by the Monge–Kantorovich optimal transportation model (see, e.g., [1, 5, 6, 7, 8] for the detailed presentation of the respective theory).

The problem we study in the present paper is as follows. A company owning the public transportation network Σ would naturally be interested in using it in the most effective way, that is, to earn the maximum profit. The only tool at its disposal which can be used to accomplish this task is the pricing policy for the use of the network, i.e., the function B. However, the population reacts to each particular pricing policy B by choosing its own pattern of behavior, namely, the itineraries and maybe even the destinations of everyday movements. In particular, to maximize the profit, the owner of the public transportation network is tempted to increase B. But one can expect that the higher the latter is, the less people are interested in using the transportation network. For instance, if B is "extremely high" (of course depending on the other factors, such as the cost function A and the size of Ω), nobody will use the public transportation network, and hence such a "greedy" policy would make the company fail. Therefore, a more modest pricing policy is expected to be optimal. The paper is concerned with the existence and the qualitative properties of such an optimal pricing policy.

2. Notation and preliminaries.

In this paper the ambient space $\Omega \subset \mathbb{R}^n$, $n \geq 2$, representing the given city is assumed to be a convex body (i.e., a bounded closed convex set with nonempty interior) equipped with the Euclidean distance. The convexity assumption on Ω is mainly introduced to simplify the presentation. Instead, more general domains Ω can be considered equipped with the geodesic distance relative to Ω, which in the case of nonconvex Ω does not coincide with the Euclidean distance. The public transportation network in Ω is assumed to be represented by a Borel set $\Sigma \subset \Omega$ of finite length, i.e., $\mathcal{H}^1(\Sigma) < \infty$.

We call two Lipschitz-continuous paths $\hat{\theta}_1$, $\hat{\theta}_2$: $[0,1] \to \Omega$ equivalent if there is a continuous surjective increasing function (usually called "reparameterization") ϕ: $[0,1] \to [0,1]$ such that $\hat{\theta}_1(t) = \hat{\theta}_2(\phi(t))$ for all $t \in [0,1]$. Let Θ stand for the set of equivalence classes of Lipschitz-continuous paths. In this way each $\theta \in \Theta$ can be clearly identified with some directed rectifiable curve. In what follows we will frequently slightly abuse the language, identifying the elements of Θ (i.e., directed rectifiable curves) with their parameterizations (i.e., Lipschitz-continuous paths parameterizing such curves) when it cannot lead to a confusion. We consider the set Θ to be equipped with the distance

$$(2.1)\quad d_\Theta(\theta_1,\theta_2) := \inf\left\{\max_{t\in[0,1]} |\hat{\theta}_1(t) - \hat{\theta}_2(t)| \,:\, \hat{\theta}_i \text{ parameterization of } \theta_i, i = 1,2\right\},$$

where $|\cdot|$ is the Euclidean norm in $\mathbb{R}^n$. It is easy to see that $\theta_\nu \to \theta$ in Θ implies the Hausdorff convergence of the respective traces, though the converse is clearly not true.

If $\{\theta_1, \theta_2\} \subset \Theta$ are such that $\theta_1(1) = \theta_2(0)$, then we define $\theta_1 \circ \theta_2 \in \Theta$ the directed rectifiable curve possessing a parameterization

$$\theta_1 \circ \theta_2(t) := \begin{cases} \theta_1(2t) & \text{if } t \in [0,1/2], \\ \theta_2(2t-1) & \text{if } t \in (1/2,1]. \end{cases}$$

It is important to remark that every subset of curves in Θ having uniformly bounded length is compact. In fact, let a sequence $\{\theta_\nu\} \subset \Theta$ satisfy $\mathcal{H}^1(\theta_\nu) \leq l$ for some $l \geq 0$ and for all $\nu \in \mathbb{N}$. Then each θ_ν admits a parameterization with Lipschitz constant not exceeding l, and hence by the Ascoli–Arzelà theorem $\theta_\nu \to \theta$ in Θ for some $\theta \in \Theta$ and for a subsequence of ν (not relabeled).

For a locally compact metric space X we denote by $\mathcal{M}^+(X)$ the space of positive Radon measures over X equipped with the $*$-weak topology. If $\varphi \in \mathcal{M}^+(X)$ and $\mathscr{B}(X)$ stands for the Borel σ-algebra of X, then by $\mathscr{B}_\varphi(X)$ we denote the completion of the latter with respect to φ (in other words, $\mathscr{B}_\varphi(X)$ is generated by $\mathscr{B}(X) \cup \varphi^{-1}(\{0\})$, i.e., by Borel sets and φ-nullsets). Then a map $f\colon X \to Y$ is called φ-measurable, if it is measurable with respect to $\mathscr{B}_\varphi(X)$. If $f\colon X \to Y$ is a φ-measurable (in particular, Borel measurable) map between locally compact metric spaces X and Y, then we denote by $f_\# \varphi$ the *push-forward* of the measure φ, i.e., the measure defined by

$$(f_\# \varphi)(B) := \varphi(f^{-1}(B))$$

for every Borel $B \subset Y$.

We recall the basic facts about Γ-convergence. For a sequence of functionals $f_\nu\colon X \to \bar{\mathbb{R}}$, $\nu \in \mathbb{N}$, defined over a metric space X, we set

$$\begin{array}{rcl} (\Gamma^- \liminf_\nu f_\nu)(x) & := & \inf\{\liminf_\nu f_\nu(x_\nu) \,:\, x_\nu \to x\}, \\ (\Gamma^- \limsup_\nu f_\nu)(x) & := & \inf\{\limsup_\nu f_\nu(x_\nu) \,:\, x_\nu \to x\}, \end{array}$$

and we say that this sequence Γ^- converges to a functional $f\colon X \to \bar{\mathbb{R}}$ as $\nu \to \infty$, written $f := \Gamma^- \lim_\nu f_\nu$, if $f = \Gamma^- \liminf_\nu f_\nu = \Gamma^- \limsup_\nu f_\nu$. If $f_\nu = g$ for all $\nu \in \mathbb{N}$, then the functional $g^- := \Gamma^- \lim_\nu f_\nu$ is called the *relaxation* (or l.s.c. envelope) of the functional $g\colon X \to \bar{\mathbb{R}}$. Clearly, $\inf_X g = \inf_X g^-$. One also observes that Γ^--convergence is stable with respect to the passage to relaxations, namely,

$$\Gamma^- \lim_\nu f_\nu = \Gamma^- \lim_\nu f_\nu^-.$$

We also define

$$\Gamma^+ \limsup_\nu f_\nu := -\Gamma^- \liminf_\nu (-f_\nu), \qquad \Gamma^+ \liminf_\nu f_\nu := -\Gamma^- \limsup_\nu (-f_\nu),$$

and hence $f := \Gamma^+ \lim_\nu f_\nu$, if $f = \Gamma^+ \liminf_\nu f_\nu = \Gamma^+ \limsup_\nu f_\nu$. It is easy to show that a Γ^--limit (resp., Γ^+-limit) of an arbitrary sequence of functionals is a l.s.c. (resp., u.s.c.) function.

Finally, for the functional $f\colon X \to \bar{\mathbb{R}}$ defined over a metric space X, we denote by $\mathrm{Argmin}_X f$ (resp., $\mathrm{Argmax}_X f$) the set of minimizers (resp., maximizers) of f in X, i.e.,

$$\begin{array}{rcl} \mathrm{Argmin}_X f & := & \{x \in X \,:\, f(x) = \inf_X f\}, \\ \mathrm{Argmax}_X f & := & \{x \in X \,:\, f(x) = \sup_X f\}. \end{array}$$

3. Problem setting.

3.1. Behavior of the population. In our model the population is characterized by two nonnegative Borel measures φ_0 and φ_1 over Ω, the former modeling the density of the population, the latter modeling the density of the destinations of their everyday movements (say, the density of services or of workplaces), and by the function $A\colon \mathbb{R}^+ \to \bar{\mathbb{R}}^+$, which is interpreted as the cost per unit mass of moving without the use of public transportation network (i.e., by own means). We request that the total masses of φ_0 and φ_1 be equal, namely,

$$\varphi_0(\Omega) = \varphi_1(\Omega).$$

In what follows we will always assume that the function A is monotone nondecreasing and continuous. From the applicative point of view, it might be useful to think that $A(0) = 0$ (i.e., the "cost of not moving anywhere" is zero), although technically we will not need such an assumption for the validity of all the results proven in what follows.

In this section we propose two different models for the choice of the optimal pricing policy B, and consider some of their equivalent formulations. As for the requirements on B, it seems quite natural to request that B be monotone nondecreasing (i.e., the price of a short-distance trip should not exceed that of a long-distance one) and that $B(0) = 0$ (i.e., the owner of the transportation network may only charge for effective movements along the network and hence may not charge for "people staying at home"). Below we develop what we think is a reasonable model of behavior of both population and the owner of the transportation network under only the above assumptions on A and B. We will see that in this model there is no way to distinguish between the pricing policy B and its l.s.c. envelope B^-, and therefore later on we will be able to assume without loss of generality that B is l.s.c.

3.1.1. Individual behavior. We first consider the question of how each citizen chooses the distance $L(B,\theta)$ to cover with the use of the transportation network once his itinerary $\theta \in \Theta$ is chosen and supposing that the pricing policy B is a monotone nondecreasing function. Clearly, with $\theta \in \Theta$ fixed, once one chooses to cover the distance l with the use of the transportation network, one pays the cost

$$g(l) := A(\mathcal{H}^1(\theta) - l) + B(l).$$

Therefore, it is reasonable to choose for $L(B,\theta)$ a minimizer of the latter function over $[0, \mathcal{H}^1(\theta \cap \Sigma)]$. Note that such a minimizer might not exist when B is not l.s.c. However, the l.s.c. envelope g^- of the function g given by the formula

$$g^-(l) = A(\mathcal{H}^1(\theta) - l) + B^-(l),$$

where B^- stands for the l.s.c. envelope of B, attains it minimum value over $[0, \mathcal{H}^1(\theta \cap \Sigma)]$, which coincides with the infimum of g over the same interval. Note that if l minimizes g^- but not g, it means that in l the function B has a jump and $B(l) > \lim_{\xi \to l^-} B(\xi)$. In other words, in this case it is convenient for a citizen to cover along the transportation network any distance $\xi < l$ arbitrarily close to l, but not exactly l, while the cost of transportation along θ will be arbitrarily close to $g^-(l)$. It is natural then to define $L(B,\theta)$ to be one of the minimizers of g^- over $[0, \mathcal{H}^1(\theta \cap \Sigma)]$. It might happen, however, that a minimizer, of g^- is not unique, i.e., covering different distances with the use of the public transportation network along the same path gives an identical cost to the individual. We therefore make the following assumption: every person chooses to cover the minimum possible distance by his own means (or, equivalently, the maximum possible distance with the use of the public transportation network) as long as the total cost for him is the same. Translated in mathematical terms, this leads to the following definition: each person traveling along θ is covering on the transportation network the distance

$$L(B,\theta) := \max\{l \in \mathrm{Argmin}_{\xi \le \mathcal{H}^1(\theta\cap\Sigma)} A(\mathcal{H}^1(\theta) - \xi) + B^-(\xi)\}. \tag{3.1}$$

It is immediate to note that the maximum in the above definition is actually attained and hence the cost of transportation along θ is given by

$$\begin{aligned} \delta(B,\theta) &:= \min_{l \in [0,\mathcal{H}^1(\theta\cap\Sigma)]} (A(\mathcal{H}^1(\theta) - l) + B^-(l)) \\ &= A(\mathcal{H}^1(\theta) - L(B,\theta)) + B^-(L(B,\theta)). \end{aligned} \tag{3.2}$$

It is worth noting that $L(B,\cdot)$ is Borel measurable, and $\delta(B,\cdot)$ is l.s.c. as claimed in Lemma 3.1 below.

LEMMA 3.1. *For each monotone nondecreasing function B the function $L(B,\cdot)$: $\Theta \to \mathbb{R}^+$ is Borel measurable and the function $\delta(B,\cdot)$: $\Theta \to \mathbb{R}^+$ is l.s.c.*

Proof. Define

$$\begin{aligned} \alpha(s,t) &:= \max\big\{\sigma \in \operatorname{Argmin}\{A(s-\sigma)+B^-(\sigma) : 0 \le \sigma \le t\}\big\}, \\ \beta(s,t) &:= \min\{A(s-\sigma)+B^-(\sigma) : 0 \le \sigma \le t\}, \end{aligned}$$

so that of course $L(B,\theta) = \alpha(\mathcal{H}^1(\theta), \mathcal{H}^1(\theta\cap\Sigma))$ and $\delta(B,\theta) = \beta\big(\mathcal{H}^1(\theta), \mathcal{H}^1(\theta\cap\Sigma)\big)$. The function β is trivially monotone nondecreasing in s and monotone nonincreasing in t. On the other hand, the function α is u.s.c. Indeed, take two sequences $s_\nu \to s$ and $t_\nu \to t$ as $\nu \to \infty$ and suppose that $\sigma_\nu := \alpha(s_\nu, t_\nu) \to \bar\sigma$. For $\sigma < \bar\sigma$ one has $\sigma < \sigma_\nu \le t_\nu$ for sufficiently large ν, and then by definition $A(s_\nu - \sigma) + B^-(\sigma) \ge A(s_\nu - \sigma_\nu) + B^-(\sigma_\nu)$. Passing to the limit in ν and recalling that A is continuous and B^- l.s.c., one gets $A(s-\sigma)+B^-(\sigma) \ge A(s-\bar\sigma)+B^-(\bar\sigma)$, so that $\alpha(s,t) \ge \bar\sigma$. The thesis now follows because $\theta \mapsto \mathcal{H}^1(\theta)$ is l.s.c. by the Golab theorem (see Theorem 4.4.7 [2]) and $\theta \mapsto \mathcal{H}^1(\theta\cap\Sigma)$ is u.s.c. by Lemma 4.1. □

3.1.2. Collective behavior: Transport measures. Now we would like to study which itineraries are chosen by the people in their everyday movements. These itineraries will be modeled by a special measure $\eta \in \mathcal{M}^+(\Theta)$ which will say, roughly speaking, how many people are choosing each particular path. We will call a measure $\eta \in \mathcal{M}^+(\Theta)$ a *transport measure* (or simply a *transport*), if

$$p_{i\#}\eta = \varphi_i, \qquad i = 0,1,$$

where the map p_i: $\Theta \to \Omega$ is defined by $p_i(\theta) := \theta(i)$, $i = 0,1$. Define now the functional $C(B)$ over $\mathcal{M}^+(\Theta)$ by the formula

$$C(B)(\eta) := \int_\Theta \delta(B,\theta)\, d\eta(\theta).$$

We will show in Proposition 3.2 that $C(B)$ admits a minimizer η_{opt} over the set of all admissible transports. From a purely heuristic point of view, the optimal transport η_{opt} says, roughly speaking, how many people take each particular path θ in their everyday movements. Further on we denote by $E_{opt}(B)$ the set of all optimal transports η_{opt} for a given B, namely,

$$E_{opt}(B) := \operatorname{Argmin}\{C(B)(\eta) : \eta \text{ admissible transport}\}.$$

3.1.3. Equivalent formulation: Transport plans. The problem of minimizing C is in fact just a version of the classical Monge–Kantorovich transport problem. To see this, define the function d_B: $\Omega\times\Omega \to \bar{\mathbb{R}}$ by setting

$$d_B(x,y) := \inf\{\delta(B,\theta) : \theta \in \Theta, \theta(0) = x, \theta(1) = y\},$$

where $\delta(B,\theta)$ is given by (3.2). The number $d_B(x,y)$ can be naturally interpreted as the cost for a single person of traveling from x to y, once the opportunity of using the public transportation network is offered. Then the everyday movement of the population to their workplaces can be also modeled by a *transport plan* $\gamma \in \mathcal{M}^+(\Omega\times\Omega)$, i.e., a finite Borel measure over $\Omega\times\Omega$ satisfying

$$\pi_{i\#}\gamma = \varphi_i, \qquad i = 0,1, \tag{3.3}$$

where π_i: $\Omega \times \Omega \to \Omega$, $i = 0, 1$, stand for the projections on the first and the second factor, respectively, namely, $\pi_i(x_0, x_1) := x_i$. The term "transport plan" refers to the fact that in a very heuristic way one can think that $\gamma(x, y)$ stands for the number of people who move from point x to point y, and hence represents the "plan of transportation" of the population φ_0 to the destinations φ_1. The population chooses the transport plan so as to minimize over all admissible transport plans (i.e., measures $\gamma \in \mathcal{M}^+(\Omega \times \Omega)$ satisfying (3.3)) the total cost of everyday movement given by the Monge–Kantorovich functional I_B defined by the relationship

$$I_B(\gamma) := \int_{\Omega\times\Omega} d_B(x, y)\, d\gamma(x, y).$$

We will denote by $MK(B)$ the respective infimum of I_B. Namely, we set

$$MK(B) := \inf \{I_B(\gamma) \,:\, \gamma \text{ admissible transport plan}\}.$$

The latter infimum is attained at some optimal transport plan γ_{opt} since the set of admissible transport plans is obviously compact in $*$-weak topology of measures, while the integrand d_B is l.s.c. (in fact, even continuous), as will be shown in Proposition 5.3(i). We remark that the optimal transport plan γ_{opt} obviously may depend on B and characterizes, though not completely, the pattern of behavior of the population once the pricing policy B is known. Roughly speaking γ_{opt} says "who goes where" given the pricing policy B for the use of the public transportation network, though it does not say "along which path." Further on we denote by $\Gamma_{opt}(B)$ the set of all optimal transport plans γ_{opt} for a given B, namely,

$$\Gamma_{opt}(B) := \text{Argmin}\, \{I_B(\gamma) \,:\, \gamma \text{ admissible transport plan}\}.$$

Define the set of the "optimal paths"

$$\widetilde{\Theta}(B) := \{\theta \in \Theta \,:\, d_B(\theta(0), \theta(1)) = \delta(B, \theta)\}$$

and denote by $Q_B(x, y)$ the set of optimal paths connecting x with y, namely,

$$Q_B(x, y) = \{\theta \in \widetilde{\Theta}(B) \,:\, \theta(0) = x, \theta(1) = y\},$$

so that

$$\widetilde{\Theta}(B) = \bigcup_{(x,y)\in\Omega\times\Omega} Q_B(x, y).$$

A trivial by-product of Proposition 5.3(ii) is the fact that both $\widetilde{\Theta}(B)$ and $Q_B(x, y)$ are closed sets (as for the latter, one can also conclude immediately from Lemma 3.1 that each $Q_B(x, y)$ is closed as a set of minima of a l.s.c. function $\delta(B, \cdot)$ on a closed set).

The assertion below provides a relationship between the Monge–Kantorovich problem of finding an optimal transport plan and the problem of minimizing $C(B)$ over all transports.

Proposition 3.2. *If B is a monotone nondecreasing function one has*

$$MK(B) = \min \{C(B)(\eta) \,:\, \eta \textit{ admissible transport}\}. \tag{3.4}$$

Moreover, if $\eta_{opt} \in E_{opt}(B)$, then $\gamma := (p_0 \times p_1)_\# \eta_{opt} \in \Gamma_{opt}(B)$. Vice versa, there is a Borel map $q_B \colon \Omega \times \Omega \to \widetilde{\Theta}(B)$ such that whenever $\gamma_{opt} \in \Gamma_{opt}(B)$, $\eta := q_{B\#}\gamma_{opt} \in E_{opt}(B)$.

Finally, every $\eta_{opt} \in E_{opt}(B)$ is concentrated over $\widetilde{\Theta}(B)$, i.e., $\eta_{opt}(\Theta \backslash \widetilde{\Theta}(B)) = 0$.

Proof. If η is an admissible transport, then $\gamma := (p_0 \times p_1)_\# \eta$ is clearly an admissible transport plan. Hence, recalling the definition (3.2) of d_B, we have

$$\begin{aligned} C(B)(\eta) &= \int_\Theta \delta(B,\theta)\, d\eta \quad \geq \quad \int_\Theta d_B((p_0 \times p_1)(\theta))\, d\eta \\ &= \int_{\Omega\times\Omega} d_B(x,y)\, d\gamma \quad = \quad I_B(\gamma) \quad \geq \quad MK(B). \end{aligned} \tag{3.5}$$

On the other hand, there is a Borel measurable selector $q_B \colon \Omega \times \Omega \to \widetilde{\Theta}(B)$ of the multivalued map $Q_B \colon \Omega \times \Omega \multimap \widetilde{\Theta}(B)$ (we use the symbol $\multimap$ instead of an arrow to stress that Q_B is multivalued). To verify this claim, it suffices to observe that the graph of Q_B defined by

$$\begin{aligned} \text{Graph}\, Q_B &:= \{(x,y,\theta) : (x,y) \in \Omega \times \Omega, \theta \in Q_B(x,y)\} \\ &= \{(x,y,\theta) \in \Omega \times \Omega \times \Theta : x = \theta(0), y = \theta(1)\} \cap (\Omega \times \Omega \times \widetilde{\Theta}(B)) \end{aligned}$$

is closed because $\widetilde{\Theta}(B)$ is closed (in view of Proposition 5.3(ii)), while $Q_B(x,y)$ is nonempty for each $(x,y) \in \Omega \times \Omega$, and we refer to the Kuratowski–Ryll–Nardzewski measurable selection theorem (Theorem III.6 [4] or Theorem 5.2.1 [9]). Then, given an arbitrary optimal transport plan $\gamma_{opt} \in \Gamma_{opt}(B)$ we observe that $\eta_{opt} := q_{B\#}\gamma_{opt}$ is an admissible transport. Moreover,

$$\begin{aligned} MK(B) &= \int_{\Omega\times\Omega} d_B(x,y)\, d\gamma_{opt} \quad = \quad \int_{\Omega\times\Omega} \delta(B, q_B(x,y))\, d\gamma_{opt} \\ &= \int_\Theta \delta(B,\theta)\, d\eta_{opt} \quad = \quad C(B)(\eta_{opt}) \quad \geq \quad \inf C(B), \end{aligned} \tag{3.6}$$

where the infimum of $C(B)$ is taken over the set of all admissible transports. Together with (3.5) this proves that η_{opt} is an optimal transport and hence the validity of the first claim of the statement is also proven. From (3.5) with η optimal transport, we get that $(p_0 \times p_1)_\# \eta$ is an optimal transport plan. Finally, since in (3.5) all the inequalities are actually equalities when η is an optimal transport, we get the validity of the last claim. □

We now introduce another auxiliary construction to be used in what follows. Define

$$\lambda := \operatorname{diam}\Omega + \mathcal{H}^1(\Sigma) \text{ and } \Theta' := \{\theta \in \Theta : \mathcal{H}^1(\theta) \leq \lambda\}, \tag{3.7}$$

so that clearly $\Theta' \subset \Theta$ is compact. We prove now the following very easy estimates.

LEMMA 3.3. *For every $(x,y) \in \Omega \times \Omega$ one has*

$$d_B(x,y) \leq A(\operatorname{diam}\Omega).$$

Further, if $\theta \in \widetilde{\Theta}(B) \setminus \Theta'$, then $B^-(L(B,\theta)) = 0$.

Proof. Let $[x,y] \in \Theta$ stand for the line segment with endpoints $x \in \Omega$ and $y \in \Omega$. One has therefore

$$d_B(x,y) \leq \delta(B,[x,y]) \leq A(\mathcal{H}^1([x,y])) = A(|x-y|) \leq A(\operatorname{diam}\Omega),$$

since $|x-y| \le \operatorname{diam}\Omega$ and A is monotone nondecreasing. Hence, the first claim is proven.

For $\theta \in \widetilde{\Theta}(B)$ we just proved

$$\delta(B,\theta) = A(\mathcal{H}^1(\theta) - L(B,\theta)) + B^-(L(B,\theta)) \le A(\operatorname{diam}\Omega).$$

If, moreover, $\mathcal{H}^1(\theta) \ge \lambda$, then minding $L(B,\theta) \le \mathcal{H}^1(\theta \cap \Sigma) \le \mathcal{H}^1(\Sigma)$ we get

$$A(\mathcal{H}^1(\theta) - L(B,\theta)) \ge A(\lambda - \mathcal{H}^1(\Sigma)) = A(\operatorname{diam}\Omega),$$

and hence $B^-(L(B,\theta)) = 0$, concluding the proof. □

Remark. If A is either strictly increasing or unbounded (i.e., $A(l) \to \infty$ as $l \to +\infty$), then it easily follows from Lemma 3.3 that all the paths in $\widetilde{\Theta}(B)$ must have uniformly bounded length (independent of B). In fact, for all $\theta \in \widetilde{\Theta}(B)$ one has $\delta(B,\theta) = A(\mathcal{H}^1(\theta) - L(B,\theta)) + B^-(L(B,\theta)) \le A(\operatorname{diam}\Omega)$; hence

$$A(\mathcal{H}^1(\theta) - \mathcal{H}^1(\Sigma)) \le A(\mathcal{H}^1(\theta) - L(B,\theta)) \le A(\operatorname{diam}\Omega),$$

which implies that $\mathcal{H}^1(\theta) \le l$ for some $l > 0$ independent of both θ and B.

3.2. Income of the owner of the transportation network. We have to calculate now how much each person pays to the owner of the public transportation network.

Consider first the case when one knows only the transport plan γ. We assume then that people moving from x to y choose an optimal path $\theta \in \widetilde{\Theta}(B)$ connecting those two points, and thus we need to calculate how much of this path is actually covered by using the public transportation network. It can happen, however, that there are many optimal paths connecting the same couple of points. We assume, then in line with the assumption made when defining L, that every person chooses the maximum possible distance to cover via the transportation network as long as the total cost for him is the same, and hence covers the distance

$$\Lambda(B,x,y) := \sup_{\theta \in Q_B(x,y)} L(B,\theta) \tag{3.8}$$

along the transportation network. Under this assumption the income of the owner of the transportation network becomes

$$\hat{G}(B,\gamma) := \int_{\Omega\times\Omega} B^-(\Lambda(B,x,y))\, d\gamma(x,y).$$

Note that we integrate $B^-(\Lambda)$ rather than $B(\Lambda)$ because, if B is not l.s.c., for people it might be more convenient to use the transportation network for the distance just slightly lower than Λ.

Since $\Lambda(B,\cdot,\cdot)$ is a Borel function as we show below, then the above integral is well defined.

LEMMA 3.4. *For each monotone nondecreasing function B the function $\Lambda(B,\cdot,\cdot)$: $\Omega\times\Omega \to \mathbb{R}^+$ is Borel measurable.*

Proof. We have

$$\Lambda(B,x,y) = \sup_{k\in\mathbb{N}} \Lambda^k(B,x,y), \text{ where } \Lambda^k(B,x,y) := \sup_{\theta\in Q_B(x,y),\mathcal{H}^1(\theta)\le k} L(B,\theta).$$

Let $(x_\nu, y_\nu) \to (x,y)$ in $\Omega \times \Omega$ as $\nu \to \infty$, and let $\theta_\nu \in Q_B(x_\nu, y_\nu)$ be such that

$$L(B,\theta_\nu) \geq \Lambda^k(B, x_\nu, y_\nu) - 1/\nu \text{ and } \mathcal{H}^1(\theta_\nu) \leq k$$

for all $\nu \in \mathbb{N}$. Then up to a subsequence (not relabeled) $\theta_\nu \to \theta$ as $\nu \to \infty$ and $\theta \in Q_B(x,y)$ according to Proposition 5.3(ii). Therefore, minding that $L(B,\cdot)$ is u.s.c. on $\widetilde{\Theta}(B)$ as a consequence of Lemma 5.7, we get

$$\Lambda^k(B,x,y) \geq L(B,\theta) \geq \limsup_\nu L(B,\theta_\nu) \geq \limsup_\nu \Lambda^k(B,x_\nu,y_\nu).$$

This means that each $\Lambda^k(B,\cdot,\cdot)$ is u.s.c., which suffices to conclude the proof. □

Remark. If A is either strictly increasing or unbounded, then by the remark following Lemma 3.3 one has that all paths in $\widetilde{\Theta}(B)$ have uniformly bounded length. Hence, using the notation of the above proof, $\Lambda = \Lambda^k$ for some $k \in \mathbb{N}$ sufficiently large, and hence $\Lambda(B,\cdot,\cdot)$ is u.s.c. Since in this case each $Q_B(x,y)$ is compact, then in fact the supremum in the definition (3.8) of Λ is attained (i.e., is a maximum).

Now consider the more particular case when the movement of the population is described by some optimal transport $\eta \in E_{opt}(B)$. Then the income of the owner of the transportation network can be expressed as

$$\hat{F}(B,\eta) := \int_\Theta B^-(L(B,\theta))\, d\eta(\theta)$$

(again, we integrate $B^-(L)$ rather than $B(L)$ because if B is not l.s.c. it might be more convenient for people to use the transportation network for a distance just slightly lower than L).

3.3. How to choose an optimal pricing policy.

Below we assume the public transportation network Σ and the function A (the cost of moving "by own means") to be given, and will be interested in the optimal choice of B. We note that all the functionals introduced actually depend on B^- rather than on B. Thus, further on we will define the class $\mathcal{B}$ of admissible pricing policies $B\colon \mathbb{R}^+ \to \mathbb{R}^+$ which are the monotone nondecreasing l.s.c. functions satisfying $B(0) = 0$.

The choice of the optimality criterion for B leads to two different models.

3.3.1. Short-term optimality.

The first and simplest model suggests that we choose a B as a maximizer of $\hat{G}(\cdot,\gamma)$ once the transport plan γ describing the everyday movement of the population is known. Namely, the problem reads as follows.

Problem 1. *Given a transport plan γ, find a $B \in \mathcal{B}$ maximizing $\hat{G}(\cdot,\gamma)$ over $\mathcal{B}$.*

Such a model refers to the situation when the pattern of behavior of the population in terms of "who goes where" is independent of the choice of the pricing policy B and therefore is simply initially fixed. It reasonably suits the company which is planning a short-term profit: within a short period of time the movements of the people are unlikely to change with the change of the pricing policy for the transportation network. We remark that in such a case, though the sources and destinations of movement of the population (given by γ) are known and independent of the pricing policy B, the actual itineraries (described by a transport η) might depend on B.

The statement below provides an equivalent formulation of Problem 1.

Proposition 3.5. *One has*

$$\begin{aligned}&\sup\{\hat{G}(B,\gamma) \,:\, B \in \mathcal{B}\}\\ &\qquad = \sup\Big\{\hat{F}(B,\eta) \,:\, B \in \mathcal{B}, \eta \in \operatorname{Argmin}\{C(B)(\eta') \,:\, (p_0 \times p_1)_\# \eta' = \gamma\}\Big\}.\end{aligned}$$

Moreover, if one of the above suprema is attained in $B \in \mathcal{B}$, then so is the other one.

Proof. Let $\eta \in \operatorname{Argmin}\{C(B)(\eta') \,:\, (p_0 \times p_1)_\# \eta' = \gamma\}$. Then η is concentrated on $\widetilde{\Theta}(B)$. In fact, otherwise, minding that $\delta(B,\theta) > d_B((p_0 \times p_1)(\theta))$ over $\Theta \setminus \widetilde{\Theta}(B)$, we get $C(\eta) > I_B(\gamma)$ by (3.5). But by (3.6) we have $I_B(\gamma) = C(q_{B\#}\gamma)$, where q_B is defined as in Proposition 3.2. Since by construction $(p_0 \times p_1)_\#(q_{B\#}\gamma) = \gamma$, this contradicts the minimality of η.

Since η is concentrated in $\tilde{\Theta}(B)$, and keeping in mind that $(p_0 \times p_1)_\#\eta = \gamma$, one has

$$\begin{aligned} \hat{F}(B,\eta) &= \int_\Theta B(L(B,\theta))\, d\eta(\theta) \\ &\le \int_\Theta B(\Lambda(B,\theta(0),\theta(1)))\, d\eta(\theta) \\ &= \int_{\Omega\times\Omega} B(\Lambda(B,x,y))\, d\gamma(x,y) \quad = \quad \hat{G}(B,\gamma). \end{aligned} \tag{3.9}$$

On the other hand, define $\widehat{\Theta}(B) \subset \widetilde{\Theta}(B)$ and $\widehat{Q}_B(x,y) \subset Q_B(x,y)$ as follows:

$$\begin{aligned} \widehat{\Theta}(B) &:= \{\theta \in \widetilde{\Theta}(B) \,:\, L(B,\theta) = \Lambda(B,\theta(0),\theta(1))\}, \\ \widehat{Q}_B(x,y) &:= \{\theta \in \widehat{\Theta}(B) \,:\, \theta(0) = x,\, \theta(1) = y\}. \end{aligned}$$

We observe that $\widehat{\Theta}(B)$ is the set where the u.s.c. function $\theta \in \widetilde{\Theta}(B) \mapsto L(B,\theta)$ equals the Borel function $\theta \in \widetilde{\Theta}(B) \mapsto \Lambda(B,\theta(0),\theta(1))$. Hence, $\widehat{\Theta}(B)$ is a Borel set. Then

$$\begin{aligned} \operatorname{Graph} \widehat{Q}_B &:= \left\{(x,y,\theta) \,:\, (x,y) \in \Omega\times\Omega, \theta \in \widehat{Q}_B(x,y)\right\} \\ &= \{(x,y,\theta) \,:\, \theta(0) = x, \theta(1) = y\} \cap (\Omega\times\Omega\times\widehat{\Theta}(B)) \end{aligned}$$

is also a Borel set since the set $\{(x,y,\theta) \,:\, \theta(0) = x, \theta(1) = y\}$ is closed. By the von Neumann–Aumann measurable selection theorem (Theorems III.222 and III.23 of [4], or, equivalently, Corollary 5.5.8 [9]) one has

$$\Delta := \{(x,y) \in \Omega\times\Omega \,:\, \widehat{Q}_B(x,y) \neq \emptyset\} \in \mathscr{B}_\gamma,$$

and, moreover, there is a γ-measurable (but not necessarily Borel measurable) selection $\hat{q}_{B,\gamma}\colon \Delta \to \widehat{\Theta}(B)$.

We show now that for every couple $(x,y) \notin \Delta$ one has

$$B(\Lambda(B,x,y)) = 0. \tag{3.10}$$

In fact, if $\{\theta_\nu\} \subset Q_B(x,y)$ is a maximizing sequence for $L(B,\cdot)$ (i.e., $L(B,\theta_\nu) \uparrow \Lambda(B,x,y)$ as $\nu \to \infty$), then necessarily $\mathcal{H}^1(\theta_\nu) \ge \lambda$ for all sufficiently large ν. Otherwise, up to a subsequence, one would have $\theta_\nu \to \theta$ as $\nu \to \infty$, and due to the fact that $L(B,\cdot)$ is u.s.c. on $\tilde{\Theta}(B)$ one would have $L(B,\theta) = \Lambda(B,x,y)$, contrary to the assumption $(x,y) \notin \Delta$. By Lemma 3.3 for all sufficiently large ν, we have $B(L(B,\theta_\nu)) = 0$. Minding that B is l.s.c. and monotone nondecreasing, we therefore get (3.10).

Consider now a $\hat{\theta} \in \Theta$ such that $\mathcal{H}^1(\hat{\theta}) > \lambda$. For every $(x,y) \in \Omega\times\Omega \setminus \Delta$ we set

$$\hat{q}_{B,\gamma}(x,y) := q_B(x,y),$$

where $q_B\colon\ \Omega\times\Omega\to\widetilde{\Theta}(B)$ is defined in Proposition 3.2. Clearly, for such (x,y), according to (3.10) one has

$$B(L(B,\hat q_{B,\gamma}(x,y)))\geq 0=B(\Lambda(B,x,y)).$$

Hence, due to the definition of $\hat q_{B,\gamma}$, the above relationship is satisfied for all $(x,y)\in\Omega\times\Omega$.

Minding that $\hat q_{B,\gamma}$ is clearly γ-measurable, we define $\eta_\gamma:=(\hat q_{B,\gamma})_{\#}\gamma$, getting then

$$\begin{array}{rcll}\hat F(B,\eta_\gamma) & = & \displaystyle\int_\Theta B(L(B,\theta))\,d\eta_\gamma(\theta) & \\ & = & \displaystyle\int_{\Omega\times\Omega} B(L(B,\hat q_{B,\gamma}(x,y)))\,d\gamma(x,y) & \\ & \geq & \displaystyle\int_{\Omega\times\Omega} B(\Lambda(B,x,y))\,d\gamma(x,y) & = \quad \hat G(B,\gamma).\end{array}$$

Together with (3.9) this concludes the proof. □

3.3.2. Long-term optimality. Another and more complex problem refers to a quite different situation when the pattern of behavior of the population (i.e., the transport plan γ) is a priori unknown and is chosen by the population so as to minimize its own travel costs. In other words, γ is chosen among optimal transport plans for the given pricing policy B, i.e., $\gamma\in\Gamma_{opt}(B)$, or, equivalently, the transport η is chosen among optimal transports for B, i.e., $\eta\in E_{opt}(B)$. To state formally the respective problem, we define

$$F(B,\eta):=\begin{cases}\hat F(B,\eta) & \text{if } \eta\in E_{opt}(B),\\ -\infty & \text{otherwise.}\end{cases}$$

The problem now reads as follows.

Problem 2. *Find a $B\in\mathcal{B}$ such that for some transport η the pair (B,η) maximizes F among all functions $B\in\mathcal{B}$ and admissible transports.*

This problem is well suited for a company looking for a long-term profit. In fact, in a long-term perspective, the directions of the population movements are adjusted to the pricing policy of the transportation network.

The latter problem admits clearly an equivalent formulation. Namely, set

$$G(B,\gamma):=\begin{cases}\hat G(B,\gamma) & \text{if } \gamma\in\Gamma_{opt}(B),\\ -\infty & \text{otherwise.}\end{cases}$$

Problem 2 is then equivalent to that of finding a $B\colon\ \mathbb{R}^+\to\bar{\mathbb{R}}^+$ maximizing $G(B,\gamma)$ among all functions $B\in\mathcal{B}$, where γ varies among all admissible transport plans, as stated in the following assertion.

Proposition 3.6. *One has*

$$\sup F=\sup G.$$

Moreover, if $(B',\eta')\in\operatorname{Argmax}F$, then by setting $\gamma':=(p_0\times p_1)_{\#}\eta'$ one has $(B',\gamma')\in\operatorname{Argmax}G$. Vice versa, if $(B',\gamma')\in\operatorname{Argmax}G$, then there is an admissible transport η', such that $(B',\eta')\in\operatorname{Argmax}F$.

Proof. With the use of Proposition 3.5, we get from (3.9) that whenever $\eta \in E_{opt}(B)$, we have $\hat{F}(B,\eta) \leq \hat{G}(B,(p_0 \times p_1)_{\#}\eta)$ and hence, recalling that $(p_0 \times p_1)_{\#}\eta \in \Gamma_{opt}(B)$ by Proposition 3.2, we get

$$\hat{F}(B,\eta) \leq \sup\{\hat{G}(B,\gamma)\,:\,\gamma \in \Gamma_{opt}(B)\}. \tag{3.11}$$

On the other hand, for a transport plan γ, the proof of Proposition 3.5 provides the existence of an admissible transport η_γ such that

$$\hat{F}(B,\eta_\gamma) \geq \hat{G}(B,\gamma). \tag{3.12}$$

It is immediate to verify by construction of η_γ that $\eta_\gamma \in E_{opt}(B)$ whenever $\gamma \in \Gamma_{opt}(B)$, because the map $\hat{q}_{B,\gamma}$ is a selector of the multivalued map Q_B. Now, (3.11) and (3.12) together show the statement. □

3.4. Game theoretic interpretation. Problem 2 admits also an obvious game theoretic interpretation. Namely, consider a game between two players: the "population" P (users of the transportation network) and the "service provider" S (the owner of the network). The set of strategies of the former is the set of admissible transport plans γ (equivalently, the set of all admissible transports η), while the set of strategies of the latter is $\mathcal{B}$. The payoff function of the population K_P is given by the formula $K_P(B,\gamma) := -I_B(\gamma)$ or, equivalently, $K_P(B,\eta) := -C(B)(\eta)$, while the payoff function of the service provider K_S is given by $K_S(B,\gamma) := G(B,\gamma)$ (equivalently, $K_S(B,\eta) := F(B,\eta)$). It is immediate to observe that when (B,γ) is a maximizing pair of the functional G (equivalently, (B,η) is a maximizing pair of the functional F), so that B solves Problem 2, then (B,γ) (equivalently, (B,η)) is a Nash equilibrium point of the above game (i.e., a pair of strategies, one for each player, such that no player has incentive to unilaterally change his action); moreover, it is an equilibrium point which is "most convenient" for the owner of the transportation network.

4. Auxiliary lemmata. In this section we collect a couple of auxiliary statements to be used in what follows.

We start with the following simple assertion.

Lemma 4.1. *Let $\mu \in \mathcal{M}^+(\Omega)$ be a finite Borel measure, and for the sequence of compact sets $\{K_\nu\}$, $K_\nu \subset \Omega$ one has $K_\nu \to K$ in the sense of Hausdorff for some $K \subset \Omega$. Then $\mu(K) \geq \limsup_\nu \mu(K_\nu)$. In particular, if $\Sigma \subset \Omega$ is a Borel set satisfying $\mathcal{H}^1(\Sigma) < \infty$, then putting $\mu := \mathcal{H}^1 \llcorner \Sigma$ one obtains*

$$\mathcal{H}^1(K \cap \Sigma) \geq \limsup_\nu \mathcal{H}^1(K_\nu \cap \Sigma).$$

Proof. For a $\varepsilon > 0$ let $\bar{U}_\varepsilon$ stand for the closed ε-neighborhood of K. One has then $K_\nu \subset U_\varepsilon$ for all sufficiently large ν, and hence

$$\limsup \mu(K_\nu) \leq \mu(\bar{U}_\varepsilon).$$

Passing to a limit in the above relationship as $\varepsilon \to 0^+$ and minding that in this case $\mu(\bar{U}_\varepsilon) \to \mu(K)$, one gets the desired result. □

We will also need the following general property of Γ-convergence, which is provided in a slightly weaker form by Theorem 7.2 [3].

Proposition 4.2. *Let f_ν, $f\colon X \to \bar{\mathbb{R}}$ be functionals defined over a metric space X, and suppose that $\inf_X f_\nu = \inf_{X'} f_\nu$ for all $\nu \in \mathbb{N}$ and for some compact $X' \subset X$. Then the following assertions hold.*

(i) *If* $f(x) \leq \liminf_\nu f_\nu(x_\nu)$ *whenever* $\{x_\nu\} \subset X'$, $x_\nu \to x$ *in* X' *as* $\nu \to \infty$, *then*

$$\inf_X f \leq \liminf_\nu \inf_X f_\nu.$$

(ii) *If, in addition to* (i), *for every* $x \in X'$ *there is a sequence* $\{\tilde{x}_\nu\} \subset X$ *such that* $\limsup_\nu f_\nu(\tilde{x}_\nu) \leq f(x)$ *(in particular, this holds when* $\Gamma^- \limsup_\nu f_\nu \leq f$*), then* $\inf_X f_\nu \to \inf_X f$ *as* $\nu \to \infty$, *and whenever* $x_\nu \in X'$ *are such that*

$$f_\nu(x_\nu) \leq \inf_X f_\nu + \varepsilon_\nu, \tag{4.1}$$

where $\varepsilon_\nu \to 0$ *as* $\nu \to \infty$ *(in particular, if* $x_\nu \in \operatorname{Argmin} f_\nu$*) and* $x_\nu \to x$ *for an* $x \in X$, *then* $x \in \operatorname{Argmin} f$, *i.e.,* $f(x) = \inf_X f$.

Remark. If instead of the assumption in claim (i) one has the stronger property $f \leq \Gamma^- \liminf_\nu f_\nu$, then the proof shows that in fact (ii) is valid for every convergent sequence $\{x_\nu\} \subset X$ (i.e., not just for sequences in X') satisfying (4.1).

5. Existence of solutions. The aim of this section is to show that the statements of Problems 2 and 1 make sense and that both problems admit solutions. Namely, we prove the following result.

THEOREM 5.1. *There is a function* $B \in \mathcal{B}$ *which solves Problem* 2 *(resp., Problem* 1 *with given transport plan* γ*).*

To prove the above theorem, we need several auxiliary statements. Our main tool will be the following Γ-convergence result.

PROPOSITION 5.2. *Assume that for a sequence of monotone nondecreasing functions* $\{B_\nu\}$ *one has* $\Gamma^- \lim_\nu B_\nu = B$. *Then the following assertions hold.*

(i) $\delta(B, \theta) \leq \liminf_\nu \delta(B_\nu, \theta_\nu)$ *whenever* $\theta_\nu \to \theta$ *in* Θ.

(ii) *Let* $\Theta' \subset \Theta$ *be an arbitrary subset of* Θ *of paths having uniformly bounded length. Then, for every* $\theta \in \Theta'$ *and for a subsequence of* ν *(not relabeled) one has*

$$\delta(B_\nu, \theta) \leq \delta(B, \theta) + \varepsilon_\nu$$

for some sequence $\varepsilon_\nu \to 0$ *independent of* θ.

In particular,

$$\Gamma^- \lim_\nu \delta(B_\nu, \cdot) = \delta(B, \cdot). \tag{5.1}$$

(iii) *For every* $\theta \in \Theta$, *when* $x_\nu \to \theta(0)$ *and* $y_\nu \to \theta(1)$ *in* Ω, *there is a sequence* $\{\hat{\theta}_\nu\} \subset \Theta$ *such that* $\hat{\theta}_\nu(0) = x_\nu$, $\hat{\theta}_\nu(1) = y_\nu$, $\hat{\theta}_\nu \to \theta$, *and* $\delta(B, \theta) \geq \limsup_\nu \delta(B_\nu, \hat{\theta}_\nu)$.

Proof. The relationship (5.1) follows immediately from (i) and (ii), since (ii) implies $\delta(B, \theta) \geq \limsup_\nu \delta(B_\nu, \tilde{\theta}_\nu)$.

To prove (i), assume $\theta_\nu \to \theta$ in Θ. Consider any subsequence of θ_ν (not relabeled) such that the sequence $l_\nu := L(B_\nu, \theta_\nu)$ is convergent, and let $l := \lim_\nu l_\nu$. Since $l_\nu \leq \mathcal{H}^1(\theta_\nu \cap \Sigma)$, we get

$$0 \leq l \leq \mathcal{H}^1(\theta \cap \Sigma)$$

in view of Lemma 4.1. On the other hand, recalling that A is continuous and monotone nondecreasing, $B(l) \leq \liminf_\nu B_\nu(l_\nu)$ (by Γ^--convergence assumption) and $\mathcal{H}^1(\theta) \leq$

$\liminf_\nu \mathcal{H}^1(\theta_\nu)$ (by the Golab Theorem [2, Theorem 4.4.7]), we obtain

$$\begin{aligned}\delta(B,\theta) &\le A(\mathcal{H}^1(\theta)-l)+B(l)\\ &\le A(\liminf_\nu(\mathcal{H}^1(\theta_\nu)-l_\nu))+\liminf_\nu B_\nu(l_\nu)\\ &\le \liminf_\nu(A(\mathcal{H}^1(\theta_\nu)-l_\nu)+B_\nu(l_\nu))\\ &= \liminf_\nu \delta(B_\nu,\theta_\nu).\end{aligned}$$

To prove (ii), choose an $\varepsilon>0$ and cover $[0,\mathcal{H}^1(\Sigma)]$ with a finite number of disjoint intervals of the form $I_i:=(\bar{l}_i-\delta_i,\bar{l}_i]$, with $\delta_i\le\varepsilon$ (possibly zero) sufficiently small such that $B(l)\ge B(\bar{l}_i)-\varepsilon/2$ for all $l\in I_i$. Such a cover exists since B is monotone nondecreasing and l.s.c., while the interval $[0,\mathcal{H}^1(\Sigma)]$ is compact. By definition of Γ^--convergence, for each $\bar{l}_i$ there is a sequence $l_{i,\nu}\to\bar{l}_i$ such that $B_\nu(l_{i,\nu})\to B(\bar{l}_i)$, so that $B_\nu(l_{i,\nu})\le B(\bar{l}_i)+\varepsilon/2$ for all sufficiently large ν, and hence

$$B_\nu(l_{i,\nu})\le B(l)+\varepsilon$$

for all $l\in I_i$. By the finiteness of the intervals I_i, there is a unique $\nu=\nu(\varepsilon)$ such that the above relationship is true for all i. Define the function l'_ν: $[0,\mathcal{H}^1(\Sigma)]\to[0,\mathcal{H}^1(\Sigma)]$ associating to any $l\in I_i$ the number $l_{i,\nu}\wedge l$. One has then $B_\nu(l'_\nu)\le B(l)+\varepsilon$ and $0\le l-l'_\nu\le\bar{l}_i-l_{i,\nu}$. Thus, possibly up to passing to a (not relabeled) subsequence of ν, we get a sequence of functions l'_ν: $[0,\mathcal{H}^1(\Sigma)]\to[0,\mathcal{H}^1(\Sigma)]$ such that

$$B_\nu(l'_\nu)\le B(l)+1/\nu,\quad l'_\nu\le l,\quad l-l'_\nu\le 1/\nu. \tag{5.2}$$

Now, for any $\theta\in\Theta'$ we write $l_\nu(\theta):=l'_\nu(L(B,\theta))$, and since $l'_\nu\le L(B,\theta)\le\mathcal{H}^1(\theta\cap\Sigma)$ for every ν, from (5.2) we get

$$A(\mathcal{H}^1(\theta)-l_\nu)+B_\nu(l_\nu)\le A(\mathcal{H}^1(\theta)-l_\nu)+B(L(B,\theta))+1/\nu, \tag{5.3}$$

where $l_\nu:=l_\nu(\theta)$.

By the uniform continuity of A on the bounded interval $[0,\max_{\theta\in\Theta'}\mathcal{H}^1(\theta)]$, we know that $A(\mathcal{H}^1(\theta)-l_\nu)\le A(\mathcal{H}^1(\theta)-L(B,\theta))+\tilde{\varepsilon}_\nu$ with $\tilde{\varepsilon}_\nu\to 0$ as $\nu\to\infty$ and $\tilde{\varepsilon}_\nu$ not depending on $\theta\in\Theta'$. Therefore, (5.3) implies for each $\theta\in\Theta'$ the following estimate:

$$\begin{aligned}\delta(B,\theta) &= A(\mathcal{H}^1(\theta)-L(B,\theta))+B(L(B,\theta))\\ &\ge A(\mathcal{H}^1(\theta)-l_\nu(\theta))+B_\nu(l_\nu)-1/\nu-\tilde{\varepsilon}_\nu\\ &\ge \delta(B_\nu,\theta)-\varepsilon_\nu,\end{aligned} \tag{5.4}$$

where $\varepsilon_\nu:=\tilde{\varepsilon}_\nu+1/\nu$. Therefore, (ii) follows.

Last, to prove (iii) we let $\alpha_\nu\in\Theta$ stand for the segment connecting x_ν with $\theta(0)$ and $\beta_\nu\in\Theta$ stand for the segment connecting $\theta(1)$ with y_ν, respectively. Now set

$$\hat{\theta}_\nu:=\alpha_\nu\circ\theta\circ\beta_\nu,$$

so that clearly $\hat{\theta}_\nu$ connects x_ν with y_ν and $\hat{\theta}_\nu\to\theta$ as $\nu\to\infty$. One has $\mathcal{H}^1(\hat{\theta}_\nu\cap\Sigma)\ge\mathcal{H}^1(\theta\cap\Sigma)$, and hence

$$\delta(B_\nu,\hat{\theta}_\nu)\le A(\mathcal{H}^1(\hat{\theta}_\nu)-L(B_\nu,\theta))+B_\nu(L(B_\nu,\theta))$$

by (3.2). Mind that

$$|\mathcal{H}^1(\theta)-\mathcal{H}^1(\hat{\theta}_\nu)|\le\mathcal{H}^1(\alpha_\nu)+\mathcal{H}^1(\beta_\nu)=|x_\nu-\theta(0)|+|\theta(1)-y_\nu|\to 0$$

as $\nu \to \infty$. This together with the uniform continuity of A on bounded intervals and (5.4) provides

$$\delta(B_\nu, \hat{\theta}_\nu) \le \delta(B, \theta) + \varepsilon'_\nu$$

for some sequence $\varepsilon'_\nu \to 0$ as $\nu \to \infty$, which concludes the proof. □

From now on we will use the notation Θ' and λ provided by (3.7). Defining f': $\Theta \to \Theta'$ by the relationship

$$f'(\theta) := \begin{cases} \theta, & \theta \in \Theta', \\ [\theta(0), \theta(1)], & \theta \notin \Theta', \end{cases} \tag{5.5}$$

one clearly has for $\theta' := f'(\theta)$ that

$$\delta(B, \theta) \ge \delta(B, \theta') \tag{5.6}$$

for all $\theta \in \Theta$. In fact, if $\theta \notin \Theta'$, then

$$\mathcal{H}^1(\theta) - L(B, \theta) > \lambda - \mathcal{H}^1(\Sigma) \ge \operatorname{diam} \Omega, \tag{5.7}$$

and hence $\delta(B, \theta) \ge A(\operatorname{diam} \Omega)$, while $\delta(B, \theta') \le A(\mathcal{H}^1([\theta(0), \theta(1)])) \le A(\operatorname{diam} \Omega)$.

We prove yet another convergence statement concerning the distance d_B.

Proposition 5.3. *Assume $x_\nu \to x$, $y_\nu \to y$ in Ω and $\Gamma^- \lim_\nu B_\nu = B$. Then*

(i) *$d_{B_\nu}(x_\nu, y_\nu) \to d_B(x, y)$. In particular, setting $B_\nu := B$, we get that the function d_B is continuous;*

(ii) *if $\theta_\nu \in \widetilde{\Theta}(B_\nu)$ and $\theta_\nu \to \theta$ in Θ, then $\theta \in \widetilde{\Theta}(B)$. In particular, setting $B_\nu := B$, we get that $Q_B(x, y)$ is closed for all $(x, y) \in \Omega \times \Omega$ and also $\widetilde{\Theta}(B)$ is closed.*

Proof. Define $\hat{\delta}(x, y, B, \cdot)$: $\Theta \to \bar{\mathbb{R}}$ by setting

$$\hat{\delta}(x, y, B, \theta) := \begin{cases} \delta(B, \theta) & \text{if } \theta(0) = x, \theta(1) = y, \\ +\infty & \text{otherwise.} \end{cases}$$

Proposition 5.2(i) and 5.2(iii) imply that

$$\hat{\delta}(x, y, B, \cdot) = \Gamma^- \lim_\nu \hat{\delta}(x_\nu, y_\nu, B_\nu, \cdot)$$

whenever $x_\nu \to x$, $y_\nu \to y$ in Ω and $\Gamma^- \lim_\nu B_\nu = B$. According to (5.6) one has

$$d_B(x, y) = \min_{\Theta'} \hat{\delta}(x, y, B, \cdot),$$

where the minimum is attained since $\Theta' \subset \Theta$ is compact, and hence $Q_B(x, y) = \operatorname{Argmin}_{\Theta'} \hat{\delta}(x, y, B, \cdot)$ for every $(x, y) \in \Omega \times \Omega$ and for every monotone nondecreasing B. Thus we get both (i) and (ii) as an immediate consequence of Proposition 4.2(ii) (recalling the remark following this proposition). □

Let $E' \subset \mathcal{M}^+(\Theta')$ stand for the set of all transports concentrated on Θ' and observe that it is compact since all the transports have the same total mass $\varphi_0(\Omega)$. We prove now the following general statement.

Lemma 5.4. *For every transport η and for each $B \in \mathcal{B}$ there is a transport $\eta' \in E'$ such that we have the following assertions:*

(i) $(p_0 \times p_1)_\# \eta' = (p_0 \times p_1)_\# \eta$.

(ii) $C(B)(\eta') \leq C(B)(\eta)$. *In particular,*

$$\eta' \in E_{opt}(B) \text{ whenever } \eta \in E_{opt}(B).$$

(iii) *If η is concentrated on $\tilde{\Theta}(B)$ (in particular, when $\eta \in E_{opt}(B)$), then so is η', and*

$$\hat{F}(B, \eta') \geq \hat{F}(B, \eta).$$

Remarks.

(A) The first claim of (ii) shows in fact that the infimum of $C(B)(\eta)$ over all admissible transports η coincides with that over E'. Moreover, since $\delta(B, \cdot)$ is l.s.c. and E' is compact, then as a by-product we have $E_{opt}(B) \cap E' \neq \emptyset$.

(B) The claim (iii) is applicable both to $\eta \in E_{opt}(B)$ (such transports are concentrated on $\widetilde{\Theta}(B)$ due to the last claim of Proposition 3.2) and to $\eta \in \operatorname{Argmin}\{C(B)(\eta') : (p_0 \times p_1)_{\#}\eta' = \gamma\}$ with γ a given transport plan (such transports are also concentrated on $\widetilde{\Theta}(B)$, as shown in the proof of Proposition 3.5).

(C) If A is either strictly increasing or unbounded, then by the remark following Lemma 3.3 one has $\mathcal{H}^1(\theta) \leq l$ with $l > 0$ independent of B for all $\theta \in \widetilde{\Theta}(B)$. Hence, letting $\tilde{E}$ stand for the set of admissible transports concentrated on the set of paths with length not exceeding l, we get that $E_{opt}(B) \subset \tilde{E}$ for all $B \in \mathcal{B}$. Clearly, $\tilde{E}$ is compact in $*$-weak topology of measures.

Proof. Set $\eta' := f'_{\#}\eta$, where f' is defined by (5.5). Then (i) is immediate, while (ii) follows from

$$C(B)(\eta') = \int_\Theta \delta(B, \theta)\, d\eta' = \int_\Theta \delta(B, f'(\theta))\, d\eta \leq \int_\Theta \delta(B, \theta)\, d\eta = C(B)(\eta),$$

which in turn is a consequence of (5.6). Finally, observe that if $\theta \in \widetilde{\Theta}(B) \setminus \Theta'$, then $B(L(B, \theta)) = 0$ by Lemma 3.3. This, minding that in view of (5.6) one has $f'(\widetilde{\Theta}(B)) \subset \widetilde{\Theta}(B)$ (which shows the first part of (iii)), implies the estimate

$$\begin{aligned}
\hat{F}(B, \eta') &= \int_{\widetilde{\Theta}(B)} B(L(B, \theta))\, d\eta' &&= \int_{\widetilde{\Theta}(B)} B(L(B, f'(\theta)))\, d\eta \\
&\geq \int_{\Theta' \cap \widetilde{\Theta}(B)} B(L(B, f'(\theta)))\, d\eta &&= \int_{\Theta' \cap \widetilde{\Theta}(B)} B(L(B, \theta))\, d\eta \\
&= \int_{\widetilde{\Theta}(B)} B(L(B, \theta))\, d\eta &&= \hat{F}(B, \eta),
\end{aligned}$$

which concludes the proof of (iii). □

We will need the following auxiliary statement which will be used in the proof of the main existence result and of Proposition 5.6 in what follows.

LEMMA 5.5. *Assume that for a sequence of nonnegative Borel functions $\{u_\nu\}$: $X \to \mathbb{R}$ defined over a compact metric space X one has $u^- = \Gamma^- \liminf_\nu u_\nu$. Then for any sequence of measures $\{\eta_\nu\} \subset \mathcal{M}^+(X)$ such that $\eta_\nu \rightharpoonup \eta$ $*$-weakly one has*

$$\int_X u^-\, d\eta \leq \liminf_\nu \int_X u_\nu\, d\eta_\nu.$$

Remark. Analogously, if $\Gamma^+ \limsup_\nu u_\nu = u^+$ and u_ν are uniformly bounded from above, then

$$\int_X u^+ \, d\eta \geq \limsup_\nu \int_X u_\nu \, d\eta_\nu.$$

Proof. It is easy to verify that

$$\Gamma^- \liminf u_\nu(x) = \sup_{\nu \in \mathbb{N}} \tau_\nu, \text{ where } \tau_\nu := \left(\inf_{m \geq \nu} u_m \right)^- . \tag{5.8}$$

Fix a $j \in \mathbb{N}$ and evaluate

$$\begin{aligned} \liminf_{\nu \to \infty} \int_X u_\nu \, d\eta_\nu &\geq \liminf_{\nu \to \infty} \int_X \left(\inf_{m \geq j} u_m \right) d\eta_\nu \\ &\geq \liminf_{\nu \to \infty} \int_X \left(\inf_{m \geq j} u_m \right)^- d\eta_\nu \\ &= \liminf_{\nu \to \infty} \int_X \tau_j \, d\eta_\nu \qquad \geq \int_X \tau_j \, d\eta. \end{aligned}$$

Since this is true for any $j \in \mathbb{N}$, by the Beppo–Levi monotone convergence theorem and (5.8) the thesis follows. □

At this moment we may claim another Γ-convergence result.

Proposition 5.6. *Assume that for a sequence of nondecreasing functions $\{B_\nu\}$ one has $\Gamma^- \lim_\nu B_\nu = B$. Then $\min C(B_\nu) \to \min C(B)$, where both minima are taken over the set of all admissible transports. Moreover, if $\eta_\nu \in E_{opt}(B_\nu) \cap E'$ and $\eta_\nu \rightharpoonup \eta$ $*$-weakly in the sense of measures, then $\eta \in E_{opt}(B)$.*

Proof. Applying the first part of Lemma 5.5 with $X := \Theta'$, $u_\nu := \delta(B_\nu, \cdot)$, and $u^- := \delta(B, \cdot)$, minding that $\Gamma^- \lim_\nu u_\nu = u^-$ in view of Proposition 5.2, yields

$$C(B)(\eta') \leq \liminf_\nu C(B_\nu)(\eta'_\nu)$$

whenever $\{\eta'_\nu\} \subset E'$ and $\eta'_\nu \rightharpoonup \eta'$ $*$-weakly in the sense of measures. On the other hand, up to a subsequence of ν (not relabeled) one has by Proposition 5.2(ii)

$$\delta(B_\nu, \theta) \leq \delta(B, \theta) + \varepsilon_\nu$$

for each $\theta \in \Theta'$, and for some sequence $\{\varepsilon_\nu\}$ (independent of θ), $\varepsilon_\nu \to 0$ as $\nu \to \infty$. Hence, integrating the above relationship in arbitrary $\eta \in E'$, we get

$$\limsup_\nu C(B_\nu)(\eta) \leq C(B)(\eta).$$

The validity of the thesis now follows directly from Proposition 4.2(ii), minding that according to the remark following Lemma 5.4 the infimum of $C(B)$ over all admissible transports coincides with that over compact set E' of the latter. □

We will also need another convergence result below for the functions $L(B, \cdot)$ and for their compositions with B.

Lemma 5.7. *Assume that for a sequence of monotone nondecreasing functions $\{B_\nu\}$ one has $\Gamma^- \lim_\nu B_\nu = B$ while $\theta_\nu \in \widetilde{\Theta}(B_\nu)$ and $\theta_\nu \to \theta$ in Θ as $\nu \to \infty$. Then*

$$\begin{aligned} \limsup\nolimits_\nu L(B_\nu, \theta_\nu) &\leq L(B, \theta), \\ \limsup\nolimits_\nu B_\nu^-(L(B_\nu, \theta_\nu)) &\leq B(L(B, \theta)). \end{aligned}$$

Remark. Applied with $B_\nu := B$ the above lemma proves in particular that $L(B,\cdot)$ and its composition with B are u.s.c. over $\widetilde{\Theta}(B)$.

Proof. Without loss of generality we assume that $l_\nu := L(B_\nu, \theta_\nu) \to l$. By Proposition 5.3(i) one has

$$d_B(\theta(0), \theta(1)) = \lim_\nu d_{B_\nu}(\theta_\nu(0), \theta_\nu(1)) = \lim_\nu A(\mathcal{H}^1(\theta_\nu) - l_\nu) + B_\nu(l_\nu).$$

Recall that $\mathcal{H}^1(\theta) \le \liminf_\nu \mathcal{H}^1(\theta_\nu)$ by the Golab theorem [2, Theorem 4.4.7], the function A is continuous and monotone nondecreasing, while $B(l) \le \liminf_\nu B_\nu(l_\nu)$, due to the Γ^--convergence assumption. We get thus from the above relationship the estimate

$$\lim_\nu A(\mathcal{H}^1(\theta_\nu) - l_\nu) + B_\nu(l_\nu) \ge A(\mathcal{H}^1(\theta) - l) + B(l).$$

Since $\mathcal{H}^1(\theta \cap \Sigma) \ge \limsup \mathcal{H}^1(\theta_\nu \cap \Sigma) \ge l$ by Lemma 4.1, we get then by definition of $\delta(B,\cdot)$ that

$$A(\mathcal{H}^1(\theta) - l) + B(l) \ge \delta(B,\theta) \ge d_B(\theta(0), \theta(1)).$$

Combining the above estimates, we have that all the inequalities above are in fact equalities, and in particular

$$A(\mathcal{H}^1(\theta) - l) + B(l) = \delta(B,\theta),$$

which implies $L(B,\theta) \ge l$ by (3.1), hence proving the first part of the statement. As for the second one, the above proven chain of equalities gives $B(l) = \lim_\nu B_\nu(l_\nu)$, so that the thesis follows since $L(B,\theta) \ge l$ and B is monotone nondecreasing. □

Finally, we prove the following truncation result.

LEMMA 5.8. *For every $B \in \mathcal{B}$, $c > A(\lambda)$, and $\eta \in E'$ one has*

$$\hat{F}(B \wedge c, \eta) = \hat{F}(B, \eta).$$

Further, $E_{opt}(B) \cap E' = E_{opt}(B \wedge c) \cap E'$.

Proof. For every $\theta \in \Theta'$ and for every $B \in \mathcal{B}$ one has $\delta(B,\theta) \le A(\mathcal{H}^1(\theta)) \le A(\lambda)$, since $\mathcal{H}^1(\theta) \le \lambda$. Hence, $B(L(B,\theta)) \le \delta(B,\theta) \le A(\lambda) < c$. Setting $\bar{B} := B \wedge c$, we have also $\bar{B}(L(\bar{B},\theta)) < c$, which implies $L(B,\theta) = L(\bar{B},\theta)$, and

$$B(L(B,\theta)) = \bar{B}(L(\bar{B},\theta)), \qquad \delta(B,\theta) = \delta(\bar{B},\theta). \tag{5.9}$$

The first equality of (5.9) implies the first claim of the statement being proven, while the second of the equalities in (5.9) proves the second claim because according to Lemma 5.4(ii) the minima of $C(B)$ and of $C(\bar{B})$ are attained at E'. □

Finally, we may prove Theorem 5.1.

Proof of Theorem 5.1. We provide here the proof only for Problem 2, since the proof for Problem 1 is completely analogous. Let $\{B_\nu, \eta_\nu\}$ be a maximizing sequence for F, and in particular, $\eta_\nu \in E_{opt}(B_\nu)$. By Lemma 5.4 we may suppose without loss of generality that each $\eta_\nu \in E'$. Since E' is compact in $*$-weak topology of measures by the remark following Lemma 5.4, then one may assume up to passing to subsequence (not relabeled) that $\eta_\nu \rightharpoonup \eta$ $*$-weakly in the sense of measures.

Let $c > A(\lambda)$. In view of Lemma 5.8 we may assume without loss of generality that $0 \le B_\nu(x) \le c$ for all $x \in \mathbb{R}^+$. Therefore, we may choose a subsequence of

B_ν (not relabeled) such that for some B: $\mathbb{R}^+ \to \mathbb{R}$ one has $\Gamma^- \lim_\nu B_\nu = B$. By Proposition 5.6 then $\eta \in E_{opt}(B)$.

Define now for each $B \in \mathcal{B}$ the map $\tilde{B}$: $\Theta \to \bar{\mathbb{R}}$ by $\tilde{B}(\theta) := B(L(B,\theta))$. Note that Lemma 5.7 implies $\Gamma^+ \limsup_\nu \tilde{B}_\nu \le \tilde{B}$. It remains to observe that in view of the remark following Lemma 5.5 one has

$$\begin{aligned}
\limsup_\nu \int_\Theta B_\nu(L(B_\nu,\theta))\, d\eta_\nu(\theta) &= \limsup_\nu \int_\Theta \tilde{B}_\nu(\theta)\, d\eta_\nu(\theta) \\
&\le \int_\Theta (\Gamma^+ \limsup_\nu \tilde{B}_\nu)(\theta)\, d\eta(\theta) \\
&\le \int_\Theta \tilde{B}(\theta)\, d\eta(\theta) \\
&= \int_\Theta B(L(B,\theta))\, d\eta(\theta),
\end{aligned}$$

and hence B solves Problem 2. ∎

6. Qualitative properties of optimal policies. The aim of this section is to prove the existence of optimal pricing policies possessing some natural regularity properties. For this purpose we introduce now the following natural notion of a *minimum optimal pricing policy* for Problem 2 (resp., Problem 1).

DEFINITION 6.1. *Let the couple of functions $\{B_1, B_2\} \in \mathcal{B}$ be optimal pricing policies for Problem* 2 *(resp., Problem* 1 *with given transport plan γ). We will say that B_1 dominates B_2, if $B_1(u) \ge B_2(u)$ for all $u \in \mathbb{R}^+$.*

A function $\hat{B} \in \mathcal{B}$ is called minimum optimal pricing policy for Problem 2 *(resp., Problem* 1 *with given transport plan γ) if it is an optimal pricing policy for the respective problem which does not dominate any other one.*

We will now prove the existence of minimum optimal pricing policies.

THEOREM 6.2. *There exists a minimum optimal pricing policy for Problem* 2 *(resp., Problem* 1 *with given transport plan γ).*

Proof. As in Theorem 5.1, we provide the proof for Problem 2, because the proof for Problem 1 is completely analogous. Let $\mathcal{B}_{opt}$ stand for an arbitrary set of optimal pricing policies for Problem 2 such that for every couple of different policies $\{B_1, B_2\} \in \mathcal{B}_{opt}$ one pricing policy dominates the other. Define $\hat{B}$: $\mathbb{R}^+ \to \mathbb{R}^+$ by the relationship

$$\hat{B}(u) := \inf\{B(u) \,:\, B \in \mathcal{B}_{opt}\}$$

for all $u \in \mathbb{R}^+$. Let $c > A(\lambda)$. According to Lemma 5.8,

$$B(u) = \inf\left\{B(u) \,:\, B \in \bar{\mathcal{B}}_{opt}\right\}, \text{ where } \bar{\mathcal{B}}_{opt} := \{B \wedge c \,:\, B \in \mathcal{B}_{opt}\}$$

for all $u \in \mathbb{R}^+$, while the elements of $\bar{\mathcal{B}}_{opt}$ are still optimal pricing policies. We may therefore obtain a sequence $\{B_\nu\} \subset \bar{\mathcal{B}}_{opt}$ converging to B pointwise, i.e., $B_\nu(u) \to \hat{B}(u)$ for all $u \in \mathbb{R}^+$ as $\nu \to \infty$. Up to a subsequence (not relabeled), we have that $\Gamma^- \lim_\nu B_\nu = \hat{B}^-$. Consider a sequence of transports $\{\eta_\nu\}$ such that $(B_\nu, \eta_\nu) \in \operatorname{Argmax} F$, and hence, in particular, $\eta_\nu \in E_{opt}(B_\nu)$ for all $\nu \in \mathbb{N}$. In view of Lemma 5.4, we may suppose without loss of generality that all $\eta_\nu \in E'$ and since E' is compact in $*$-weak topology of measures, then again up to a subsequence (not relabeled), one has $\eta_\nu \rightharpoonup \eta$ in $\mathcal{M}^+(\Theta)$. By Proposition 5.6 one has $\eta \in E_{opt}(\hat{B}^-)$.

Now, in a complete analogy to the proof of Theorem 5.1, we observe referring to Lemmata 5.5 and 5.7 that

$$\max F \quad = \quad \limsup_{\nu} \int_{\Theta} B_{\nu}(L(B_{\nu},\theta))\, d\eta_{\nu}(\theta) \quad \leq \quad \int_{\Theta} \hat{B}^{-}(L(B,\theta))\, d\eta(\theta)$$

and thus, minding that $\hat{B}^{-} \in \mathcal{B}$, one has that $\hat{B}^{-}$ is an optimal pricing policy for Problem 2. Clearly, $\hat{B}^{-}(u) \leq \hat{B}(u)$ for all $u \in \mathbb{R}^{+}$ and hence does not dominate any pricing policy from $\mathcal{B}_{opt}$. Due to the arbitrarity of choice of $\mathcal{B}_{opt}$, we conclude referring to Zorn's lemma the existence of a minimum element with respect to the domination relation in the set of all optimal pricing policies for Problem 2. This minimum element is in fact an optimal pricing policy. □

Remark. We do not know whether in fact the minimum optimal pricing policy is *unique* even for Problem 1 with fixed transport plan γ. If that were the case, then the minimum optimal pricing policy would be necessarily the pointwise minimum of optimal pricing policies for this problem, as one can easily deduce from the above proof. It is worth noting that to prove the uniqueness of the minimum optimal pricing policy it would be enough to show that $B_1 \wedge B_2$ is an optimal pricing policy whenever both B_1 and B_2 are also.

6.1. Regularity. We are able to prove now that every minimum optimal pricing policy for Problem 2 (resp., Problem 1) is always continuous (moreover, its modulus of continuity is determined by that of A).

Denote by ω_A the modulus of continuity of A over $[0,\lambda]$, where $\lambda := \operatorname{diam}\Omega + \mathcal{H}^1(\Sigma)$ as defined by (3.7). Namely, let

$$\omega_A(l) := \sup\{|A(u) - A(v)| \,:\, |u-v| \leq l, 0 \leq u \leq \lambda, 0 \leq v \leq \lambda\}$$

for every $l \leq \lambda$, and set for convenience $\omega_A(l) := \omega_A(\lambda)$ whenever $l > \lambda$. It is easy to observe that $\omega_A\colon \mathbb{R}^{+} \to \mathbb{R}^{+}$ defined in this way is a continuous monotone nondecreasing function.

We are able now to prove the following regularity theorem, which shows that the modulus of continuity of every minimum optimal pricing policy B is dominated by that of A.

THEOREM 6.3. *Let B be a minimum optimal pricing policy for Problem* 2 *(resp., Problem* 1 *with given transport plan γ). Then*

$$B(u) - B(v) \leq \omega_A(|u-v|)$$

for all $u, v \in \mathbb{R}^{+}$.

Remark. In particular, the above theorem shows that every minimum optimal pricing policy B is Lipschitz continuous whenever A is also (and the Lipschitz constant of B does not exceed that of A).

Proof. As usual, we provide the proof for Problem 2, since the proof for Problem 1 is completely analogous. If the claim is false, then there is a $u \in \mathbb{R}^{+}$ and an $l > 0$ such that $B(u+l) > B(u) + \omega_A(l)$. Let $\alpha(v) := B(u) + \omega_A(v-u)$ and note that $\alpha\colon [u, +\infty) \to \mathbb{R}^{+}$ is clearly continuous since so is ω_A. Because B is l.s.c. and α is continuous, then the set $\{v \geq u \,:\, B(v) > \alpha(v)\}$ is open, and since it contains $u+l$ then there is a maximum open interval $(a,b) \ni u+l$ such that

$$B(v) > \alpha(v) \text{ for all } l \in (a,b).$$

Note that

$$B(a) \leq \alpha(a) \tag{6.1}$$

due to the maximality of (a, b). We define a l.s.c. monotone nondecreasing function $z\colon (a, b) \to \mathbb{R}^+$ by the formula

$$z(v) := (\alpha(v) + B(v))/2, \qquad v \in (a, b).$$

Now set

$$\bar{B}(v) := \begin{cases} B(v), & v \notin (a, b), \\ z(v), & v \in (a, b). \end{cases}$$

Clearly, $\bar{B} \in \mathcal{B}$. In fact, the only nontrivial assertion to verify is that $\bar{B}$ is monotone nondecreasing, i.e., that $\bar{B}(v_1) \leq \bar{B}(v_2)$ whenever $v_1 \leq v_2$. The latter claim is obvious if both $v_1 \in (a, b)$ and $v_2 \in (a, b)$ (or both $v_1 \notin (a, b)$ and $v_2 \notin (a, b)$), since z (resp., B) is monotone nondecreasing. If $v_1 \leq a$, $v_2 \in (a, b)$, then

$$\bar{B}(v_1) = B(v_1) \leq B(a) \leq \alpha(a) \leq \alpha(v_2) \leq z(v_2) = \bar{B}(v_2),$$

the second inequality in the latter chain of estimates being true due to (6.1). Finally, if $v_1 \in (a, b)$, $v_2 \geq b$, then

$$\bar{B}(v_1) = z(v_1) \leq B(v_1) \leq B(b) \leq B(v_2) = \bar{B}(v_2).$$

Observe now that $\bar{B}(v) < B(v)$ for all $v \in (a, b)$ by construction. Hence, by minimality of B, one has that $\bar{B}$ cannot solve Problem 2. This means that whenever η is such that $(B, \eta) \in \operatorname{Argmax} F$, and $\bar{\eta} \in E_{opt}(\bar{B})$, then

$$\hat{F}(\bar{B}, \bar{\eta}) < \hat{F}(B, \eta). \tag{6.2}$$

On the other hand, one has for every $\theta \in \Theta'$ that

$$L(B, \theta) \notin (a, b). \tag{6.3}$$

In fact, if $\mathcal{H}^1(\theta \cap \Sigma) \geq v$ for some $v \in (a, b)$, then

$$\begin{array}{rcl} A(\mathcal{H}^1(\theta) - v) + B(v) & > & A(\mathcal{H}^1(\theta) - v) + B(u) + \omega_A(v - u) \\ & \geq & A(\mathcal{H}^1(\theta) - u) + B(u), \end{array} \tag{6.4}$$

where in the latter relationship we used the fact that for $\theta \in \Theta'$ one has $\mathcal{H}^1(\theta) \leq \lambda$. Clearly, (6.4) implies $L(B, \theta) \neq v$ and hence (6.3). Analogously,

$$L(\bar{B}, \theta) \notin (a, b) \tag{6.5}$$

for $\theta \in \Theta'$ since still

$$\bar{B}(v) = z(v) > B(u) + \omega_A(v - u) = \bar{B}(u) + \omega_A(v - u)$$

whenever $v \in (a, b)$.

By Lemma 5.4 we may suppose without loss of generality that η is concentrated over Θ'. But for every $\theta \in \Theta'$ one has, in view of (6.5) and the definition of $\bar{B}$, that

$$\begin{array}{rcl} \displaystyle\min_{l \in [0, \mathcal{H}^1(\theta \cap \Sigma)]} A(\mathcal{H}^1(\theta) - l) + \bar{B}(l) & = & A(\mathcal{H}^1(\theta) - L(\bar{B}, \theta)) + \bar{B}(L(\bar{B}, \theta)) \\ & = & A(\mathcal{H}^1(\theta) - L(\bar{B}, \theta)) + B(L(\bar{B}, \theta)), \end{array}$$

which implies

$$L(\bar{B}, \theta) = L(B, \theta) \tag{6.6}$$

and also

$$\delta(\bar{B}, \theta) = \delta(B, \theta).$$

The latter relationship implies $\eta \in E_{opt}(\bar{B})$, while from (6.6) together with (6.5) and the definition of $\bar{B}$ one gets

$$\bar{B}(L(\bar{B}, \theta)) = B(L(B, \theta))$$

for η-a.e. $\theta \in \Theta$, and therefore $\hat{F}(\bar{B}, \bar{\eta}) = \hat{F}(B, \eta)$, contradicting (6.2). The latter contradiction concludes the proof. □

6.2. Natural bounds. We now prove that the function A (the cost of transportation without the use of the transport network) gives a natural bound to solutions of the problems being studied, i.e., to the optimal pricing policies.

Corollary 6.4. *Suppose that the function A is subadditive, i.e.,*

$$A(u + v) \leq A(u) + A(v) \textit{ for every } u, v \in \mathbb{R}^+.$$

Then every minimum optimal pricing policy B for Problem 2 (resp., for Problem 1 with given transport plan γ) satisfies $B(u) \leq A(u)$ for all $u \in [0, \mathcal{H}^1(\Sigma)]$.

Proof. Minding that $B(0) = 0$, we get

$$B(u) = B(u) - B(0) \leq \omega_A(u)$$

by Theorem 6.3. Let $v \in [0, \lambda]$ be such that $\omega_A(u) = A(v + u) - A(v)$. Minding the subadditivity of A, we get $\omega_A(u) \leq A(u)$, hence proving $B(u) \leq A(u)$. □

We will show in Example 7.2 that in fact the condition of subadditivity of A is inevitable for the statement of Corollary 6.4 to hold.

7. Some examples.

Here we present some of the simplest examples of the optimal pricing policies solving Problems 2 and 1.

7.1. Single citizen. Consider the following sample two-dimensional situation (i.e., $n = 2$). Let $\varphi_1 := \delta_0$ be a Dirac measure concentrated in the origin of the coordinate system and $\varphi_0 := \delta_S$ be a Dirac measure concentrated in a point $S = (l, h)$. In this case the population is clearly represented by a single citizen living in S, while the service (or working place) is unique and is located in the origin of the coordinate system. We assume the transportation network $\Sigma := [0, l]$ to be the line segment of length l along the x-axis (see Figure 1). Recall also that we are interested only in l.s.c. pricing policies B.

Note that in this sample situation there is only one admissible transport plan $\gamma = \varphi_0 \otimes \varphi_1$, hence every solution to Problem 2 is also a solution to Problem 1, and vice versa.

Clearly, regardless of the cost functions A and B, for the passenger who moves between S and O it is convenient to choose the path θ_t which is a union of two line segments: a line segment connecting S with a point on Σ which has coordinates $(t, 0)$, $t \leq l$, and another one connecting the latter to O. In this case the total cost for the passenger is

$$\delta(B, \theta_t) = A\left((h^2 + (l - t)^2)^{1/2}\right) + B(t), \tag{7.1}$$

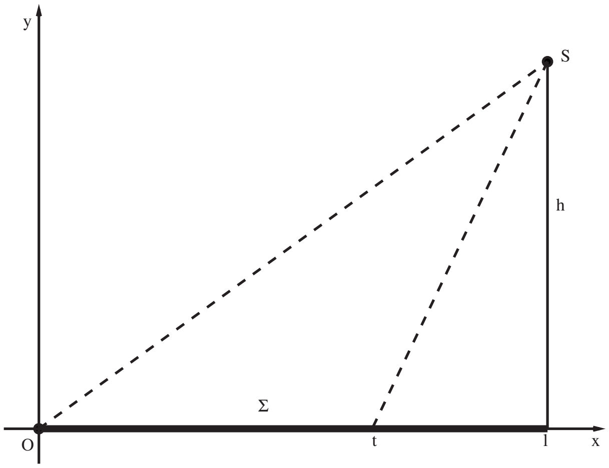


FIG. 1. *Sample transportation network* Σ *and the choice of the optimal path.*

and hence

$$d_B(S,0) = \min_{0\le t\le l} \delta(B,\theta_t) \text{ and } MK(B) = d_B(S,0).$$

Therefore, the scope of the "population" (the users of the transportation network) can be understood as that of simply finding, given a pricing policy B, an optimal $t \in [0,l]$ which minimizes the cost $\delta(B,\theta_t)$ among all $t \in [0,l]$. The optimal transports $\eta_{opt} \in E_{opt}(B)$ will be thus concentrated only on the paths θ_t with t optimal in the above sense. Let t_B be the maximum among all optimal $t \in [0,l]$. Note that all the paths θ_t with t optimal provide the same cost of movement $\delta(B,\theta_t)$ for the passenger. It is also worth observing that the Dirac measure η_B concentrated on the optimal path θ_{t_B} is the unique transport maximizing $\hat{F}(B,\cdot)$ among all optimal transports.

It is easy to observe therefore that under our assumption the owner of the transportation network earns $B(t_B)$ and hence is interested in maximizing the latter over all nonnegative monotone nondecreasing B satisfying $B(0) = 0$. In other words, Problem 2 reduces to that of maximizing the functional $\tilde{F}$ defined by $\tilde{F}(B) := B(t_B)$ over all $B \in \mathcal{B}$. To solve this problem, we note that

$$d_B(S,0) \le A(\mathcal{H}^1([OS])) = A\left((h^2+l^2)^{1/2}\right).$$

Hence,

$$\delta(B,\theta_{t_B}) = A\left((h^2+(l-t_B)^2)^{1/2}\right) + B(t_B) \le A\left((h^2+l^2)^{1/2}\right),$$

which gives a bound to the maximum of $B(t_B)$: in fact,

$$B(t_B) \le A\Big((h^2+l^2)^{1/2}\Big) - A(h).$$

We claim that there is a pricing policy B such that the latter estimate is an equality. This policy would of course provide the maximum of $\tilde{F}$ (and hence of $\hat{F}$). For this situation, one necessarily has $t_B = l$ and then

$$B(t_B) = B(l) = A\left((h^2+l^2)^{1/2}\right) - A\,(h)\,. \tag{7.2}$$

In this case

$$MK(B) = d_B(S, 0) = \delta(B, \theta_{t_B}) = A(h) + B(l). \tag{7.3}$$

On the other hand, for every $t < t_B = l$ one should have

$$\delta(B, \theta_t) \geq \delta(B, \theta_{t_B}),$$

which implies, in view of (7.1) and (7.3), that

$$B(t) \geq A(h) + B(l) - A\left((h^2 + (l-t)^2)^{1/2}\right).$$

Plugging (7.2) into the latter estimate, we get

$$B(t) \geq A\left((h^2 + l^2)^{1/2}\right) - A\left((h^2 + (l-t)^2)^{1/2}\right), \qquad t < l. \tag{7.4}$$

Summing up, we conclude that a monotone nondecreasing l.s.c. function B satisfying $B(0) = 0$ and verifying simultaneously (7.2) and (7.4) solves Problem 2 in our situation. Since of course (7.2) and (7.4) can be simultaneously satisfied, we infer that the optimal pricing policies are exactly those $B \in \mathcal{B}$ for which both (7.2) and (7.4) hold. The minimum optimal pricing policy is therefore the function $B_{\min}$ given by the formula

$$B_{\min}(t) := A\left((h^2 + l^2)^{1/2}\right) - A\left((h^2 + (l-t)^2)^{1/2}\right), \qquad t \in [0, l].$$

We consider now some particular cases.

Example 7.1. If $A(t) = t$, then a function $B \in \mathcal{B}$ solves Problem 2 if and only if

$$\begin{array}{rcl} B(l) & = & (h^2 + l^2)^{1/2} - h, \\ B(t) & \geq & (h^2 + l^2)^{1/2} - (h^2 + (l-t)^2)^{1/2}, \qquad t < l. \end{array}$$

The minimum optimal solution to Problem 2 is given then by the continuous concave function (see Figure 2)

$$B_{\min}(t) := (h^2 + l^2)^{1/2} - (h^2 + (l-t)^2)^{1/2}, \qquad t \leq l.$$

Example 7.2. If $A(t) = t^2$, then a function $B \in \mathcal{B}$ solves Problem 2 if and only if

$$\begin{array}{rcl} B(l) & = & l^2, \\ B(t) & \geq & 2lt - t^2, \qquad t < l. \end{array}$$

The minimum optimal solution to Problem 2 is given in this case by the continuous concave function (see Figure 3)

$$B_{\min}(t) := 2lt - t^2, \qquad t \leq l.$$

Note that in this case $B_{\min}(t) > A(t)$ for all $t \in (0, l)$, showing that Corollary 6.4 is sharp.

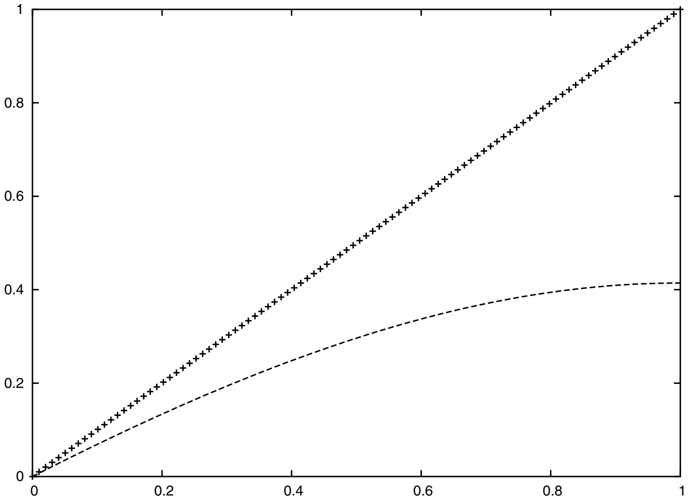


FIG. 2. *Graph of A (dotted line) and of the minimum optimal pricing policy $B_{\min}$ (dashed line) for $A(t) := t$ and $h = l = 1$.*

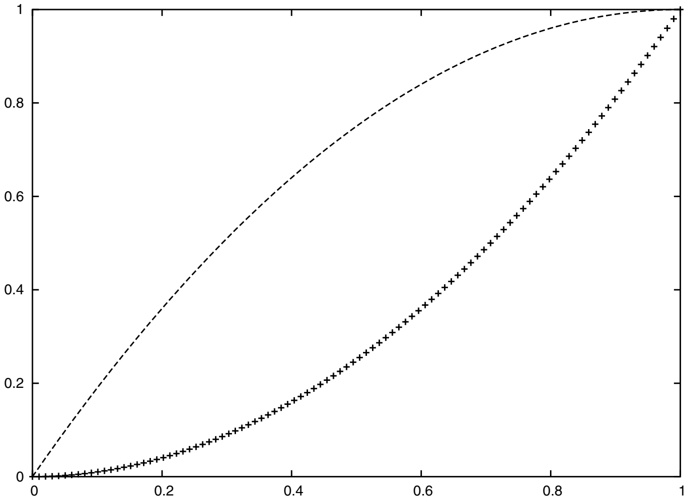


FIG. 3. *Graph of A (dotted line) and of the minimum optimal pricing policy $B_{\min}$ (dashed line) for $A(t) := t^2$ and $h = l = 1$.*

7.2. Two citizens. Let Ω and Σ be as in the above paragraph, while $\varphi_1 := 2\delta_0$ and $\varphi_0 := \delta_{S_1} + \delta_{S_2}$, where $S_i = (l, h_i)$, $i = 1, 2$. This model corresponds to the case of a population represented by two citizens living in S_1 and S_2 but having only a single workplace (or service) O. Here, as in the single citizen case, there is only one admissible transport plan $\gamma = \varphi_0 \otimes \varphi_1$, hence every solution to Problem 2 is also a solution to Problem 1, and vice versa.

Clearly, for a passenger moving between S_i and O it is convenient to choose the polygonal path θ_{t_i} consisting of a line segment connecting S_i with a point on Σ which has coordinates $(t_i, 0)$, $t_i \leq l$ (which is the part of the itinerary in which the passenger moves by own means), and a line segment connecting the latter point to O (which is the part of the itinerary in which the passenger moves with the help of the transportation network), so that the total cost for the passenger is

$$\delta(B,\theta_{t_i}) = A\left((h_i^2+(l-t_i)^2)^{1/2}\right) + B(t_i). \tag{7.5}$$

Therefore, each t_i is chosen as the greatest number in $[0,l]$ minimizing $\delta(B,\theta_{t_i})$. On the other hand, let

$$B_i(t) := A\left((h_i^2+l^2)^{1/2}\right) - A\left((h_i^2+(l-t)^2)^{1/2}\right), \qquad t\in[0,l], \tag{7.6}$$

and mind that according to (7.4) each B_i is the minimum optimal solution for Problem 2 (and also for Problem 1) with $\varphi_1=\delta_0$ and $\varphi_0=\delta_{S_i}$, i.e., for the single citizen case. With this notation one has

$$\delta(B,\theta_{t_i}) = B(t_i) - B_i(t_i) + A\left((h_i^2+l^2)^{1/2}\right),$$

and since the last term in the right-hand side of the above expression is independent of t_i, we get that each t_i is actually chosen as the greatest real number $t\in[0,l]$ minimizing $B(t)-B_i(t)$. This gives $t_i=t_i(B)$. The income F of the owner of the transportation network is then calculated by

$$F(B,t_1(B),t_2(B)) := B(t_1(B)) + B(t_2(B)). \tag{7.7}$$

To simplify the calculations, assume from now on that $A(t)=t$. Let $h_1\le h_2$, so that a direct calculation shows $B_1(t)\ge B_2(t)$ and $B_1'(t)\ge B_2'(t)$ for all $t\in[0,l]$. The first important observation is that $t_1\ge t_2$. In fact, otherwise one would have

$$B(t_1)-B_1(t_1) < B(t_2)-B_1(t_2), \text{ while } B(t_2)-B_2(t_2)\le B(t_1)-B_2(t_1),$$

and hence

$$\begin{aligned}\int_{t_1}^{t_2} B_1'(\tau)\,d\tau &= B_1(t_2)-B_1(t_1) < B(t_2)-B(t_1)\\ &\le B_2(t_2)-B_2(t_1) = \int_{t_1}^{t_2} B_2'(\tau)\,d\tau,\end{aligned}$$

which is impossible due to the fact that $B_1'(t)\ge B_2'(t)$ for all $t\in[0,l]$.

Note now that if B is an optimal pricing policy, then $B(t)\ge B_2(t)$ for all $t\in[0,l]$, since otherwise $B\vee B_2$ would give a strictly better pricing policy. On the other hand, one has $B(t)\le B_2(t)$ for all $t\in[0,t_2]$, since otherwise the citizen living in S_2 would not cover the distance t_2 with the use of the transportation network. Therefore, $B(t)=B_2(t)$ for all $t\in[0,t_2]$.

Our next observation is that $t_1=l$. In fact, if $t_1<l$, then $B(t)-B_1(t) > B(t_1)-B_1(t_1)$ for each $t_1<t\le l$. But then, taking instead of B the new pricing policy $\tilde{B}$ defined by

$$\tilde{B}(t) := \begin{cases} B(t) & \text{if } 0\le t\le t_1,\\ B(t_1)+B_1(t)-B_1(t_1) & \text{if } t_1<t\le l,\end{cases}$$

the effect will be that $t_1(\tilde{B}) = l > t_1(B)$ (i.e., that the citizen living in S_1 will move the maximum possible distance l with the help of the transportation network, rather than $t_1(B)$), while in the meantime $t_2(\tilde{B}) = t_2(B)$ (i.e., the citizen living in S_2 will still move the distance $t_2(B)$ with the help of the network). Therefore, the gain of the owner of the transportation network becomes

$$\begin{array}{rcl} F(\tilde{B}, t_1(\tilde{B}), t_2(\tilde{B})) & = & \tilde{B}(t_2(B)) + \tilde{B}(l) \\ & = & B(t_2(B)) + B(t_1) + B_1(l) - B_1(t_1(B)) \\ & = & F(B, t_1(B), t_2(B)) + (B_1(l) - B_1(t_1(B))) \\ & > & F(B, t_1(B), t_2(B)). \end{array}$$

The latter contradiction with the optimality of B shows then that $t_1 = l$.

We note now that if B is the minimum optimal pricing policy, then

$$B(t) = B(t_2) + B_1(t) - B_1(t_2)$$

for all $t \in [t_2, l]$. In fact, with this choice every path θ_t with $t \in [t_2, l]$ gives the same value of the cost $\delta(B, \theta_t)$ to the citizen living in S_1 (so that $t_1(B) = l$ only because of our assumption that each person chooses the maximum possible distance to cover using of the transportation network as long as the total cost for him is the same).

Summing up, we arrive at a conclusion that the minimum optimal pricing policy in the situation we are studying is given by

$$B(t) = \begin{cases} B_2(t) & \text{if } 0 \le t \le t_2, \\ B_2(t_2) + B_1(t) - B_1(t_2) & \text{if } t_2 < t \le l. \end{cases} \tag{7.8}$$

According to (7.7), minding that $t_1(B) = l$, the income of the owner of the transportation network then becomes

$$F(B, t_1(B), t_2(B)) = 2B_2(t_2) + B_1(l) - B_1(t_2).$$

Therefore, t_2 has to maximize $2B_2(t) - B_1(t)$ among all $t \in [0, l]$, and hence, by (7.6), it maximizes the function

$$f(t) := \left(h_1^2 + (l-t)^2\right)^{1/2} - 2\left(h_2^2 + (l-t)^2\right)^{1/2}$$

among all $t \in [0, l]$. A straightforward computation ensures $f'(t) > 0$ whenever $3(l-t)^2 > h_2^2 - 4h_1^2$, and $f'(t) < 0$ in the case of the opposite inequality. One can therefore consider three possible cases.

Case 1. $4h_1^2 \ge h_2^2$. Then $t_2 = l$ and hence the minimum optimal pricing policy is $B(t) = B_2(t)$ for all $t \in [0, l]$. Both citizens therefore use the transportation network to the full length l.

Case 2. $h_2^2 - 3l^2 < 4h_1^2 < h_2^2$. Then

$$t_2 = l - \left(\frac{h_2^2 - 4h_1^2}{3}\right)^{1/2}$$

and the minimum optimal pricing policy B is given by (7.8) (see Figure 4).

Case 3. $4h_1^2 \le h_2^2 - 3l^2$. Then $t_2 = 0$ (so that the citizen living in S_2 does not use the network) and hence the minimum optimal pricing policy is $B(t) = B_1(t)$ for all $t \in [0, l]$.

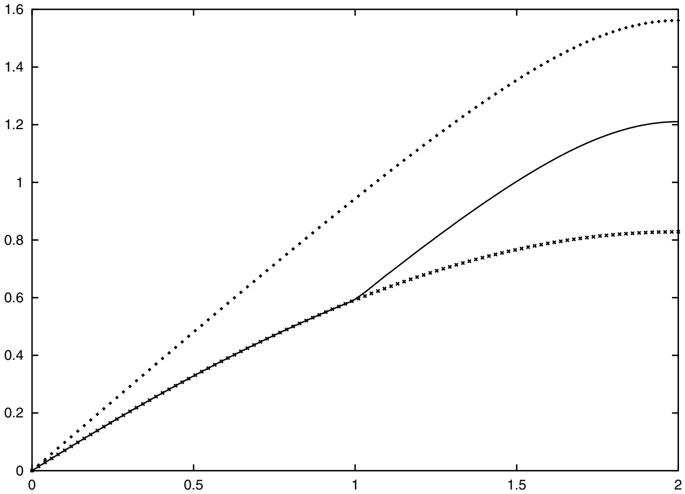


FIG. 4. *Graphs of B_1 (upper dotted line), B_2 (lower dotted line), and B (solid line) for $l = 2$, $h_1 = 1/2$, and $h_2 = 2$, in which case $t_2 = 1$.*

REFERENCES

[1] L. AMBROSIO, *Lecture notes on optimal transport problems*, in Mathematical Aspects of Evolving Interfaces, Lecture Notes in Math. 1812, Springer-Verlag, New York, 2003, pp. 1–52.

[2] L. AMBROSIO AND P. TILLI, *Topics on Analysis in Metric Spaces*, Oxford Lecture Ser. Math. Appl. 25, Oxford University Press, New York, 2004.

[3] A. BRAIDES AND A. DEFRANCESCHI, *Homogenization of Multiple Integrals*, Oxford Lecture Ser. Math. Appl. 12, University Press, Oxford, New York, 1998.

[4] C. CASTAING AND M. VALADIER, *Convex Analysis and Measurable Multifunctions*, Lecture Notes in Math. 580, Springer-Verlag, New York, 1977.

[5] L. C. EVANS AND W. GANGBO, *Differential equations methods for the Monge-Kantorovich mass transfer problem*, Mem. Amer. Math. Soc., 137 (1999), no. 653.

[6] L. C. EVANS, *Partial differential equations and Monge-Kantorovich mass transfer*, in Current Developments in Mathematics, International Press, Cambridge, MA, 1997, pp. 65–126.

[7] S. T. RACHEV AND L. RÜSCHENDORF, *Mass Transportation Problems. Vol.* I. *Theory*, Probab. Appl. (NY), Springer-Verlag, New York, 1998.

[8] S. T. RACHEV AND L. RÜSCHENDORF, *Mass Transportation Problems. Vol.* II. *Applications*, Probab. Appl. (NY), Springer-Verlag, New York, 1998.

[9] S. M. SRIVASTAVA, *A Course on Borel Sets*, Grad. Texts in Math. 180, Springer-Verlag, New York, 1998.

SIAM J. OPTIM.
Vol. 16, No. 3, pp. 854–870

REFINEMENTS OF STATIONARY POINTS WITH APPLICATIONS TO NONCOOPERATIVE GAMES AND ECONOMICS[*]

GERARD VAN DER LAAN[†], DOLF TALMAN[‡], AND ZAIFU YANG[§]

Abstract. It is well known that any continuous function f defined on a nonempty compact and convex set X has a stationary point. In many circumstances there may exist multiple stationary points and some of them may be undesirable from the viewpoint of stability. In this paper we introduce a new method of eliminating those undesirable stationary points while at the same time retaining some desirable stationary points. The main idea of refining the concept of stationary point is to perturb simultaneously both the domain set X, by taking a sequence of sets in the (relative) interior of X converging to X, and the solution concept, by replacing the concept of stationary point by a coincidence point with some well-defined mapping. If a stationary point is the limit of a sequence of coincidence points, we say that the stationary point is stable with respect to this sequence of subsets of X and the coincidence mapping. It is shown that stable stationary points exist for a large class of perturbations. A stable point is said to be normal-stable if we take the normal cone as the coincidence mapping, implying that any coincidence point on a subset in the sequence is a stationary point of f on this subset. It is shown that a normal-stable stationary point always exists for any sequence of subsets which starts from an interior point and converges to X in a continuous way. Special cases of normal-stability are perfect stationary points and robust stationary points. In addition, several practical applications of these new concepts are provided.

Key words. stationary point, stability, refinement, perturbation, equilibrium

AMS subject classifications. 49D35, 54C60, 90A14, 90C30, 90C33

DOI. 10.1137/04060799X

1. Introduction.

Let X be a subset of the n-dimensional Euclidean space $\mathbb{R}^n$ and let f be a function from X to $\mathbb{R}^n$. Then a *stationary point* or solution to the variational inequality problem with respect to f is a point x^* in X satisfying

$$(x^* - x)^\top f(x^*) \geq 0 \quad \text{for all } x \in X. \tag{1.1}$$

It is well known that a stationary point exists if f is a continuous function and X is a nonempty convex, compact set. The concept of stationary point has many important applications in various fields. For instance, in noncooperative game theory, economic equilibrium theory, fixed point theory, nonlinear optimization theory and engineering, a stationary point gives a solution to the problem under investigation. In many of these applications the multiplicity of stationary points may ask for a refined solution concept; see, for example, van Damme [2] and Kehoe [7]. Although the conditions for guaranteeing the existence of a stationary point are rather weak, conditions for guaranteeing the existence of a unique stationary point are often very demanding and usually not satisfied. For instance, in game-theoretic and economic applications

[*] Received by the editors May 11, 2004; accepted for publication (in revised form) June 29, 2005; published electronically January 12, 2006.
http://www.siam.org/journals/siopt/16-3/60799.html

[†] Department of Econometrics and Tinbergen Institute, Free University, De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands (glaan@feweb.vu.nl).

[‡] Department of Econometrics & Operations Research and CentER, Tilburg University, P.O. Box 90153, 5000 Tilburg, The Netherlands (talman@uvt.nl).

[§] Faculty of Business Administration, Yokohama National University, Yokohama 240-8501, Japan (yang@ynu.ac.jp). This work was done while this author was a research fellow of the Alexander von Humboldt Foundation, at the Institute of Mathematical Economics, University of Bielefeld, Bielefeld, Germany.

there can be any finite (odd) number of equilibria, being stationary points of some specific function, and there may even exist higher-dimensional sets of equilibria. Then a refinement may reduce the number of stationary points or equilibria considerably by requiring additional properties to be satisfied.

Within the field of noncooperative game theory are two well known refinements of Nash equilibria for games being mixed extensions of games with a finite number of pure strategies: the *perfect* equilibria introduced by Selten [18] and the *proper* equilibria of Myerson [12]. Both refinements are based on small perturbations of the strategy space, being the Cartesian product of unit simplices. The set of perfect equilibria is a nonempty subset of the set of Nash equilibria and the set of proper equilibria is a nonempty subset of the set of perfect equilibria. Motivated by this study in game theory, van der Laan, Talman, and Yang [9] and Yang [22] investigated the stability of stationary points of functions on polytopes. They extended the concept of properness of equilibria to that of a robust stationary point for arbitrary (continuous) functions on polytopes and also developed a simplicial algorithm for computing robust stationary points. By their algorithms the refined Nash equilibria, such as perfect or proper Nash equilibria, can be efficiently computed. For further results on the stability of (Nash) equilibria, one may refer to, e.g., Kohlberg and Mertens [8], Kajii and Morris [6], and Ui [20].

On the other hand, in applied mathematics and operations research, stability, semistability, and strong stability of solutions for nonlinear programming, systems of nonlinear equations, and variational inequality problems have been studied from a different angle; see, e.g., Pang and Ralph [15], Facchinei and Pang [4], and Robinson [16] and references therein. They investigate solutions that survive against certain perturbations of the function and then identify classes of functions for which the concerned problem has a stable solution. So, when applied to noncooperative games, such a perturbation could possess other or even more desirable properties than those of perfect or proper Nash equilibria. However, not every game or problem has a a stable solution. For example, in van Damme [2] it is shown that essential Nash equilibria, where the marginal payoff function is perturbed, may not exist. Also evolutionary stable equilibria may fail to exist; see Weibull [21]. In the spirit of Selten [18] and Myerson [12], we do not perturb the function but want to refine the concept of stationary point in such way that the refinement always exists in the case when the function f is continuous and its domain X is nonempty, convex, and compact.

The aim of this paper is to study the refinement of stationary points within a general framework. We propose a general method of eliminating stationary points that fail to survive against a sequence of specific perturbations and is such that at least one stationary point survives. The stationary points that survive against these perturbations will be called *stable stationary points*. The main idea of the refinement is to perturb both the domain and the solution concept of stationary point. The domain X will be perturbed by taking a sequence of sets in the (relative) interior of X converging to X, while the concept of stationary point will be replaced by a coincidence point. The refinement depends both on the way the sequence of subsets of X is chosen and the way in which the coincidence mapping is defined. For both choices there are many possibilities. The only restrictions will be that a coincidence point exists on each subset of the sequence and that every convergent (sub)sequence of coincidence points converges to a stationary point on X. Such a stationary point being the limit of a sequence of coincidence points on a sequence of subsets is then called stable with respect to the underlying sequence of subsets and the chosen coincidence mapping. Given the way the sequence of subsets and coincidence mapping are chosen, an induced stable stationary point has additional properties that other stationary points may not. This provides the possibility of eliminating stationary points not having certain desirable additional properties. In the case when we take as coincidence mapping the normal cone, we call a stable stationary point a *normal-stable* stationary point for the chosen sequence of subsets converging to X.

We also introduce the natural refinements of perfectness and robustness and show that a normal-stable stationary point satisfies these respective properties when the sequence of subsets is appropriately chosen. When applied to noncooperative games being mixed extensions of games with a finite number of pure strategies, the set of perfect stationary point appears to coincide with the set of perfect Nash equilibria and any robust stationary point appears to yield a proper mixed strategy Nash equilibrium. For symmetric two-player games the existence of a proper symmetric (mixed-strategy) equilibrium follows from the existence of a robust stationary point. We further apply the concept of stable stationary point to the *replicator dynamics* in the field of evolutionary game theory. It is well known that the set of equilibria typically is a strict subset of the set of stationary points of the replicator dynamics; see Weibull [21]. By taking an appropriate coincidence mapping, we are able to refine the stationary points of the replicator dynamics in such a way that every stable stationary point with respect to this coincidence mapping is an equilibrium. Moreover, it is shown that such a stable stationary point always exists. This result is in sharp contrast to many solution concepts in evolutionary game theory that may fail to exist under the same conditions. In this application the coincidence mapping is not the normal cone, and thus the stable point is not normal-stable. Finally, we apply the refinements of perfectness and robustness to a general equilibrium model with constant returns to scale production technologies.

The rest of the paper is organized as follows. In section 2 we introduce the concepts of stability and normal-stability and we provide existence proofs. The concepts of perfectness and robustness are discussed in section 3. Section 4 discusses several applications in noncooperative games, evolutionary games, and exchange economies.

2. Stable stationary points. Let X be a given nonempty compact, convex subset of the n-dimensional Euclidean space $\mathbb{R}^n$. It is well known that every continuous function f from X to $\mathbb{R}^n$ has at least one solution to the variational inequality problem (1.1); see Eaves [3] and Hartman and Stampacchia [5]. In the case of a point-to-set mapping ϕ from X to the collection of nonempty subsets of $\mathbb{R}^n$ a solution to the variational inequality problem, where in (1.1) the vector $f(x^*)$ should be an element of the set $\phi(x^*)$, exists if the mapping ϕ is upper semicontinuous and, for all $x \in X$, $\phi(x)$ is a convex and compact subset of $\mathbb{R}^n$; see Yang [23]. In this paper we restrict ourselves to continuous functions. However, all results can also be generalized straightforwardly to upper semicontinuous point-to-set mappings.

In what follows, Aff (X) denotes the affine hull of X. Without loss of generality we may assume that for some integer m, $0 \leq m \leq n$, Aff (X) is an $(n-m)$-dimensional subspace of $\mathbb{R}^n$ and can be written as

$$\text{Aff}\ (X) = \{x \in \mathbb{R}^n \mid C^\top x = d\},$$

where C is an $n \times m$ matrix having full rank m and d is an m-vector. In the case when m is equal to 0, the set X is full-dimensional and we define Aff $(X) = \mathbb{R}^n$. The linear subspace Y denotes the set $Y = \{y \in \mathbb{R}^n \mid y = C\nu,\ \nu \in \mathbb{R}^m\}$. Notice that $Y = \{0^n\}$ if $m = 0$, where 0^n denotes the n-vector of zeros.

Given an m-dimensional compact and convex subset Z of Aff (X), let

$$N(Z, x) = \{y \in \mathbb{R}^n \mid y^\top x \geq y^\top x' \text{ for all } x' \in Z\}$$

denote the normal cone of Z at $x \in Z$. It holds that $N(Z, \cdot)$ is an upper semicontinuous mapping on Z. Moreover, for every $x \in Z$ the set $N(Z, x)$ is a closed and convex cone containing the set Y, and $N(Z, x) = Y$ when x lies in the relative interior of Z. Clearly, $x^* \in Z$ is a stationary point of a function f from Z to $\mathbb{R}^n$ if and only if $f(x^*) \in N(Z, x^*)$. For a point-to-set mapping ϕ on Z the latter condition becomes $\phi(x^*) \cap N(Z, x^*) \neq \emptyset$.

As discussed in the introduction, there can be more than one or even an infinite number of solutions to the variational inequality problem (1.1). In this section we introduce a general refinement concept, which selects a nonempty subset of the set of stationary points, giving a certain stability property to the stationary points within this subset. The general idea is to perturb both the set X and the concept of stationary point in such a way that every convergent subsequence of generalized stationary points converges to a solution of the variational inequality problem. A solution that is not the limit of any such subsequence is not stable with respect to the chosen perturbations. To guarantee the existence of a stable stationary point it is sufficient to assume that a generalized stationary point exists on any perturbed subset and that there exists a convergent subsequence of generalized stationary points converging to a stationary point.

To describe formally the idea of refinement we introduce two mappings: $\mathcal{X}$ and G. The mapping $\mathcal{X}\colon [0,1] \to X$ defines the perturbation of the set X and has to satisfy the following two conditions, where Int A denotes the interior of a set $A \subset X$ with respect to the set Aff (X):

(X1) The mapping $\mathcal{X}\colon [0,1] \to X$ is continuous and for each $\epsilon \in [0,1]$ the set $\mathcal{X}(\epsilon)$ is a nonempty, convex, and compact subset of X.

(X2) $\mathcal{X}(0) = X$ and $\mathcal{X}(\epsilon') \subset \text{Int}\mathcal{X}(\epsilon)$ for every $0 \leq \epsilon < \epsilon' \leq 1$.

For example, let X be described by the set $\{x \in \mathbb{R}^n \mid h(x) \leq 0, C^\top x = d\}$ for some convex function h from $\mathbb{R}^n$ to $\mathbb{R}$. Notice that such a function h always exists, since X is compact and convex. Then we may take $\mathcal{X}(\epsilon) = \{x \in \mathbb{R}^n \mid h(x) \leq -\omega\epsilon, C^\top x = d\}$, where $\omega > 0$ is such that $\mathcal{X}(1) \neq \emptyset$. Another possibility is to take $\mathcal{X}(\epsilon) = \epsilon\{v\} + (1-\epsilon)X$ for some point v in the relative interior of X. Notice that due to both conditions (X1) and (X2) it holds that for every $x \in X \setminus \mathcal{X}(1)$ there exists a unique ϵ, $0 \leq \epsilon < 1$, such that x lies in the relative boundary of $\mathcal{X}(\epsilon)$.

For a given mapping $\mathcal{X}$ satisfying conditions (X1) and (X2), the second mapping $G\colon X \to \mathbb{R}^n$ defines the concept of generalized stationary point on each set $\mathcal{X}(\epsilon)$. This mapping has to satisfy the following three conditions, where Bnd A denotes the relative boundary of a set $A \subset X$ with respect to Aff (X):

(G1) G is upper semicontinuous on X and for each $x \in X$ the set $G(x)$ is a convex, closed cone in $\mathbb{R}^n$ containing Y.

(G2) For every $x \in$ Bnd $\mathcal{X}(\epsilon)$ and $y \in N(\mathcal{X}(\epsilon), x) \setminus \{0^n\}$, $0 < \epsilon < 1$, there exists $w \in G(x)$ such that $y^\top w > 0$.

(G3) For every $x \in$ Bnd X it holds that $G(x) \subseteq N(X, x)$.

The conditions say that when x lies in the boundary of $\mathcal{X}(\epsilon)$ the set $G(x)$ is a cone containing Y and points in the same direction as the normal cone $N(\mathcal{X}(\epsilon), x)$ in the sense that for every nonzero element of $N(\mathcal{X}(\epsilon), x)$ there is an element in $G(x)$ making a positive angle with it. Moreover, $G(x)$ is a subset of the normal cone if x lies in the boundary of X.

Definition 2.1. *A pair* $(\mathcal{X}, G)$ *of mappings is* regular *when it satisfies conditions* (X1), (X2), (G1), (G2), *and* (G3).

Given a pair $(\mathcal{X}, G)$ and some ϵ, $0 \leq \epsilon < 1$, let the mapping $G^\epsilon \colon \mathcal{X}(\epsilon) \to \mathbb{R}^n$ be defined by

$$G^\epsilon(x) = \begin{cases} Y & \text{when } x \in \text{Int } \mathcal{X}(\epsilon) \\ G(x) & \text{when } x \in \text{Bnd } \mathcal{X}(\epsilon). \end{cases}$$

For a function f from X to $\mathbb{R}^n$ and a pair $(\mathcal{X}, G)$, we now define an ϵ-stable stationary point of f as follows.

Definition 2.2. *Given a pair $(\mathcal{X}, G)$, for some ϵ, $0 \leq \epsilon < 1$, a point $x \in X$ is an* ϵ-stable stationary point *of f with respect to $(\mathcal{X}, G)$ if $x \in \mathcal{X}(\epsilon)$ and $f(x) \in G^\epsilon(x)$.*

An ϵ-stable stationary point x of f with respect to $(\mathcal{X}, G)$ is a coincidence point of the function f with the mapping G^ϵ and therefore either lies in the interior of $X(\epsilon)$ and is a stationary point of f on X or lies in the boundary of $\mathcal{X}(\epsilon)$ and $f(x)$ is an element of $G(x)$. For this reason the mapping G is called a coincidence mapping. Notice that for $\epsilon = 0$ an ϵ-stable stationary point of f is a stationary point of f on X, also when x lies in the boundary of X.

Definition 2.3. *A stationary point x^* of a function f from X to $\mathbb{R}^n$ is* stable with respect to $(\mathcal{X}, G)$ *(abbreviated* $(\mathcal{X}, G)$-stable*) if there exists a sequence of positive numbers $(\epsilon_k)_{k \in \mathbb{N}}$ with limit 0 such that x^* is the limit of a sequence of ϵ_k-stable stationary points of f with respect to $(\mathcal{X}, G)$ for k going to infinity.*

An $(\mathcal{X}, G)$-stable stationary point x^* lies either in the interior of X or in the boundary of X and in every small neighborhood of x^* there exists a point x in the interior of X such that $f(x) \in G(x)$. Note that the process of refining stationary points can be naturally regarded as a dynamic process or an iterative process; see van der Laan, Talman, and Yang [9]. See Saari [17] for iterative price mechanisms.

The next theorem states that every continuous function on X has an $(\mathcal{X}, G)$-stable stationary point when $(\mathcal{X}, G)$ is regular.

Theorem 2.4. *Let f be a continuous function from a nonempty convex, compact set X in $\mathbb{R}^n$ to $\mathbb{R}^n$ and let $(\mathcal{X}, G)$ be a regular pair of mappings. Then there exists an $(\mathcal{X}, G)$-stable stationary point of f on X.*

Proof. First we prove that for every ϵ, $0 < \epsilon < 1$, an ϵ-stable stationary point of f with respect to $(\mathcal{X}, G)$ exists. For some ϵ, $0 < \epsilon < 1$, it follows from (G1) that the mapping G^ϵ is upper semicontinuous and that, for any $x \in \mathcal{X}(\epsilon)$, $G^\epsilon(x)$ is a nonempty, convex, and closed set. Since for all $x \in X$ the set $G(x)$ is a cone, condition (G2) implies that for any $x \in$ Bnd $\mathcal{X}(\epsilon)$ and $y \in N(\mathcal{X}(\epsilon), x)$ there exists $w \in G^\epsilon(x)$ satisfying $y^\top w \geq y^\top f(x)$. From Fan's coincidence theorem applied to the mappings $\{f(\cdot)\}$ and $G^\epsilon(\cdot)$ restricted to the nonempty, convex, and compact set $\mathcal{X}(\epsilon)$ it follows that there exists a coincidence point x^ϵ in $\mathcal{X}(\epsilon)$ satisfying $f(x^\epsilon) \in G^\epsilon(x^\epsilon)$. Hence, $f(x^\epsilon) \in Y$ if $x^\epsilon \in$ Int $\mathcal{X}(\epsilon)$ and $f(x^\epsilon) \in G(x^\epsilon)$ if $x^\epsilon \in$ Bnd $\mathcal{X}(\epsilon)$; i.e., x^ϵ is an ϵ-stable stationary point of f with respect to $(\mathcal{X}, G)$.

Now take any sequence of positive numbers ϵ_k, $k \in \mathbb{N}$, smaller than one, converging to zero, and for every $k \in \mathbb{N}$ let x^k be an ϵ_k-stable stationary point of f with respect to $(\mathcal{X}, G)$. Since X is compact, without loss of generality we may assume that the sequence $(x^k)_{k \in \mathbb{N}}$ is convergent and converges to some x^* in X. Hence, x^* is the limit of a sequence of ϵ_k-stable stationary points of f with respect to $(\mathcal{X}, G)$ for ϵ_k converging to zero when k goes to infinity. We still have to prove that x^* is a stationary point of f. If x^* lies in the interior of X, then, because of the continuity of $\mathcal{X}$ and the properties of the mapping $\mathcal{X}$ given in (X2), the point x^k lies in the interior of $\mathcal{X}(\epsilon_k)$ for k large enough, which implies that $f(x^k) \in Y$ for k large enough, and therefore $f(x^*) \in Y$; i.e., x^* is a stationary point of f. If x^* lies in the boundary of

X, we may assume without loss of generality that for every $k \in \mathbb{N}$ the point x^k lies in the boundary of $\mathcal{X}(\epsilon_k)$. Therefore, for every $k \in \mathbb{N}$ it holds that $f(x^k) \in G(x^k)$. Since f is continuous, G is upper semicontinuous, and x^k converges to x^*, we have that $f(x^*) \in G(x^*)$. Condition (G3) then implies that $f(x^*) \in N(X, x^*)$; i.e., x^* is a stationary point of f. □

The theorem implies that, for every given regular $(\mathcal{X}, G)$, any function f satisfying the same conditions under which a stationary point is known to exist has a stationary point being stable with respect to $(\mathcal{X}, G)$. Of course the reverse does not hold. Not every stationary point needs to be a stable stationary point with respect to a chosen pair $(\mathcal{X}, G)$. Notice that any interior stationary point is stable with respect to any pair, but that the stability of a stationary point on the boundary of X depends on the chosen pair and it even may be that a stationary point is not stable for any pair. So, for any regular pair $(\mathcal{X}, G)$, the set of stable points is a nonempty subset of the set of stationary points.

In the remainder of this section we consider the special case that the mapping G is the normal cone mapping; i.e., for given mapping $\mathcal{X}$, $G(x) = N(\mathcal{X}(\epsilon), x)$ if x lies in the boundary of $\mathcal{X}(\epsilon)$ for some ϵ, $0 \leq \epsilon < 1$, and $G(x) = \mathbb{R}^n$ when $x \in \mathcal{X}(1)$. When G is the normal cone mapping, we say that a stable stationary point with respect to $(\mathcal{X}, G)$ is normal-stable with respect to $\mathcal{X}$. Observe that an ϵ-stable point of f with respect to $(\mathcal{X}, G)$ is a stationary point of f on $\mathcal{X}(\epsilon)$ when G is the normal cone mapping.

DEFINITION 2.5. *For $f\colon X \to \mathbb{R}^n$ and a pair $(\mathcal{X}, G)$, an $(\mathcal{X}, G)$-stable point is* normal-stable *with respect to $\mathcal{X}$ (abbreviated $\mathcal{X}$* normal-stable*) if G is the normal cone mapping with respect to $\mathcal{X}$.*

The next theorem shows the regularity of a pair $(\mathcal{X}, G)$ for the normal cone mapping G.

THEOREM 2.6. *Let the mapping $\mathcal{X}$ satisfy* (X1) *and* (X2) *and let G be the normal cone mapping with respect to $\mathcal{X}$. Then the pair $(\mathcal{X}, G)$ is regular.*

Proof. Clearly, G satisfies conditions (G2) and (G3). It remains to show that G also satisfies condition (G1). For each $x \in X$, $G(x)$ is a nonempty, convex, and closed cone in $\mathbb{R}^n$ containing the linear subspace Y. So, we need only to show that G is upper semicontinuous on X. By definition, G is upper semicontinuous on $\mathcal{X}(1)$. Take any $y \in X \setminus \mathcal{X}(1)$. Let $(y^k)_{k\in\mathbb{N}}$ be a sequence of points in X converging to y and let $(g^k)_{k\in\mathbb{N}}$ be a sequence satisfying $g^k \in G(y^k)$ for all $k \in \mathbb{N}$ and converging to some g. Since $y \notin \mathcal{X}(1)$, we may assume without loss of generality that for all $k \in \mathbb{N}$ it holds that $y^k \in X \setminus \mathcal{X}(1)$. Let ϵ, $0 \leq \epsilon < 1$, be such that $y \in$ Bnd $\mathcal{X}(\epsilon)$. Due to conditions (X1) and (X2) on $\mathcal{X}$ there exists a unique sequence of nonnegative numbers $(\epsilon_k)_{k\in\mathbb{N}}$ converging to ϵ and satisfying that $y^k \in$ Bnd $\mathcal{X}(\epsilon_k)$ for all $k \in \mathbb{N}$. To show that $g \in G(y)$, take any x in $\mathcal{X}(\epsilon)$. Then, again according to conditions (X1) and (X2) there exists a sequence $(x^k)_{k\in\mathbb{N}}$ satisfying $x^k \in \mathcal{X}(\epsilon_k)$ for all $k \in \mathbb{N}$ and converging to x. Since $x^k \in \mathcal{X}(\epsilon_k)$ and $g^k \in G(y^k) = N(\mathcal{X}(\epsilon_k), y^k)$, we have for all $k \in \mathbb{N}$ that

$$x^{k\top} g^k \leq y^{k\top} g^k.$$

Taking the limits on both sides for k going to infinity, with x being the limit of x^k, y being the limit of y^k, g being the limit of g^k, we obtain that

$$x^\top g \leq y^\top g.$$

Since x is an arbitrary point in $\mathcal{X}(\epsilon)$, we obtain that $g \in N(\mathcal{X}(\epsilon), y) = G(y)$, showing that G is upper semicontinuous on X, and thus G satisfies (G1). Hence the pair $(\mathcal{X}, G)$ is regular. □

The next corollary follows immediately from Theorems 2.4 and 2.6.

Corollary 2.7. *Let f be a continuous function from a nonempty, convex, and compact set X to $\mathbb{R}^n$ and let $\mathcal{X} : [0,1] \to X$ be a mapping satisfying* (X1) *and* (X2)*. Then f has an $\mathcal{X}$ normal-stable stationary point on X.*

The corollary implies that there always exists a stationary point which is the limit point of a sequence of stationary points restricted to $\mathcal{X}(\epsilon_k)$, $k \in \mathbb{N}$, with a $\lim_{k\to\infty} \epsilon_k = 0$. Of course, again it holds that the set of normal-stable points is a subset of the set of stationary points and depends on the mapping $\mathcal{X}$.

3. Perfect and robust stationary points. Two well known refinements of Nash equilibria for games with a finite number of pure strategies are the concepts of perfect and proper equilibria. These concepts are refinements of stationary points on the mixed strategy space, which is the Cartesian product of several unit simplices; see section 4. In this section we generalize these concepts to sets X of a more general type and then show the existence by applying Corollary 2.7 for a specifically chosen mapping $\mathcal{X}$. In the following, for $k \in \mathbb{N}$, let I_k denote the set of the first k positive integers. In many situations the set X is of the form

$$X = \{x \in \mathbb{R}^n \mid \max\{h_1(x), h_2(x), \ldots, h_l(x)\} \le 0,\ C^\top x = d\} \tag{3.1}$$

for some $l \in \mathbb{N}$, where, for every $i \in I_l$, the function $h_i(\cdot)$ is a continuously differentiable convex function on $\mathbb{R}^n$, with gradient at x denoted by $g^i(x)$. In what follows in this section we consider such sets. We assume that none of the constraints $h_i(x) \le 0$, $i \in I_l$, is redundant and that X is simple, in the sense that for any $I \subseteq I_l$ it holds that the set $F(I) = \{x \in X \mid h_i(x) = 0,\ i \in I\}$ is either empty or homeomorphic to an $(n - m - |I|)$-dimensional convex, compact set. Notice that $F(\emptyset) = X$. Then for a point x in the boundary of X it holds that the normal cone of X at x equals

$$N(X,x) = \left\{ y \in \mathbb{R}^n \;\middle|\; y = \sum_{i\in I} \mu_i g^i(x) + \sum_{j\in I_m} \nu_j c^j, \mu_i \ge 0,\ j \in I,\ \nu_j \in \mathbb{R},\ j \in I_m \right\},$$

where $I = \{i \in I_l \mid h_i(x) = 0\}$ and c^j, $j \in I_m$, is the jth column of the matrix C. Let $\mathcal{I}$ be the collection of subsets I of I_l satisfying that the set $F(I)$ is nonempty. Clearly, $x \in X$ is a stationary point of a function f from X to $\mathbb{R}^n$ if and only if there exists $I \in \mathcal{I}$ such that $h_i(x) = 0$ for all $i \in I$ and

$$f(x) = \sum_{i\in I} \mu_i g^i(x) + \sum_{j\in I_m} \nu_j c^j$$

for some $\mu_i \ge 0$ for $i \in I$ and $\nu_j \in \mathbb{R}$ for $j \in I_m$.

We now define a perfect stationary point on X as the limit of a sequence of ϵ-perfect points.

Definition 3.1. *Let f be a function from X to $\mathbb{R}^n$, where X is of the form* (3.1)*.*

(i) *For $\epsilon > 0$, a point $x \in X$ is an ϵ-*perfect stationary point *of f if $x \in$ Int X and there exists $I \in \mathcal{I}$ satisfying $f(x) = \sum_{i\in I} \mu_i g^i(x) + \sum_{j\in I_m} \nu_j c^j$ for some $\mu_i > 0$ for $j \in I$, $\nu_j \in \mathbb{R}$ for $j \in I_m$, and $h_i(x) \ge -\epsilon$ for all $i \in I$.*

(ii) *A point $x^* \in X$ is a* perfect stationary point *of f if there exists a sequence of positive numbers $(\epsilon_k)_{k\in\mathbb{N}}$ with limit 0 such that x^* is the limit of a sequence of ϵ_k-perfect stationary points of f for k going to infinity.*

The next theorem shows that a perfect stationary point always exists if f is a continuous function by proving that a perfect stationary point is a normal-stable stationary point when the mapping $\mathcal{X}$ is chosen in an appropriate way.

THEOREM 3.2. *Let f be a continuous function from a nonempty, convex, and compact set X of the form* (3.1) *to $\mathbb{R}^n$. Then f has a perfect stationary point on X.*

Proof. Define the mapping $\mathcal{X}$ by

$$\mathcal{X}(\epsilon) = \{x \in \mathbb{R}^n \mid \max\{h_1(x), h_2(x), \ldots, h_l(x)\} \leq \omega\epsilon,\ C^\top x = d\},$$

where $\omega > 0$ is such that $\mathcal{X}(1) \neq \emptyset$, and take G to be the normal cone mapping. Since X is simple and no constraints are redundant for X, ω can be chosen in such a way that for every ϵ, $0 \leq \epsilon \leq 1$, the set $\mathcal{X}(\epsilon)$ is simple, has no redundant constraints, and the collection $\mathcal{I}$ of nonempty subsets I of I_k, satisfying that the set $\{x \in X(\epsilon) \mid h_i(x) = \omega\epsilon$ for all $i \in I\}$ is nonempty, is the same as for $\epsilon = 0$. Clearly, for this ω, $\mathcal{X}$ satisfies (X1) and (X2) and it follows from Corollary 2.7 that f has an $\mathcal{X}$ normal-stable point x^* on X. It remains to show that x^* is a perfect stationary point of f. Since x^* is an $\mathcal{X}$ normal-stable stationary point of f, there exists a sequence $(\eta_k)_{k\in\mathbb{N}}$ of positive numbers smaller than one converging to zero and a sequence of η_k-stable stationary points x^k of f with respect to $\mathcal{X}$ such that the sequence $(x^k)_{k\in\mathbb{N}}$ converges to x^*. Since x^k is an η_k-stable stationary point of f with respect to $\mathcal{X}$ and G is the normal cone mapping, x^k is a stationary point of f on $\mathcal{X}(\eta_k)$, and so for all $k \in \mathbb{N}$ there exists $I^k \in \mathcal{I}$ satisfying $f(x^k) = \sum_{i\in I^k} \mu_i g^i(x^k) + \sum_{j\in I_m} \nu_j c^j$ for some $\mu_i > 0$ for $i \in I^k$, $\nu_j \in \mathbb{R}$ for $j \in I_m$, and $h_i(x^k) = -\omega\eta_k$ for all $i \in I^k$. Therefore, x^k is an ϵ_k-perfect stationary point of f with $\epsilon_k = \omega\eta_k$. Since the sequence $(\eta_k)_{k\in\mathbb{N}}$ converges to zero, the sequence $(\epsilon_k)_{k\in\mathbb{N}}$ also converges to zero, which proves that x^* is a perfect stationary point of f. □

Observe that the definition of an ϵ-stable stationary point x is slightly weaker than what is shown in the proof. In (i) of Definition 3.1 it is required only that $h_i(x) \geq -\epsilon$ if $\mu_i > 0$; i.e., x lies at most ϵ away from all the boundary constraints $h_i(y) = 0$ for which the gradients $g^i(x)$ generate the cone containing $f(x)$. In the proof we show that $h_i(x) = -\epsilon$ if $\mu_i > 0$.

Example 1. Let X be the two-dimensional unit ball $B = \{x \in \mathbb{R}^2 \mid \ \| x \|_2 \ \leq 1\}$ and let the function $f\colon B \to \mathbb{R}^2$ be given by $(f_1(x), f_2(x)) = (x_1 + 1, x_2)$. Clearly, a point $x^* \in \operatorname{Bnd} B$ is a stationary point of f if and only if $f(x^*) = \lambda x^*$ for some $\lambda \geq 0$. The function f has two stationary points of f and both lie in the boundary of B: $(-1, 0)$ with function value $f(-1, 0) = (0, 0)$ and $(1, 0)$ with function value $f(1, 0) = (2, 0)$. However, only $(1, 0)$ is a perfect stationary point of f, since for any ϵ, $0 < \epsilon < 1$, the point $((1-\epsilon)^{1/2}, 0)$ is the unique stationary point of f on $\mathcal{X}(\epsilon)$.

Example 2. In this example we consider the case that $h_i(x) = a^{i\top}x - b^i$, where $a^i \in \mathbb{R}^n \setminus \{0^n\}$ and $b_i \in \mathbb{R}$, for all $i \in I_l$. So, X is a polytope P in $\mathbb{R}^n$ described in polyhedral form by

$$P = \{x \in \mathbb{R}^n \mid a^{i\top}x - b_i \leq 0 \text{ for all } i \in I_l,\ C^\top x = d\}.$$

We assume that none of the constraints is redundant. For each subset I of I_l, let

$$F(I) = \{x \in P \mid a^{i\top}x = b_i \text{ for all } i \in I\}.$$

Note that $F(\emptyset) = P$. Then $I \in \mathcal{I}$ if and only if $F(I)$ is not empty. It is assumed that P is simple in the sense that for every $I \in \mathcal{I}$ the dimension of face $F(I)$ is equal to $n - m - |I|$. Finally, for $I \in \mathcal{I}$, define

$$A(I) = \left\{ y \in \mathbb{R}^n \,\middle|\, y = \sum_{i\in I} \mu_i a^i + \sum_{j\in I_m} \nu_j c^j,\ \mu_i \geq 0 \text{ for all } i \in I,\ \nu_j \in \mathbb{R}, \text{for all } j \in I_m \right\}.$$

Notice that $A(\emptyset) = Y$ and that $A(I) = N(X, x)$ if $x \in \text{Int } F(I)$. Hence, we have the following straightforward but important observation that $x^* \in P$ is a stationary point of a function f from P to $\mathbb{R}^n$ if and only if there exists $I^* \in \mathcal{I}$ satisfying $x^* \in F(I^*)$ and $f(x^*) \in A(I^*)$; see, e.g., Talman and Yamamoto [19] and Burke and Moré [1]. A stationary point x^* of f is perfect if x^* is the limit of a sequence of points x^k in the interior of P satisfying that there exists a sequence of positive numbers $(\epsilon_k)_{k\in\mathbb{N}}$ with limit 0 such that for each $k \in \mathbb{N}$ there is a set $I^k \in \mathcal{I}$ with $f(x^k) \in A(I^k)$ and $h_i(x^k) = a^{i\top} x^k \geq b_i - \epsilon_k$ for all $i \in I^k$.

For the case of X being a unit simplex the concept of a robust stationary point was introduced by Yang [22, 23] and was generalized for the case of a polytope by van der Laan, Talman, and Yang [9]. Here we generalize robustness to the case of sets X of the form (3.1). It appears that an $\mathcal{X}$ normal-stable stationary point is a robust stationary point for a specifically chosen mapping $\mathcal{X}$.

Definition 3.3. *Let f be a function from X to $\mathbb{R}^n$, where X is of the form* (3.1).

(i) *For $\epsilon > 0$, a point $x \in X$ is an* ϵ-robust stationary point *of f if $x \in$ Int X and there exists $I \in \mathcal{I}$ such that $f(x) = \sum_{i\in I} \mu_i g^i(x) + \sum_{j\in I_m} \nu_j c^j$ for some $\mu_i > 0$ for $i \in I$ and $\nu_j \in \mathbb{R}$ for $j \in I_m$ satisfying $h_i(x) \geq -\epsilon$ for all $i \in I$ and $h_i(x) \geq \epsilon h_j(x)$ whenever $\mu_i > \mu_j$.*

(ii) *A point $x^* \in X$ is a* robust stationary point *of f if there exists a sequence of positive numbers $(\epsilon_k)_{k\in\mathbb{N}}$ with limit 0 such that x^* is the limit of a sequence of ϵ_k-robust stationary points of f for k going to infinity.*

A point x is an ϵ-robust stationary point of f for some $\epsilon > 0$ if there exists an index set I such that $f(x)$ lies in the cone generated by Y and the vectors $g^i(x)$, $i \in I$, and simultaneously x is both ϵ-perfect and lies at least ϵ times closer to the ith boundary constraint $\{y \mid h_i(y) = 0\}$ than to the jth boundary constraint $\{y \mid h_j(y) = 0\}$ whenever $\mu_i > \mu_j$. So, this condition strengthens the condition for perfectness, which requires only that x is at most ϵ away from the ith boundary constraint when $\mu_i > 0$. Notice that the concepts of properness and perfectness coincide when $l = 1$.

Theorem 3.4. *Let f be a continuous function from a nonempty, convex, and compact set X of the form* (3.1) *to $\mathbb{R}^n$. Then f has a robust stationary point on X.*

Proof. Without loss of generality we assume that $l \geq 2$. For $I \in \mathcal{I}$ and ϵ, $0 < \epsilon \leq 1$, define the function h_I and the number $r_I(\epsilon)$ by

$$h_I(x) = \sum_{i\in I} h_i(x) \text{ and } r_I(\epsilon) = -\omega \sum_{j=n+1-|I|}^{n} \left(\frac{\epsilon}{2}\right)^j$$

for some $\omega > 0$. Then define the mapping $\mathcal{X}$ by

$$\mathcal{X}(\epsilon) = \{x \in \mathbb{R}^n \mid h_I(x) \leq r_I(\epsilon),\ I \in \mathcal{I},\ C^\top x = d\}, \quad \epsilon \in [0, 1]. \tag{3.2}$$

Since X is simple and has no redundant constraints, there exists $\omega > 0$ such that for every ϵ, $0 < \epsilon < 1$, the set $\mathcal{X}(\epsilon)$ is simple, has no redundant constraints, and has always the same collection $\mathcal{I}$. Clearly, for this ω the mapping $\mathcal{X}$ satisfies conditions (X1) and (X2), and thus f has an $\mathcal{X}$ normal-stable stationary point x^* on X. We will show that x^* is a robust stationary point of f. Since x^* is normal-stable with respect to $\mathcal{X}$, there exists a sequence $(\eta_k)_{k\in\mathbb{N}}$ of positive numbers smaller than one converging to zero and a sequence of η_k-stable stationary points x^k of f with respect to $\mathcal{X}$ and the normal cone mapping G, such that $(x^k)_{k\in\mathbb{N}}$ converges to x^*. Analogous with the proof in van der Laan, Talman, and Yang [9] in case X is a polytope P, and by using the fact that G is the normal cone mapping, it follows from the construction of the mapping $\mathcal{X}$ that for every $k \in \mathbb{N}$ there exists $I^k \in \mathcal{I}$ such that

$$f(x^k) = \sum_{i \in I^k} \mu_i g^i(x^k) + \sum_{j \in I_m} \nu_j c^j$$

for some $\mu_i > 0$ for all $i \in I^k$ and satisfying that

$$h_i(x^k) \geq -\omega\eta_k \text{ for all } i \in I^k$$

and for any pair of indices i and j in I^k,

$$h_i(x^k) \geq \frac{\eta_k}{2} h_j(x^k) \text{ when } \mu_i > \mu_j.$$

Hence, for every $k \in \mathbb{N}$ it holds that

$$h_i(x^k) \geq -\epsilon_k \text{ when } \mu_i > 0$$

and

$$h_i(x^k) \geq \epsilon_k h_j(x^k) \text{ when } \mu_i > \mu_j,$$

where $\epsilon_k = \eta_k \max\{\omega, \ \frac{1}{2}\}$. Therefore, x^k is an ϵ_k-robust stationary point of f for all $k \in \mathbb{N}$. Since the sequence $(\eta_k)_{k \in \mathbb{N}}$ converges to zero, the sequence $(\epsilon_k)_{k \in \mathbb{N}}$ also converges to zero, which proves that x^* is a robust stationary point of f. □

Example 3. Let X be the unit simplex in $\mathbb{R}^3$, i.e., $X = \{x \in \mathbb{R}^3 \mid -x_j \leq 0, \ j = 1, 2, 3 \text{ and } x_1 + x_2 + x_3 = 1\}$. Let the function $f\colon X \to \mathbb{R}^3$ be given by $f(x) = Ax$, where

$$A = \begin{bmatrix} 11 & 10 & 1 \\ 10 & 10 & 3 \\ 1 & 3 & 3 \end{bmatrix}.$$

There are three stationary points, namely the three unit vectors $e^1 = (1, 0, 0)^\top$, $e^2 = (0, 1, 0)^\top$, and $e^3 = (0, 0, 1)^\top$. Only the first two survive the criterion of perfectness. Clearly, when we take $x = (\epsilon, \epsilon, 1 - 2\epsilon)^\top$, then $f_1(x) = 1 + 19\epsilon$, $f_2(x) = 3 + 14\epsilon$ and $f_3(x) = 3 - 2\epsilon$, so that for ϵ small enough $f(x)$ cannot be written as a linear combination of $g^1(x) = (-1, 0, 0)^\top$, $g^2(x) = (0, -1, 0)^\top$, and $c = (1, 1, 1)^\top$ with nonnegative weights for $g^1(x)$ and $g^2(x)$, and therefore e^3 is not perfect. The second stationary point is not robust. When we take $x = (\epsilon, 1 - \epsilon - \epsilon^2, \epsilon^2)^\top$, then $f_1(x) = 10 + \epsilon - 9\epsilon^2$, $f_2(x) = 10 - 7\epsilon^2$, and $f_3(x) = 3 - 2\epsilon$ so that for ϵ small enough $f_1(x) > f_2(x)$, and thus $f(x)$ cannot be written as a linear combination of $g^1(x) = (-1, 0, 0)^\top$, $g^3(x) = (0, 0, -1)^\top$, and $c = (1, 1, 1)^\top$ with nonnegative weights for $g^1(x)$ and $g^3(x)$, and therefore e^2 is not robust.

4. Applications.

4.1. Noncooperative games in normal form.

We first consider mixed extensions of noncooperative games, in which each player has a finite number of pure strategies, in the following *finite games.* Let there be N players. Player j, $j \in I_N$, can choose from n_j different actions in the set A^j. If player j, $j \in I_N$, chooses action a_j, then the payoff to player i, $i \in I_N$, is equal to some number $u_i(a)$, where $a = (a_1, \ldots, a_N)$ is an element of the action space $A = \Pi_{j \in I_N} A^j$. Each player j, $j \in I_N$, can randomize the choice of actions by taking a strategy $x^j = (x^j_1, \ldots, x^j_{n_j})$ in the $(n_j - 1)$-dimensional unit simplex $S^{n_j} = \{x^j \in \mathbb{R}^{n_j} \mid x^j_i \geq 0,\, i \in I_{n_j},\, \sum_{i=1}^{n_j} x^j_i = 1\}$, where x^j_k, $k \in I_{n_j}$, denotes the probability with which player j chooses the kth action. The Cartesian product of the strategy sets S^{n_j}, $j \in I_N$, is the strategy set of the game and is denoted by the simplotope S with typical element $x = (x^1, \ldots, x^N)$. Clearly, S is a simple polytope with dimension equal to $n - N$, where n is the total number of actions in the game, i.e., $n = \sum_{j=1}^N n_j$, and can be written as a polyhedron of the form (3.1) by

$$S = \left\{ x \in \mathbb{R}^n \;\middle|\; -x^j_k \leq 0 \text{ for all } j,k,\; \sum_{k=1}^{n_j} x^j_k = 1, \text{ for all } j \right\}. \tag{4.1}$$

For $x \in S$, $v_j(x)$ denotes the expected payoff for player j, $j \in I_N$, when strategy x is being played, i.e.,

$$v_j(x) = \sum_{a \in A} \Pi_{i \in I_N} x^i_{a_i} u_j(a),$$

and $f^j_k(x)$ denotes the marginal payoff for player j, $j \in I_N$, when player j chooses action k, $k \in A^j$, and the other players play according to strategy x, i.e.,

$$f^j_k(x) = \sum_{\{a \in A \mid a_j = k\}} \Pi_{i \neq j} x^i_{a_i} u_j(a).$$

We now recall the following definitions, where (x^j, x^{*-j}) denotes the strategy vector x^* with x^{*j} replaced by x^j.

Definition 4.1.

1. (Nash [13]) *A strategy* $x^* \in S$ *is a* Nash equilibrium *if for every* $j \in I_N$ *it holds that* $v_j(x^*) \geq v_j(x^j, x^{*-j})$ *for all* $x^j \in S^{n_j}$.
2. (Selten [18]) *A strategy* $x^* \in S$ *is a* perfect (Nash) equilibrium *if it is the limit of a sequence of* ϵ_k*-perfect equilibria for a sequence of positive numbers* ϵ_k, $k \in \mathbb{N}$, *converging to zero, where a strategy* x *is an* ϵ*-perfect equilibrium if* $x \in \text{Int } S$ *and* $x^j_k \leq \epsilon$ *whenever* $f^j_k(x) < \max_h f^j_h(x)$.
3. (Myerson, [12]) *A strategy* $x^* \in S$ *is a* proper (Nash) equilibrium *if it is the limit of a sequence of* ϵ_k*-proper equilibria for a sequence of positive numbers* ϵ_k, $k \in \mathbb{N}$, *converging to zero, where a strategy* x *is an* ϵ*-proper equilibrium if* $x \in$ Int S *and* $x^j_k \leq \epsilon x^j_h$ *whenever* $f^j_k(x) < f^j_h(x)$.

Clearly, $x \in S$ is a Nash equilibrium if and only if x is a stationary point of the marginal payoff function f on S, i.e., with $I = \{(j,k) | x^j_k = 0\}$, $f(x) = \sum_{(j,k) \in I} \mu_{j,k} a^{j,k} + \sum_{j=1}^N \nu_j c^j$, where $\nu_j = \max_h f^j_h(x)$ for all $j \in I_N$ and $\mu_{j,k} = \max_h f^j_h(x) - f^j_k(x) \geq 0$ for all $(j,k) \in I$, letting for ease of notation $(j,k) = \sum_{i=1}^{j-1} n_i + k$. A perfect equilibrium $x \in S$ is the limit of a sequence of completely mixed strategies at which the nonoptimal actions are chosen with arbitrarily small probability. So, x is a perfect equilibrium if and only if x is a perfect stationary point of the function f. A proper equilibrium is the limit of a sequence of completely mixed strategies at which the lower the marginal payoff of an action of a player is, the smaller the probability should be with which this player chooses that action. Every proper equilibrium is a perfect equilibrium and every perfect equilibrium is a Nash equilibrium.

In the literature, properness is known to be the most refined concept of Nash equilibrium that still exists for every noncooperative game in normal form. The concept of robustness, as introduced on a polytope in the previous section, requires more than properness.

DEFINITION 4.2. *A strategy $x^* \in S$ is a* robust (Nash) equilibrium *if it is the limit of a sequence of ϵ_k-robust equilibria for a sequence of positive numbers ϵ_k, $k \in \mathbb{N}$, converging to zero, where a strategy x is called an ϵ-robust equilibrium if $x \in \operatorname{Int} S$ and $x^j_k < \epsilon x^i_l$ whenever $\max_h f^j_h(x) - f^j_k(x) > \max_h f^i_h(x) - f^i_l(x)$.*

The definition implies that the worse an action in the game is, the smaller the probability should be with which that action is chosen. So, strengthening the properness conditions, the robustness condition requires that the probability of an action decreases by at least a factor ϵ if the marginal payoff becomes worse, is taken over all players simultaneously instead of each player separately.

THEOREM 4.3. *Any noncooperative game in normal form has a robust equilibrium and every robust equilibrium is a proper Nash equilibrium.*

Proof. Theorem 3.4 says that the marginal payoff function f has a robust stationary point x^*. Hence, x^* is the limit of a sequence of ϵ_k-robust stationary points x^k of f for ϵ_k going to zero. Taking into account the form (4.1) of S, it follows that if x^k is an ϵ_k-robust stationary point of f, then x^k is an ϵ_k-robust equilibrium. Therefore, x^* is a robust Nash equilibrium. $\square$

In Yang [23] a simplicial variable dimension algorithm is described to compute a robust Nash equilibrium for finite games. Clearly, any robust equilibrium is proper. For finite games, it is an open question whether there may exist proper equilibria that are not robust. In fact, we conjecture that for finite games any proper equilibrium is also robust.[1] However, in the more general case of games with a continuum of strategies the more general concept of robustness may be useful to eliminate undesirable equilibria.

Next, we consider the class of symmetric finite two-player games. Such a game can be summarized by two $n \times n$ payoff matrices A and B with $B = A^\top$, where n is the number of pure strategies for both players 1 and 2. In the following we denote such a symmetric bimatrix game by the matrix A. Given a mixed strategy pair $(x^1, x^2) \in S$, where $S = S^n \times S^n$, the payoff of the players is then given by $v_1(x^1, x^2) = x^{1\top} A x^2$ for player 1 and $v_2(x^1, x^2) = x^{1\top} A^\top x^2$ for player 2. For this class of games in particular symmetric equilibria are of special importance. A Nash equilibrium (x^1, x^2) is called *symmetric* if $x^1 = x^2$. A (mixed) strategy x is called an *equilibrium strategy* if (x, x) is a symmetric Nash equilibrium. It is shown by Nash [14] that every symmetric bimatrix game has a symmetric equilibrium. In the literature it is taken for granted that if a symmetric game has an equilibrium of some kind (such as Nash, perfect, or proper, and so on), then the game has a symmetric equilibrium of that kind. Nevertheless, for example, the existence of a symmetric proper Nash equilibrium in a symmetric bimatrix game has never been shown in the literature. Here we demonstrate this result by applying Theorem 3.4. In fact we show the existence of a robust equilibrium strategy.

THEOREM 4.4. *Any symmetrix bimatrix game has a symmetric proper Nash equilibrium.*

Proof. Given a symmetric bimatrix game A, we define the function $f\colon S^n \to \mathbb{R}^n$ by

$$f(x) = Ax.$$

[1] This conjecture was raised by one of the referees.

For $\epsilon \in (0,1)$, let $x \in S^n$ be a completely mixed strategy such that

$$x_k \leq \epsilon x_l \text{ if } f_k(x) < f_l(x) \text{ for all } k,\ l \in I_N.$$

Then the pair $(x,x) \in S$ is a symmetric ϵ-proper Nash equilibrium, since (x,x) satisfies the conditions of an ϵ-proper equilibrium given in Definition 4.1 with $f^1 = f^2 = f$.

Next, take $X = S^n$ and the mapping $\mathcal{X}$ as defined in formula (3.2), where the set X is given by the polyhedral set S^n written in the form (3.1) by

$$X = S^n = \left\{ x \in \mathbb{R}^n \;\middle|\; -x_k \leq 0 \text{ for all } k \in I_n, \quad \sum_{k=1}^{n} x_k = 1 \right\}.$$

Then Theorem 3.4 says that f has a robust stationary point x^*. Since any stationary point of f is an equilibrium strategy, it follows that x^* is a robust equilibrium strategy. Since by definition, x^* is the limit of a sequence of ϵ_k-robust stationary points x^k of f on S^n with ϵ_k going to zero, (x^k, x^k) is a symmetric ϵ_k-proper Nash equilibrium as defined in Definition 4.1 with $f^1 = f^2 = f$. Hence (x^*, x^*) is a symmetric proper Nash equilibrium. □

4.2. Replicator and price dynamics. Symmetric games have appeared to be very important in evolutionary game theory, in which individuals are repeatedly drawn from a large monomorphic population to play a symmetric two-player game. In evolutionary game theory, the evolutionary stable strategy proposed by Maynard Smith and Price [11] and Maynard Smith [10], is probably the most well-known concept. A strategy $x \in S^n$ is said to be an *evolutionary stable strategy* (ESS), if for any strategy $y \neq x$ in S^n there exists some $\epsilon_y \in (0,1)$ such that for all $\epsilon \in (0, \epsilon_y)$ it holds that

$$x^\top Aw > y^\top Aw \text{ where } w = \epsilon y + (1-\epsilon)x.$$

Clearly this implies that $x^\top Ax \geq x^\top Ay$ for all y so that any ESS is an equilibrium strategy, but not the reverse. So, the concept of ESS is a refinement of the concept of equilibrium strategy. However, the existence of an ESS is not guaranteed for every symmetric bimatrix game; see, e.g., van Damme [2] and Weibull [21]. In this subsection we first introduce a new refinement of equilibrium strategy, called a sign-stable equilibrium, and then show that every symmetric bimatrix game has such a sign-stable equilibrium. For $j = 1, \ldots, n$, define

$$z_j(x) = A_j x - x^\top Ax, \qquad x \in S^n.$$

In evolutionary game theory, $z_j(x)$ denotes the excess fitness of action j at mixed strategy x, being the marginal payoff of action j minus the average payoff over all actions at strategy x, where A_j denotes the jth column of the matrix A. We assume that $z : S^n \to \mathbb{R}^n$ is a continuous function. Notice that for every $x \in S^n$ it holds that $x^\top z(x) = 0$. Clearly, x^* is an equilibrium strategy if and only if x^* is a stationary point of z, i.e., $z(x^*) \leq 0^n$, and $z_j(x^*) < 0$ implies that $x_j^* = 0$.

The probability x_j is interpreted to be the fraction of players using action j within a monomorphic population of a large number of players. So, the fitness can be seen as the difference of the (expected) payoff of a player of population j and the expected payoff in the population as a whole. It is further assumed that players with a higher fitness get more offspring, resulting in the so-called replicator dynamics given by

$$dx(t)/dt = f(x(t)), \qquad t \geq 0,$$

with $f\colon S^n \to \mathbb{R}^n$ given by

$$f_j(x) = x_j z_j(x), \qquad j = 1, \ldots, n.$$

The replicator dynamics models the population dynamics. The function f has the property that $\sum_{j=1}^n f_j(x) = 0$ for any $x \in S^n$ so that the solution path of the replicator dynamics $dx(t)/dt = f(x(t))$ stays in S^n; see, for example, Weibull [21].

Each stationary point of z, and thus each equilibrium strategy, is a stationary point of the corresponding function f and is even a zero point of f. However, the reverse is not true. Not every stationary point of f is a stationary point of z. For example, all vertices of S^n are stationary points of f, but not all of them need to be equilibrium points. We will show that a sign-stable stationary point of the function f is a stationary point of z and therefore an equilibrium, where $x \in S^n$ is a *sign-stable stationary point* of f if it is the limit of a convergent subsequence of ϵ_k-sign-stable zero points of f for a sequence of positive real numbers $(\epsilon_k)_{k\in\mathbb{N}}$ with $\lim_k \epsilon_k = 0$. For $0 < \epsilon < 1$, a point $x \in$ Int S^n is called an ϵ-sign-stable zero point of f if $x_j \leq \frac{\epsilon}{n}$ when $f_j(x) < 0$ and $x_j \geq n^{-1}$ when $f_j(x) > 0$. In the context of a symmetric bimatrix game a sign-stable stationary point of f will be called *a sign-stable equilibrium strategy.* We will prove the existence of a sign-stable equilibrium strategy by defining the coincidence mapping G appropriately. It should be noticed that this mapping G is not the normal cone mapping.

In the following, $e(i)$ denotes the ith unit vector in $\mathbb{R}^n$ and e the n-vector of ones.

THEOREM 4.5. *Let $z : S^n \to \mathbb{R}^n$ be a continuous function satisfying $x^\top z(x) = 0$ for all $x \in S^n$ and let $f : S^n \to \mathbb{R}^n$ be defined by $f_j(x) = x_j z_j(x)$ for all $j \in I_n$ and $x \in S^n$. Then a sign-stable stationary point of f exists and every sign-stable stationary point of f is a stationary point of z.*

Proof. For ϵ, $0 \leq \epsilon \leq 1$, let

$$\mathcal{X}(\epsilon) = \left\{ x \in S^n \,\middle|\, \min_j x_j \geq \frac{\epsilon}{n} \right\}.$$

Clearly, $\mathcal{X}(\cdot)$ is a continuous mapping; $\mathcal{X}(0) = S^n$; for every ϵ, $0 \leq \epsilon \leq 1$, the set $X(\epsilon)$ is a nonempty, compact, and convex set; $\mathcal{X}(1) = \{\frac{e}{n}\}$; and $\mathcal{X}(\epsilon') \subset$ Int $\mathcal{X}(\epsilon)$ for every $0 \leq \epsilon < \epsilon' \leq 1$. For ϵ, $0 < \epsilon \leq 1$, and I being a proper subset of the set $I_n = \{1, \ldots, n\}$, the face $F^\epsilon(I)$ of $X(\epsilon)$ is given by $F^\epsilon(I) = \{x \in \mathcal{X}(\epsilon) \mid x_i = \frac{\epsilon}{n},\ i \in I\}$ and the normal cone $N(\mathcal{X}(\epsilon), x)$ at a point $x \in$ Int $F^\epsilon(I)$ is given by the set

$$A(I) = \left\{ y \in \mathbb{R}^n \,\middle|\, y = \nu e - \sum_{i \in I} \mu_i e(i),\ \nu \in \mathbb{R},\ \mu_i \geq 0,\ i \in I \right\}.$$

For $x \in S^n$, define $G(x) = \mathbb{R}^n$ if $x = \frac{1}{n}e$; otherwise

$$\begin{aligned} G(x) = \{y \in \mathbb{R}^n \mid \quad & y = w + \lambda e,\ \lambda \in \mathbb{R},\ \textstyle\sum_{i=1}^n w_i = 0, \\ & w_i \leq 0 \text{ if } x_i = \min_h x_h, \\ & w_i \geq 0 \text{ if } x_i = \max_h x_h, \\ & w_i = 0 \text{ otherwise}\}. \end{aligned}$$

Clearly, $G(\cdot)$ satisfies condition (G1). To show that $G(\cdot)$ satisfies condition (G2), take any $x \in F^\epsilon(I)$ and $y \in A(I) \setminus \{0^n\}$ for ϵ, $0 < \epsilon < 1$, with I being a proper subset of I_n. Thus $x_i = \min_h x_h$ for all $i \in I$ and $y = \nu e - \sum_{i \in I} \mu_i e(i)$ for some $\nu \in \mathbb{R}$ and $\mu_i \geq 0$, $i \in I$, not all equal to zero. If $\mu_i = 0$ for all $i \in I$ and therefore $\nu \neq 0$, take $z = \nu e$; then $z \in G(x)$ and $z^\top y = n\nu^2 > 0$. If $\mu_i > 0$ for some $i \in I$, take $z = e(j) - e(i)$ for some j with $x_j = \max_h x_h$; then $z \in G(x)$ and $z^\top y = \mu_i > 0$. Hence, $G(\cdot)$ satisfies condition (G2). With respect to (G3), if x is in the boundary of S^n, then $N(S^n, x) = A(I)$, where $I = \{i \mid x_i = 0\}$, and so $G(x) \subset N(S^n, x)$. From Theorem 2.4 it now follows that there exists an $(\mathcal{X}, G)$-stable stationary point of f on S^n.

Hence, for every ϵ, $0 < \epsilon < 1$, there exists $x^\epsilon \in P(\epsilon)$ satisfying $f(x^\epsilon) = \nu e$ for some $\nu \in \mathbb{R}$ if $x^\epsilon \in$ Int $\mathcal{P}(\epsilon)$ and $f(x^\epsilon) \in G(x^\epsilon)$ if $x^\epsilon \in$ Bnd $\mathcal{P}(\epsilon)$. Since $\sum_{i=1}^n f_i(x^\epsilon) = 0$ we obtain that $\nu = 0$ and so $f(x^\epsilon) = 0^n$ if $x^\epsilon \in$ Int $P(\epsilon)$. If $x^\epsilon \in$ Bnd $\mathcal{P}(\epsilon)$, then there exists $\delta(\epsilon) \geq n^{-1}$ such that $x_j^\epsilon = \frac{\epsilon}{n}$ if $f_j(x^\epsilon) < 0$, $x_j^\epsilon = \delta(\epsilon)$ if $f_j(x^\epsilon) > 0$, and $\frac{\epsilon}{n} \leq x_j^\epsilon \leq \delta(\epsilon)$ if $f_j(x^\epsilon) = 0$; i.e., x^ϵ is an ϵ-sign-stable zero point of f. Take any convergent subsequence $(x^{\epsilon_k})_{k\in\mathbb{N}}$ of such points with $\lim_k \epsilon_k = 0$ and let x^* be the limit of this subsequence. Suppose $z_j(x^*) < 0$ for some component j, then for large enough k it holds that $f_j(x^{\epsilon_k}) < 0$ and therefore $x^{\epsilon_k} = \frac{\epsilon_k}{n}$ for k large enough. Hence, after taking limits we obtain that $z_j(x^*) < 0$ implies $x_j^* = 0$. Since $n^{-1} \leq \delta(\epsilon_k) \leq 1$ for all $k \in \mathbb{N}$, we may assume without loss of generality that the sequence $(\delta(\epsilon_k))_{k\in\mathbb{N}}$ converges to some $\delta^* > 0$. This implies that $x_j^* > 0$ if $z_j(x^*) > 0$. Since $\sum_{j=1}^n f_j(x^*) = 0$ we get that $f_j(x^*) = 0$ for all $j \in I_n$, and therefore $z_j(x^*) = 0$ if $x_j^* > 0$. Hence, x^* is a stationary point of z. ☐

The theorem says that the replicator dynamics function f always has a sign-stable stationary point and that every sign-stable stationary point of f induces an equilibrium for the underlying function z. As a result, we have that every symmetric bimatrix game A has a sign-stable equilibrium strategy x.

4.3. A general equilibrium model with constant returns to scale. We consider a general equilibrium model with constant returns to scale production technologies. The model consists of k firms, l households, and n commodities. The aggregated excess demand for commodities by households at price vector $p \in \mathbb{R}^n_+ \setminus \{0^n\}$ is described by the vector $z(p)$. The function z satisfies continuity, homogeneity of degree zero and Walras' law, i.e., $p^\top z(p) = 0$ for any price vector p. Each firm $i \in I_k$ is characterized by an input-output or activity vector $a^i(p) \in \mathbb{R}^n$. At price vector p the vector $a^i(p)$ maximizes the net profit of firm i per unit of activity; i.e., $\pi_i(p) = p^\top a^i(p)$ is the net profit rate of firm i and satisfies continuity, convexity, homogeneity of degree one, and $\delta\pi_i(p)/\delta p_j = a^i_j(p)$ for all j. In case the activity level of firm i is equal to μ_i, the production vector at price vector p is equal to $\mu_i a^i(p)$ and the profit is equal to $\mu_i \pi^i(p) = \mu_i p^\top a^i(p)$.

An activity vector $\mu^* \in \mathbb{R}^k_+$ and a price vector $p^* \in \mathbb{R}^n_+ \setminus \{0^n\}$ constitute a (Walrasian) equilibrium if $z(p^*) = \sum_{i\in I_k} a^i(p^*)\mu_i^*$, $p^{*\top} a^i(p^*) \leq 0$ for all $i \in I_k$, and $\mu_i^* p^{*\top} a^i(p^*) = 0$ for all $i \in I_k$. This means that in equilibrium market demand is equal to market supply for every commodity, no firm can make a positive profit rate (otherwise the activity level of that firm would be infinity), and firms produce only if the profit rate is zero (profit of each firm is zero). It is well known that such an equilibrium always exists if the assumption of no production without input holds true.

Because of the homogeneity of degree zero of both the excess demand and the activity vectors the feasible price set can be taken to belong to the unit simplex S^n. Since in equilibrium no firm can make a positive profit, the set of feasible prices is given by the set $X \subset S^n$ of the form (3.1) given by

$$X = \{p \in \mathbb{R}^n_+ \mid \pi_i(p) \leq 0 \text{ for all } i \in I_k,\ e^\top p = 1\}.$$

Since there cannot be production without input and because of the properties of the profit functions, X is a nonempty, convex, and compact set in $\mathbb{R}^n$. Suppose that X is simple, no constraints are redundant, and X has dimension $n-1$. Then it follows immediately that $(p^*, \mu^*) \in S^n \times \mathbb{R}^k$ is an equilibrium if and only if p^* is a stationary point of the excess demand function z on the set X, satisfying $z(p^*) = \sum_{i\in I} \mu_i^* a^i(p^*)$, where $I = \{i \in I_k \mid \pi_i(p^*) = 0\}$. Notice that due to Walras' law at a stationary point the weight ν^* of the vector e is equal to zero.

It is well known that in an economy with constant returns to scale production technologies equilibria typically lie on the boundary of the feasible price set X and that there can be easily multiple equilibria; see, e.g., Kehoe [7]. The set of equilibria can be refined as described in the previous sections. For example, there exists a perfect (Walrasian) equilibrium, being the limit of a sequence of ϵ_k-perfect equilibria for ϵ_k converging to zero, where p is called an ϵ-perfect equilibrium if p lies in the interior of X and $z(p) = \sum_{i\in I_k} \mu_i a^i(p)$ is such that $\mu_i > 0$ implies $\pi_i(p) \geq -\epsilon$. This means that the loss (negative profit) rate of an active firm should be at most ϵ. Also, there exists a robust (Walrasian) equilibrium, being the limit of a sequence of ϵ_k-robust equilibria for ϵ_k converging to zero, where p is called an ϵ-robust equilibrium if p lies in the interior of X and $z(p) = \sum_{i\in I_k} \mu_i a^i(p)$ is such that $\mu_i > \mu_j$ implies $\pi_i(p) \geq \epsilon\pi_j(p)$. This means that the loss rate of a firm with a higher activity level should be at most ϵ times the loss rate of a firm with a lower activity level.

Acknowledgments. The third author wishes to thank the Alexander von Humboldt Foundation and the Institute of Mathematical Economics Bielefeld, for their support.

REFERENCES

[1] J. V. Burke and J. J. Moré, *Exposing constraints,* SIAM J. Optim., 4 (1994), pp. 573–595.

[2] E. E. C. van Damme, *Stability and Perfection of Nash Equilibria*, Springer-Verlag, Berlin, 1987.

[3] B. C. Eaves, *On the basic theory of complementarity,* Math. Programming, 1 (1971), pp. 68-75.

[4] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Vols. I, II, Springer, New York, 2003.

[5] P. Hartman and G. Stampacchia, *On some nonlinear elliptic differential functional equations,* Acta Math., 115 (1966), pp. 271–310.

[6] A. Kajii and S. Morris, *The robustness of equilibria to incomplete information,* Econometrica, 65 (1997), pp. 1283–1309.

[7] T. M. Kehoe, *Computation and multiciplicity of equilibria*, in Handbook of Mathematical Economics, Vol. IV, W. Hildenbrand and H. Sonnenschein, eds., North-Holland, Amsterdam, 1991, pp. 2049–2143.

[8] E. Kohlberg and J. F. Mertens, *On the strategic stability of equilibria,* Econometrica, 54 (1986), pp. 1003–1038.

[9] G. van der Laan, A. J. J. Talman, and Z. Yang, *Existence and approximation of robust solutions of variational inequality problems over polytopes,* SIAM J. Control Optim., 37 (1998), pp. 333–352.

[10] J. Maynard Smith, *Evolution and the Theory of Games*, Cambridge University Press, Cambridge, UK, 1982.

[11] J. Maynard Smith and G. R. Price, *The logic of animal conflict,* Nature, 246 (1973), pp. 15–18.

[12] R. B. Myerson, *Refinements of Nash equilibrium concepts,* Internat. J. Game Theory, 8 (1978), pp. 73–80.

[13] J. Nash, *Equilibrium points in N-person games,* Proc. Nat. Acad. Sci. U.S.A., 36 (1950), pp. 48–49.

[14] J. Nash, *Noncooperative games,* Ann. Math., 54 (1951), pp. 286–295.

[15] J.-S. Pang and D. Ralph, *Piecewise smoothness, local invertibility, and parametric analysis of normal maps,* Math. Oper. Res., 21 (1996), pp. 401–426.

[16] S. M. Robinson, *Localized normal maps and the stability of variational conditions,* Set-Valued Anal., 12 (2004), pp. 259–274.

[17] D. G. Saari, *Iterative price mechanisms,* Econometrics, 53 (1985), pp. 1117–1131.

[18] R. Selten, *Reexamination of the perfectness concept for equilibrium points in extensive games,* Internat. J. Game Theory, 4 (1975), pp. 25–55.

[19] A. J. J. Talman and Y. Yamamoto, *A simplicial algorithm for stationary point problems on polytopes,* Math. Oper. Res., 14 (1989), pp. 383–399.

[20] T. Ui, *Robust equilibria of potential games,* Econometrica, 69 (2001), pp. 1373–1380.

[21] J. W. Weibull, *Evolutionary Game Theory*, MIT Press, Cambridge, MA, 1995.

[22] Z. Yang, *A simplicial algorithm for computing robust stationary points of a continuous function on the unit simplex,* SIAM J. Control Optim., 34 (1996), pp. 491–506.

[23] Z. Yang, *Computing Equilibria and Fixed Points: The Solution of Nonlinear Inequalities*, Kluwer Academic Publishers, Boston, 1999.

SIAM J. Optim.
Vol. 16, No. 3, pp. 871–890

COMPLEX QUADRATIC OPTIMIZATION AND SEMIDEFINITE PROGRAMMING*

SHUZHONG ZHANG† AND YONGWEI HUANG†

Abstract. In this paper we study the approximation algorithms for a class of discrete quadratic optimization problems in the Hermitian complex form. A special case of the problem that we study corresponds to the max-3-cut model used in a recent paper of Goemans and Williamson [*J. Comput. System Sci.*, 68 (2004), pp. 442–470]. We first develop a closed-form formula to compute the probability of a complex-valued normally distributed bivariate random vector to be in a given angular region. This formula allows us to compute the expected value of a randomized (with a specific rounding rule) solution based on the optimal solution of the complex semidefinite programming relaxation problem. In particular, we present an $[m^2(1-\cos\frac{2\pi}{m})/8\pi]$-approximation algorithm, and then study the limit of that model, in which the problem remains NP-hard. We show that if the objective is to maximize a positive semidefinite Hermitian form, then the randomization-rounding procedure guarantees a worst-case performance ratio of $\pi/4 \approx 0.7854$, which is better than the ratio of $2/\pi \approx 0.6366$ for its counterpart in the real case due to Nesterov. Furthermore, if the objective matrix is real-valued positive semidefinite with nonpositive off-diagonal elements, then the performance ratio improves to 0.9349.

Key words. Hermitian quadratic functions, approximation ratio, randomized algorithms, complex semidefinite programming relaxation

AMS subject classifications. 90C20, 90C22

DOI. 10.1137/04061341X

1. Introduction.

The pioneering work of Goemans and Williamson [8] has caused a great deal of excitement in the field of optimization, as it used a new tool, semidefinite programming (SDP) in continuous optimization, through randomization and probabilistic analysis, to yield an excellent approximation ratio for a classical combinatorial optimization problem, known as the *max-cut* problem. This groundbreaking work has been extended in various ways since its first appearance. Among others, Frieze and Jerrum [6] extended the method to solve the general max-k-cut problem. Bertsimas and Ye [4] introduced another randomization scheme using normal distributions, to achieve the same approximation result as in Goemans and Williamson's original paper [8]. The Bertsimas–Ye analysis makes use of an important result in statistics, which states that the probability of a bivariate (2-dimensional) normally distributed random vector to be in the first orthant can be expressed analytically using elementary functions. This is impossible, however, for any dimension higher than three; see [1]. Recently, Goemans and Williamson [9] proposed another novel approach to solve the max-3-cut problem using the unit circle in the complex plane as a key modeling ingredient. In this paper we show that it is possible to compute the probability of the bivariate *complex-valued* normally distributed random vector to be in a specific angular region in $\mathbf{C}^2$ (see section 2). We then consider the following quadratic optimization problem in complex variables: maximize $z^{\mathrm{H}}Qz$, subject

*Received by the editors August 13, 2004; accepted for publication (in revised form) July 6, 2005; published electronically January 12, 2006. This research was supported by Hong Kong RGC earmarked grants CUHK4174/03E and CUHK418505.

http://www.siam.org/journals/siopt/16-3/61341.html

†Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, Hong Kong (zhang@se.cuhk.edu.hk, http://www.se.cuhk.edu.hk/∼zhang; ywhuang@se.cuhk.edu.hk).

to $z_k^m = 1$, $k = 1, \dots, n$, where z_k is a complex variable and is the kth component of the vector z, and $m \geq 2$ is an integer parameter of the model. Thanks to the new probability formula to be developed in section 2, we are able to compute the expected quality of a particular randomized solution for solving the above quadratic optimization model. The model of Goemans and Williamson for max-3-cut ($m = 3$) turns out to be a special case of this general model. It is interesting to study the limit of this model; that is, the case where $m \to \infty$ and the constraints become $|z_k| = 1$. It turns out that the problem remains NP-hard. However, the corresponding complex SDP relaxation yields an approximation ratio of $\pi/4 \approx 0.7854$, whereas for its counterpart in the real case the ratio is $2/\pi \approx 0.6366$ as shown by Nesterov [11]. If the off-diagonal elements of the objective matrix are real-valued and nonpositive, then the approximation ratio is actually 0.9349.

This paper is organized as follows. In section 2 we discuss the computation of the probability for the complex-valued normal distributions. In section 3 we apply the results developed in section 2 to solve complex-valued quadratic optimization problems. In particular, section 3.1 discusses the Hermitian quadratic function maximization problem, where the complex decision variables take discrete values. Section 3.2 presents an approximation algorithm for the problem. Section 3.3 considers the continuous version of the problem. Section 3.4 considers a special case where a sign restriction on the objective matrix is observed. Finally, we conclude the paper in section 4.

Notation. Throughout, we denote by $\bar{a}$ the conjugate of a complex number a and by $\mathbf{C}^n$ the space of n-dimensional complex vectors. For a given vector $z \in \mathbf{C}^n$, z^{H} denotes the conjugate transpose of z. The spaces of $n \times n$ real symmetric and complex Hermitian matrices are denoted by $\mathcal{S}^n$ and $\mathcal{H}^n$, respectively. For a matrix $Z \in \mathcal{H}^n$, we write Re Z and Im Z for the real and imaginary part of Z, respectively. Matrix Z being Hermitian implies that Re Z is symmetric and Im Z is skew-symmetric. We denote by $\mathcal{S}_+^n$ ($\mathcal{S}_{++}^n$) and $\mathcal{H}_+^n$ ($\mathcal{H}_{++}^n$) the cones of real symmetric positive semidefinite (positive definite) and complex Hermitian positive semidefinite (positive definite) matrices, respectively. The notation $Z \succeq (\succ 0)$ means that Z is positive semidefinite (positive definite). For two complex matrices Y and Z, their inner product $Y \bullet Z = \mathrm{Re}\ (\mathrm{tr}\ Y^{\mathrm{H}} Z) = \mathrm{tr}\ \left[(\mathrm{Re}\ Y)^{\mathrm{T}}(\mathrm{Re}\ Z) + (\mathrm{Im}\ Y)^{\mathrm{T}}(\mathrm{Im}\ Z)\right]$, where tr denotes the trace of a matrix and $^{\mathrm{T}}$ denotes the transpose of a matrix.

2. Complex bivariate normal distribution.

It is well known that the density function of an n-dimensional real-valued multivariate normal distribution is given as follows:

$$f(x) = \frac{1}{(2\pi)^{n/2}\sqrt{\det \Omega}} \exp\left(-\frac{1}{2}(x-\mu)^{\mathrm{T}}\Omega^{-1}(x-\mu)\right),$$

where $\mu \in \Re^n$ is the mean and $\Omega \in \mathcal{S}_{++}^n$ is the covariance matrix.

Let us consider a complex-valued normally distributed random variable in $\mathbf{C}$, with the mean value $z_0 \in \mathbf{C}$ and variance $\sigma^2 \in \Re_+$. (For more information on the complex-valued normal distributions, we refer the reader to [2]). Similar to the real-valued case, its density function can be written as

$$f(z) = \frac{1}{\pi\sigma^2} \exp\left(-|z-z_0|^2/\sigma^2\right), \, z \in \mathbf{C}.$$

We denote the complex-valued normal distribution by $\mathcal{N}_c(z_0, \sigma^2)$ with mean z_0 and variance σ^2.

Using Euler's formula, i.e., letting $z - z_0 = \rho e^{\boldsymbol{i}\theta}$, we have

$$f(\rho, \theta) = \frac{\rho}{\pi\sigma^2} \exp\left(-\frac{\rho^2}{\sigma^2}\right), \text{ with } (\rho, \theta) \in [0, +\infty) \times [0, 2\pi),$$

where the variable transformation is

$$\left\{ \begin{array}{lcl} \text{Re}\ (z - z_0) & = & \rho \cos\theta, \\ \text{Im}\ (z - z_0) & = & \rho \sin\theta. \end{array} \right.$$

As a matter of terminology, ρ is usually called the modulus of $z - z_0$, also denoted as $|z - z_0|$; θ is called the argument of $z - z_0$, denoted as Arg $(z - z_0)$.

The density of the joint (complex-valued) normal distribution $z = (z_1, z_2, \ldots, z_n)$, with $z_k \in \mathbf{C}$, $k = 1, 2, \ldots, n$, has the following form:

$$f(z) = \frac{1}{(\pi)^n \det \Omega} \exp\left(-(z - \mu)^{\mathrm{H}} \Omega^{-1} (z - \mu)\right),$$

where $z, \mu \in \mathbf{C}^n$, and $\Omega \in \mathcal{H}^n_{++}$; μ is the mean vector, and Ω is the covariance matrix.

Let us denote the above complex-valued normal distribution as $\mathcal{N}_c(\mu, \Omega)$.

The bivariate case is of particular interest to us. Consider a complex-valued, bivariate normal random vector. Suppose that it has zero-mean. Furthermore, suppose that its covariance matrix is

$$\Omega = \left[\begin{array}{cc} 1 & \lambda \\ \bar{\lambda} & 1 \end{array} \right] \succ 0,$$

where $\bar{\lambda} \in \mathbf{C}$ denotes the conjugate of $\lambda \in \mathbf{C}$. In particular, let $\lambda = \gamma e^{\boldsymbol{i}\alpha}$, and so $\bar{\lambda} = \gamma e^{-\boldsymbol{i}\alpha}$. Since $\Omega \succ 0$, it follows that $1 - \gamma^2 > 0$. Moreover,

$$\Omega^{-1} = \frac{1}{1 - \gamma^2} \left[\begin{array}{cc} 1 & -\gamma e^{\boldsymbol{i}\alpha} \\ -\gamma e^{-\boldsymbol{i}\alpha} & 1 \end{array} \right].$$

Then, by letting $z_1 = \rho_1 e^{\boldsymbol{i}\theta_1}$ and $z_2 = \rho_2 e^{\boldsymbol{i}\theta_2}$, we may rewrite the density function as

$$\begin{aligned} & f(\rho_1, \rho_2, \theta_1, \theta_2) \\ & \quad = \frac{\rho_1 \rho_2}{\pi^2 (1 - \gamma^2)} \exp\left(-\frac{1}{1 - \gamma^2} \left[\begin{array}{c} \rho_1 e^{\boldsymbol{i}\theta_1} \\ \rho_2 e^{\boldsymbol{i}\theta_2} \end{array} \right]^{\mathrm{H}} \left[\begin{array}{cc} 1 & -\gamma e^{\boldsymbol{i}\alpha} \\ -\gamma e^{-\boldsymbol{i}\alpha} & 1 \end{array} \right] \left[\begin{array}{c} \rho_1 e^{\boldsymbol{i}\theta_1} \\ \rho_2 e^{\boldsymbol{i}\theta_2} \end{array} \right] \right) \\ & \quad = \frac{\rho_1 \rho_2}{\pi^2 (1 - \gamma^2)} \exp\left(-\frac{\rho_1^2 + \rho_2^2 - 2\rho_1\rho_2\gamma \cos(\alpha + \theta_2 - \theta_1)}{1 - \gamma^2} \right), \end{aligned}$$

where the domain of the variables is given as

$$(\rho_1, \rho_2, \theta_1, \theta_2) \in [0, +\infty)^2 \times [0, 2\pi)^2.$$

Now let us further introduce a variable transformation

$$\left\{ \begin{array}{lcl} \rho_1 & = & \rho \cos\xi, \\ \rho_2 & = & \rho \sin\xi \end{array} \right.$$

with the domain $(\rho, \xi) \in [0, +\infty) \times [0, \pi/2]$. The density function can be further written as

$$\begin{aligned} f(\rho, \xi, \theta_1, \theta_2) &= \frac{\rho^3 \cos\xi \sin\xi}{\pi^2(1-\gamma^2)} \exp\left(-\frac{\rho^2 - 2\gamma\rho^2 \cos\xi \sin\xi \cos(\alpha + \theta_2 - \theta_1)}{1-\gamma^2} \right) \\ &= \frac{\rho^3 \sin 2\xi}{2\pi^2(1-\gamma^2)} \exp\left(-\frac{\rho^2 - \rho^2\gamma \sin 2\xi \cos(\alpha + \theta_2 - \theta_1)}{1-\gamma^2} \right), \end{aligned}$$

and the domain is $(\rho, \xi, \theta_1, \theta_2) \in [0, +\infty) \times [0, \pi/2] \times [0, 2\pi)^2$.

Consider $0 \le \theta_1^b < \theta_1^e \le 2\pi$ and $0 \le \theta_2^b < \theta_2^e \le 2\pi$. Below we shall compute the probability that $(\theta_1, \theta_2) \in [\theta_1^b, \theta_1^e] \times [\theta_2^b, \theta_2^e]$.

Let us denote

$$\begin{aligned} P :&= \text{Prob}\,\{\theta_1^b \le \theta_1 \le \theta_1^e;\, \theta_2^b \le \theta_2 \le \theta_2^e\} \\ &= \int_{\theta_1^b}^{\theta_1^e} \int_{\theta_2^b}^{\theta_2^e} \int_0^{\pi/2} \left[\int_0^\infty \frac{\rho^3 \sin 2\xi}{2\pi^2(1-\gamma^2)} \exp\left(-\frac{\rho^2 - \rho^2\gamma \sin 2\xi \cos(\alpha + \theta_2 - \theta_1)}{1-\gamma^2} \right) d\rho \right] \\ &\quad \times d\xi d\theta_2 d\theta_1. \end{aligned}$$

To compute the above integration, we note the following facts.

LEMMA 2.1.

(i) *For a given $a > 0$, it holds that*

$$\int_0^\infty \rho^3 \exp(-a\rho^2) d\rho = \frac{1}{2a^2}.$$

(ii) *Suppose that $-1 < b < 1$ is a given real number. Then, with respect to the variable θ, it holds that*

$$\int \frac{\sin\theta}{(1 - b\sin\theta)^2} d\theta = -\frac{\cos\theta}{(1-b^2)(1-b\sin\theta)} + \frac{2b}{(1-b^2)^{3/2}} \arctan \frac{\tan(\theta/2) - b}{\sqrt{1-b^2}} + C.$$

(iii) *With respect to the variable θ, it holds that*

$$\int \left[\frac{1}{1-\gamma^2\cos^2(\theta)} + \frac{\gamma\cos\theta \arccos(-\gamma\cos\theta)}{(1-\gamma^2\cos^2(\theta))^{3/2}} \right] d\theta = \frac{1}{1-\gamma^2} \left(\theta + \frac{\gamma \sin\theta \arccos(-\gamma\cos\theta)}{\sqrt{1-\gamma^2\cos^2(\theta)}} \right) + C.$$

(iv) *With respect to the variable θ, it holds that*

$$\int \left[\frac{\gamma \sin(\beta - \theta) \arccos(-\gamma\cos(\theta - \beta))}{\sqrt{1-\gamma^2\cos^2(\theta-\beta)}} \right] d\theta = \frac{1}{2} \left(\arccos(-\gamma\cos(\theta-\beta)) \right)^2 + C.$$

Part (i) of the lemma is straightforward, and the rest of the lemma can be readily verified by differentiation.

Applying Lemma 2.1 (i) and (ii), we get

$$\begin{aligned}P &= \frac{1}{4\pi^2(1-\gamma^2)}\int_{\theta_1^b}^{\theta_1^e}\int_{\theta_2^b}^{\theta_2^e}\left[\int_0^{\pi/2}\sin 2\xi\left(\frac{1-\gamma^2}{1-\gamma\sin 2\xi\cos(\alpha+\theta_2-\theta_1)}\right)^2 d\xi\right]d\theta_2 d\theta_1\\ &= \frac{1-\gamma^2}{4\pi^2}\int_{\theta_1^b}^{\theta_1^e}\int_{\theta_2^b}^{\theta_2^e}\left[\int_0^{\pi/2}\frac{\sin 2\xi}{(1-\gamma\cos(\alpha+\theta_2-\theta_1)\sin 2\xi)^2}d\xi\right]d\theta_2 d\theta_1\\ &= \frac{1-\gamma^2}{4\pi^2}\int_{\theta_1^b}^{\theta_1^e}\int_{\theta_2^b}^{\theta_2^e}\left[\frac{1}{1-\gamma^2\cos^2(\alpha+\theta_2-\theta_1)}\right.\\ &\quad\left.+\frac{\gamma\cos(\alpha+\theta_2-\theta_1)\arccos\left(-\gamma\cos(\alpha+\theta_2-\theta_1)\right)}{(1-\gamma^2\cos^2(\alpha+\theta_2-\theta_1))^{3/2}}\right]d\theta_2 d\theta_1.\end{aligned}$$

Using Lemma 2.1 (iii) we obtain

$$\begin{aligned}P &= \frac{1}{4\pi^2}\left[(\theta_1^e-\theta_1^b)(\theta_2^e-\theta_2^b)+\int_{\theta_1^b}^{\theta_1^e}\frac{\gamma\sin(\theta_2^e+\alpha-\theta_1)\arccos(-\gamma\cos(\theta_2^e+\alpha-\theta_1))}{\sqrt{1-\gamma^2\cos^2(\theta_2^e+\alpha-\theta_1)}}d\theta_1\right.\\ &\quad\left.-\int_{\theta_1^b}^{\theta_1^e}\frac{\gamma\sin(\theta_2^b+\alpha-\theta_1)\arccos(-\gamma\cos(\theta_2^b+\alpha-\theta_1))}{\sqrt{1-\gamma^2\cos^2(\theta_2^b+\alpha-\theta_1)}}d\theta_1\right],\end{aligned}$$

and then using Lemma 2.1 (iv) we have

$$\begin{aligned}P &= \frac{(\theta_1^e-\theta_1^b)(\theta_2^e-\theta_2^b)}{4\pi^2}+\frac{1}{8\pi^2}\left[\left(\arccos(-\gamma\cos(\theta_1^e-\theta_2^e-\alpha))\right)^2\right.\\ &\quad-\left(\arccos(-\gamma\cos(\theta_1^b-\theta_2^e-\alpha))\right)^2\\ &\quad\left.+\left(\arccos(-\gamma\cos(\theta_1^b-\theta_2^b-\alpha))\right)^2-\left(\arccos(-\gamma\cos(\theta_1^e-\theta_2^b-\alpha))\right)^2\right].\end{aligned}$$

Summarizing, we have proven the following result by a limiting argument.

THEOREM 2.2. *For the complex-valued bivariate normal random vector* $\left[\begin{smallmatrix} z_1 \\ z_2\end{smallmatrix}\right] \in \mathcal{N}_c(\mu,\Omega)$ *with*

$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \textit{and} \quad \Omega = \begin{bmatrix} 1 & \gamma e^{\boldsymbol{i}\alpha} \\ \gamma e^{-\boldsymbol{i}\alpha} & 1 \end{bmatrix} \in \mathcal{H}_+^2,$$

it holds that

$$\begin{aligned}&\text{Prob}\left\{\theta_1^b \le \text{Arg } z_1 \le \theta_1^e;\ \theta_2^b \le \text{Arg } z_2 \le \theta_2^e\right\}\\ &= \frac{(\theta_1^e-\theta_1^b)(\theta_2^e-\theta_2^b)}{4\pi^2}+\frac{1}{8\pi^2}\left[\left(\arccos(-\gamma\cos(\theta_1^e-\theta_2^e-\alpha))\right)^2\right.\\ &\quad-\left(\arccos(-\gamma\cos(\theta_1^b-\theta_2^e-\alpha))\right)^2\\ &\quad\left.+\left(\arccos(-\gamma\cos(\theta_1^b-\theta_2^b-\alpha))\right)^2-\left(\arccos(-\gamma\cos(\theta_1^e-\theta_2^b-\alpha))\right)^2\right].\end{aligned}$$

3. Quadratic programs and complex SDP formulations.

3.1. Discrete complex quadratic optimization. Suppose that Q is a Hermitian matrix. Consider the following quadratic programming problem with discrete decision variables:

$$\begin{array}{lll}(\mathrm{P}) & \max & z^{\mathrm{H}}Qz \\ & \text{s.t.} & z_k \in \{1, \omega, \ldots, \omega^{m-1}\}, \ k = 1, \ldots, n,\end{array}$$

where $m \geq 2$ and $\omega = e^{\boldsymbol{i}\frac{2\pi}{m}} = \cos\frac{2\pi}{m} + \boldsymbol{i}\sin\frac{2\pi}{m}$. As we shall see later, this is an extension of Goemans and Williamson's model for solving the max-3-cut problem; see [9].

Denote the optimal value of (P) to be $v(P)$. Consider the following complex-valued mapping F_m:

$$F_m(z) := \frac{m(2-\omega^{-1}-\omega)}{8\pi^2}\sum_{j=0}^{m-1}\omega^j(\arccos(-\mathrm{Re}\ (\omega^{-j}z)))^2.$$

For a Hermitian matrix Z with $|Z_{kl}| \leq 1$ for all k, l, define the componentwise matrix function

$$F_m(Z) := (F_m(Z_{kl}))_{n\times n} \in \mathcal{H}^n.$$

It is easy to verify that $F_m(\bar{z}) = \overline{F_m(z)}$. Therefore, if Z is Hermitian, then so is $F_m(Z)$.

LEMMA 3.1. *We have*

$$1 = \frac{m(2-\omega^{-1}-\omega)}{8\pi^2}\sum_{j=0}^{m-1}\omega^j(\arccos(-\mathrm{Re}\ (\omega^{-j})))^2.$$

Moreover, $F_m(z) = z$ *for any* $z \in \{1, \omega, \ldots, \omega^{m-1}\}$.

Proof. We observe that

$$\begin{aligned}&\frac{m(2-\omega^{-1}-\omega)}{8\pi^2}\sum_{j=0}^{m-1}\omega^j\left(\arccos\left(-\cos\left(\frac{j}{m}2\pi\right)\right)\right)^2\\ &= \frac{m(2-\omega^{-1}-\omega)}{8\pi^2}\sum_{j=0}^{m-1}\omega^j\pi^2\left(1-\frac{2j}{m}\right)^2\\ (1) \qquad &= \frac{2-\omega^{-1}-\omega}{8m}\left(4\sum_{j=0}^{m-1}j^2\omega^j - 4m\sum_{j=0}^{m-1}j\omega^j\right).\end{aligned}$$

Moreover, we have

$$\sum_{j=0}^{m-1}j^2\omega^j = \frac{m^2(\omega-1)-2m\omega}{(\omega-1)^2} \text{ and } \sum_{j=0}^{m-1}j\omega^j = \frac{m}{\omega-1}.$$

Substituting the above equations into (1) yields the intended result.

Suppose $z = \omega^{j_0}$ for some $j_0 \in \{0, 1, \dots, m-1\}$. Then,

$$\begin{aligned}
&\frac{m(2-\omega^{-1}-\omega)}{8\pi^2}\sum_{j=0}^{m-1}\omega^j(\arccos(-\text{Re }(\omega^{-j}z)))^2 \\
&= \frac{m(2-\omega^{-1}-\omega)}{8\pi^2}\sum_{j=0}^{m-1}\omega^j\left(\arccos\left(-\cos\left(\frac{j_0-j}{m}2\pi\right)\right)\right)^2 \\
&= \frac{m(2-\omega^{-1}-\omega)}{8\pi^2}\sum_{j=0}^{m-1}\omega^j\left(\arccos\left(-\cos\left(\frac{j-j_0}{m}2\pi\right)\right)\right)^2 \\
&= \omega^{j_0}\frac{m(2-\omega^{-1}-\omega)}{8\pi^2}\sum_{j=-j_0}^{m-1-j_0}\omega^j\left(\arccos\left(-\cos\left(\frac{j}{m}2\pi\right)\right)\right)^2 \\
&= \omega^{j_0} = z.
\end{aligned}$$

This completes the proof for Lemma 3.1. □

Hence we can rewrite (P) as

$$\begin{array}{ll}
\max & Q \bullet F_m(zz^{\mathrm{H}}) \\
\text{s.t.} & z_k \in \{1, \omega, \dots, \omega^{m-1}\},\ k = 1, \dots, n.
\end{array}$$

Consider the following nonlinear complex SDP problem:

$$\begin{array}{lll}
\text{(SP)} & \max & Q \bullet F_m(Z) \\
& \text{s.t.} & Z_{kk} = 1,\ k = 1, \dots, n, \\
& & Z \succeq 0.
\end{array}$$

Let $v(SP)$ denote the optimal value of (SP).

THEOREM 3.2. *It holds that* $v(P) = v(SP)$.

Proof. Let $\hat{z}$ be optimal to (P); then, by Lemma 3.1, $\hat{Z} = \hat{z}\hat{z}^{\mathrm{H}}$ is a feasible solution for (SP) and $F_m(\hat{Z}) = \hat{Z}$. Therefore, $v(SP) \geq Q \bullet F_m(\hat{Z}) = Q \bullet \hat{Z} = v(P)$.

On the other hand, for every feasible solution Z of (SP), we randomly generate a complex vector ξ such that $\xi \in \mathcal{N}_c(0, Z)$, and assign

$$z_k = \sigma(\xi_k) = \begin{cases}
1 & \text{if Arg } \xi_k \in [0, \frac{1}{m}2\pi), \\
\omega & \text{if Arg } \xi_k \in [\frac{1}{m}2\pi, \frac{2}{m}2\pi), \\
\vdots & \\
\omega^j & \text{if Arg } \xi_k \in [\frac{j}{m}2\pi, \frac{j+1}{m}2\pi), \\
\vdots & \\
\omega^{m-1} & \text{if Arg } \xi_k \in [\frac{m-1}{m}2\pi, 2\pi)
\end{cases} \tag{2}$$

for $k = 1, \ldots, n$. Suppose that $Z_{kl} = \gamma e^{\boldsymbol{i}\alpha}$. Then, by Theorem 2.2 we have

$$\begin{aligned}
&\text{Prob}\{z_k = z_l\omega^j, z_l = \omega^r\}\\
&= \text{Prob}\{z_k = \omega^{j+r}, z_l = \omega^r\}\\
&= \text{Prob}\left\{\text{Arg }\xi_k \in \left[\frac{j+r}{m}2\pi, \frac{j+r+1}{m}2\pi\right), \text{Arg }\xi_l \in \left[\frac{r}{m}2\pi, \frac{r+1}{m}2\pi\right)\right\}\\
&= \frac{1}{m^2} + \frac{1}{8\pi^2}\left(2\left(\arccos\left(-\gamma\cos\left(\frac{j}{m}2\pi - \alpha\right)\right)\right)^2\right.\\
&\quad - \left(\arccos\left(-\gamma\cos\left(\frac{j-1}{m}2\pi - \alpha\right)\right)\right)^2\\
&\quad \left.- \left(\arccos\left(-\gamma\cos\left(\frac{j+1}{m}2\pi - \alpha\right)\right)\right)^2\right)
\end{aligned}$$

for any $j, r \in \{0, 1, \ldots, m-1\}$. Therefore, for any given k and l we have

$$\begin{aligned}
&\text{Prob}\{z_k\bar{z}_l = \omega^j\}\\
&= \sum_{r=0}^{m-1}\text{Prob}\{z_k = z_l\omega^j, z_l = \omega^r\}\\
&= \frac{1}{m} + \frac{m}{8\pi^2}\left(2\left(\arccos\left(-\gamma\cos\left(\frac{j}{m}2\pi - \alpha\right)\right)\right)^2\right.\\
&\quad - \left(\arccos\left(-\gamma\cos\left(\frac{j-1}{m}2\pi - \alpha\right)\right)\right)^2\\
&\quad \left.- \left(\arccos\left(-\gamma\cos\left(\frac{j+1}{m}2\pi - \alpha\right)\right)\right)^2\right). \qquad (3)
\end{aligned}$$

It follows that

$$\begin{aligned}
&\mathsf{E}[z_k\bar{z}_l]\\
&= \sum_{j=0}^{m-1}\omega^j\text{Prob}\{z_k\bar{z}_l = \omega^j\}\\
&= \frac{m}{8\pi^2}\sum_{j=0}^{m-1}\omega^j\left(2\left(\arccos\left(-\gamma\cos\left(\frac{j}{m}2\pi - \alpha\right)\right)\right)^2\right.\\
&\quad - \left(\arccos\left(-\gamma\cos\left(\frac{j-1}{m}2\pi - \alpha\right)\right)\right)^2\\
&\quad \left.- \left(\arccos\left(-\gamma\cos\left(\frac{j+1}{m}2\pi - \alpha\right)\right)\right)^2\right)\\
&= \frac{m}{8\pi^2}\sum_{j=0}^{m-1}(2\omega^j - \omega^{j-1} - \omega^{j+1})\left(\arccos\left(-\gamma\cos\left(\frac{j}{m}2\pi - \alpha\right)\right)\right)^2\\
&= \frac{m(2 - \omega^{-1} - \omega)}{8\pi^2}\sum_{j=0}^{m-1}\omega^j\left(\arccos\left(-\gamma\cos\left(\frac{j}{m}2\pi - \alpha\right)\right)\right)^2\\
&= \frac{m(2 - \omega^{-1} - \omega)}{8\pi^2}\sum_{j=0}^{m-1}\omega^j(\arccos(-\text{Re }(\omega^{-j}Z_{kl})))^2. \qquad (4)
\end{aligned}$$

By the linearity of mathematical expectation, we get

$$\mathsf{E}[z^{\mathrm{H}}Qz] = Q \bullet F_m(Z).$$

Since the solution z so generated is feasible to (P), we have

$$\begin{aligned} v(P) &\geq \mathsf{E}[z^{\mathrm{H}}Qz] \\ &= Q \bullet Z, \end{aligned}$$

for every feasible solution Z of (SP). This combined with $v(SP) \geq v(P)$ yields the desired result. □

In particular, if $m = 2$, then one can verify that problem (P) reduces to

$$\begin{array}{ll} \max & x^{\mathrm{T}}Qx \\ \text{s.t.} & x_k \in \{\pm 1\},\ k = 1, \ldots, n, \end{array}$$

and problem (SP) reduces to

$$\begin{array}{ll} \max & \frac{2}{\pi} Q \bullet \arcsin(X) \\ \text{s.t.} & X_{kk} = 1,\ k = 1, \ldots, n, \\ & X \succeq 0, \end{array}$$

where $\arcsin(X) := [\arcsin(X_{kl})]_{n \times n}$. In that case, Theorem 3.2 specializes to Theorem 2.9 in Goemans and Williamson [8] or Theorem 1 in Zhang [15]. If $m = 3$, then (P) is

$$\begin{array}{ll} \max & z^{\mathrm{H}}Qz \\ \text{s.t.} & z_k \in \{1, \omega, \omega^2\},\ k = 1, \ldots, n, \end{array}$$

with $\omega = e^{\boldsymbol{i}\frac{2\pi}{3}}$. In fact, Goemans and Williamson ([9]) model the max-3-cut problem as

$$\begin{array}{lll} \text{(M3C)} & \max & \sum_{1 \leq k < l \leq n} w_{kl}(z_k - z_l)^{\mathrm{H}}(z_k - z_l) \\ & \text{s.t.} & z_k = \{1, \omega, \omega^2\},\ k = 1, \ldots, n, \end{array}$$

and they consider the following complex SDP relaxation:

$$\begin{array}{ll} \max & \sum_{1 \leq k < l \leq n} w_{kl}(2 - 2\mathrm{Re}\ Z_{kl}) \\ \text{s.t.} & Z_{kk} = 1,\ k = 1, \ldots, n, \\ & \mathrm{Re}\ Z_{kl} \geq -1/2,\ \mathrm{Re}\ \omega Z_{kl} \geq -1/2,\ \mathrm{Re}\ \omega^2 Z_{kl} \geq -1/2,\ 1 \leq k < l \leq n, \\ & Z \succeq 0. \end{array}$$

Let the optimal solution of the SDP relaxation be Z^*. Then, Theorem 3.2 asserts that the expected value of the randomized solution based on Z^* is

$$\sum_{1 \leq k < l \leq n} w_{kl}(2 - 2\mathrm{Re}\ F_3(Z^*_{kl})),$$

where $F_3(z) = \frac{9}{8\pi^2}\left[(\arccos(-\mathrm{Re}\ z))^2 + \omega(\arccos(-\mathrm{Re}\ (\omega^2 z)))^2 + \omega^2(\arccos(-\mathrm{Re}\ (\omega z)))^2\right]$.

Since $(\arccos(x))^2$ is a convex function, it follows that

$$\begin{aligned}
&\text{Re } F_3(Z^*_{kl})\\
&= \frac{9}{8\pi^2}\left[(\arccos(-\text{Re } Z^*_{kl}))^2 - \frac{1}{2}\left((\arccos(-\text{Re } (\omega^2 Z^*_{kl})))^2 + (\arccos(-\text{Re } (\omega Z^*_{kl})))^2\right)\right]\\
&\le \frac{9}{8\pi^2}\left[(\arccos(-\text{Re } Z^*_{kl}))^2 - \left(\arccos\left(-\frac{1}{2}\text{Re } \left(\omega Z^*_{kl} + \omega^2 Z^*_{kl}\right)\right)\right)^2\right]\\
&= \frac{9}{8\pi^2}\left[(\arccos(-\text{Re } Z^*_{kl}))^2 - \left(\arccos\left(\frac{1}{2}\text{Re } Z^*_{kl}\right)\right)^2\right].
\end{aligned}$$

Further noticing that

$$\min_{-\frac{1}{2}\le x<1} \frac{2 + \frac{9}{4\pi^2}\left[\left(\arccos(\frac{x}{2})\right)^2 - (\arccos(-x))^2\right]}{2-2x} = 0.8360\ldots,$$

the approximation ratio of Goemans and Williamson [9] thus follows from the fact that

$$\begin{aligned}
&\sum_{1\le k<l\le n} w_{kl}(2 - 2\text{Re } F_3(Z^*_{kl}))\\
&\ge \sum_{1\le k<l\le n} w_{kl}\left\{2 - 2\times\frac{9}{8\pi^2}\left[(\arccos(-\text{Re } Z^*_{kl}))^2 - \left(\arccos\left(\frac{1}{2}\text{Re } Z^*_{kl}\right)\right)^2\right]\right\}\\
&\ge 0.836 \times \sum_{1\le k<l\le n} w_{kl}\left(2 - 2\text{Re } Z^*_{kl}\right)\\
&\ge 0.836 \times v^*(M3C).
\end{aligned}$$

The above analysis is due to Goemans and Williamson [9]. Therefore, in this sense (3) is a generalization of Theorem 1 of [9] and our rounding procedure (2) is an extension of the procedure in section 5.1 of [9].

3.2. Bounds on the approximation ratios. In this subsection, we investigate approximation algorithms for (P) with positive semidefinite Q via complex SDP relaxation.

Consider the following complex SDP relaxation for (P):

$$\begin{array}{lll}
\text{(CSDP)} & \max & Q \bullet Z\\
& \text{s.t.} & Z_{kk} = 1,\ k = 1,\ldots,n,\\
& & Z \succeq 0.
\end{array}$$

Suppose that Z^* is an optimal solution of (CSDP). We draw a random vector $\xi \in \mathcal{N}_c(0, Z^*)$, and generate a feasible solution $z \in \mathbf{C}^n$ of (P) by applying the rounding procedure (2).

In what follows, we wish to establish an approximation ratio $\alpha \in (0,1]$ for the approximation algorithm, i.e., an α such that

$$\mathsf{E}[Q \bullet zz^{\text{H}}] \ge \alpha(Q \bullet Z^*),$$

for the randomized solution z.

To begin with, we need the following technical lemma, whose proof is given in the appendix of this paper.

Lemma 3.3. *Suppose that $Z \in \mathcal{H}^n$ is positive semidefinite. Then*

$$F_2(Z) \succeq \frac{1}{\pi}(Z + Z^T) = \frac{2}{\pi} \operatorname{Re} Z \ \text{ and } \ F_m(Z) \succeq \frac{m^2(1-\cos\frac{2\pi}{m})}{8\pi} Z \text{ for } m \geq 3.$$

Therefore, according to (4) and Lemma 3.3, the expectation of the objective value of z can be estimated as

$$\begin{aligned} \mathsf{E}[Q \bullet zz^{\mathrm{H}}] &= Q \bullet F_m(Z^*) \\ &\geq \alpha_m (Q \bullet Z^*), \end{aligned}$$

where

$$\alpha_m = \begin{cases} \frac{2}{\pi} & \text{if } m = 2, \\ \frac{m^2(1-\cos\frac{2\pi}{m})}{8\pi} & \text{if } m \geq 3. \end{cases}$$

Hence we arrive at the approximation ratio α_m for our randomized algorithm for solving (P) ($m \geq 2$). Summarizing, we have the following theorem.

Theorem 3.4. *Suppose that $Q \succeq 0$. Then there holds $\mathsf{E}[Q \bullet zz^H] \geq \alpha_m v(\mathrm{P})$, where z is obtained by the randomized algorithm and $v(\mathrm{P})$ is the optimal value of* (P)*. In particular, $\alpha_3 \geq 0.5371$, $\alpha_4 \geq 0.6366$, $\alpha_5 \geq 0.6873$, $\alpha_{10} \geq 0.7599$, and $\alpha_{100} \geq 0.7851$.*

In the case of $m = 2$, (CSDP) is actually a real SDP problem. According to Lemma 3.3, one asserts that the real version of relaxation problem (CSDP) yields a $\frac{2}{\pi}$-approximation ratio, which is in accordance with the result of Nesterov [11].

Prior to our result in this section, we learned through private communications that So, Zhang, and Ye [12] used a very different technique based on Grothendieck's inequality and obtained the same approximation ratio result for the discrete complex quadratic optimization problem (P). We believe that both techniques are interesting and useful in their own right.

3.3. Continuous complex quadratic optimization. By taking the limit, i.e., $m \to \infty$, the quadratic optimization model (P) becomes

$$\begin{array}{lll} \text{(CP)} & \max & z^{\mathrm{H}} Q z \\ & \text{s.t.} & |z_k| = 1,\ k = 1, \ldots, n, \end{array}$$

where $Q \in \mathcal{H}^n_+$. In that case, the problem is equivalent to

$$\begin{array}{lll} \text{(SCP)} & \max & Q \bullet F(Z) \\ & \text{s.t.} & Z_{kk} = 1,\ k = 1, \ldots, n, \\ & & Z \succeq 0 \end{array}$$

with

$$\begin{aligned} F(z) &:= \lim_{m\to\infty} F_m(z) \\ &= \frac{1}{4\pi} \int_0^{2\pi} e^{\boldsymbol{i}\theta} \left(\arccos(-\gamma\cos(\theta - \alpha))\right)^2 d\theta, \end{aligned}$$

where $\gamma = |z| \leq 1$ and $\alpha = \operatorname{Arg} z$.

The applications of Hermitian quadratic optimization models such as (P) and (CP) can be found, e.g., in Luo, Luo, and Kisialiou [10] for applications in signal processing. Although in [10] the minimization version of the problem was considered, from the viewpoint of optimization both formulations are equivalent (see reduction below).

PROPOSITION 3.5. *Problem* (CP) *is strongly NP-hard in general.*

Proof. The optimization problem in the form

$$\begin{array}{ll} \max & |z^{\mathrm{T}} A z| \\ \text{s.t.} & z_k \in \mathbf{C},\ |z_k| \le 1,\ k = 1, \dots, n, \end{array}$$

is called *complex programming* and was shown in [13] to be NP-hard in general. Problem (CP) is related to complex programming, but they are not the same: the objective in (CP) takes the Hermitian form and is assumed to be positive semidefinite. The proof for Proposition 3.5 to be presented below is due to Tom Luo of Minnesota University, who sketched this proof to us in a private communication.

As a first step we shall prove that the problem

$$\begin{array}{ll} \min & z^{\mathrm{H}} Q z \\ \text{s.t.} & |z_k| = 1,\ k = 1, \dots, n, \end{array}$$

is NP-hard in general, where $Q \in \mathcal{H}^n_+$.

To this end, we consider a reduction from the following NP-complete matrix partition problem; i.e., given a matrix $G = [G_1, \dots, G_N] \in \Re^{M \times N}$, decide whether or not a subset of $\{1, \dots, N\}$ exists, say I, such that

$$\sum_{k \in I} G_k = \frac{1}{2} \sum_{k=1}^{N} G_k.$$

The NP-completeness of the above problem follows from the fact that when $M = 1$ and all the components are positive integers the above problem reduces to the famous partition problem, which is NP-complete (see, e.g., page 223 of [7]).

Let the decision vector be

$$z = (z_0, z_1, \dots, z_N, z_{N+1}, \dots, z_{2N})^{\mathrm{T}} \in \mathbf{C}^{2N+1}.$$

Let $n = 2N + 1$ and

$$A := \begin{pmatrix} -e_N & I_N & I_N \\ -\frac{1}{2} G e_N & G & 0_N^{\mathrm{T}} \end{pmatrix} \in \Re^{(M+N) \times n},$$

where $e_N \in \Re^N$ is the vector of all ones. Let $Q := A^{\mathrm{T}} A$.

Next we show that a matrix partition exists if and only if there is $z \in \mathbf{C}^n$, with $|z_k| = 1$ for all k, such that $z^{\mathrm{H}} Q z = 0$. Clearly, $z^{\mathrm{H}} Q z = 0$ is equivalent to $Az = 0$; that is,

$$0 = -z_0 + z_k + z_{N+k},\ k = 1, \dots, N, \tag{5}$$

$$0 = -\frac{1}{2} \left(\sum_{k=1}^{N} G_k \right) z_0 + \sum_{k=1}^{N} G_k z_k. \tag{6}$$

Let $z_k / z_0 = e^{\boldsymbol{i}\theta_k}$ for $k = 1, \dots, 2N$. Using (5) we have

$$\cos\theta_k + \cos\theta_{N+k} = 1, \tag{7}$$

$$\sin\theta_k + \sin\theta_{N+k} = 0, \tag{8}$$

where $k = 1, \ldots, N$. Equations (7) and (8) imply that $\theta_k \in \{-\pi/3, \pi/3\}$. This in particular means that $\cos\theta_k = \cos\theta_{N+k} = 1/2$ for $k = 1, \ldots, N$. Since

$$\text{Re}\ \left(-\frac{1}{2}\left(\sum_{k=1}^{N} G_k\right) + \sum_{k=1}^{N} G_k z_k/z_0\right) = -\frac{1}{2}\sum_{k=1}^{N} G_k + \sum_{k=1}^{N} G_k \cos\theta_k = 0$$

is always satisfied, (6) is true if and only if

$$\text{Im}\ \left(-\frac{1}{2}\left(\sum_{k=1}^{N} G_k\right) + \sum_{k=1}^{N} G_k z_k/z_0\right) = \sum_{k=1}^{N} G_k \sin\theta_k = 0,$$

which amounts to the existence of a matrix partition.

Let $\lambda_{\max}$ be the maximum eigenvalue of Q. By observing that

$$\begin{array}{ll} \min & z^{\mathrm{H}} Q z \\ \text{s.t.} & |z_k| = 1,\ k = 1, \ldots, n, \end{array}$$

is equivalent to

$$\begin{array}{ll} \max & z^{\mathrm{H}} (\lambda_{\max} I - Q) z \\ \text{s.t.} & |z_k| = 1,\ k = 1, \ldots, n, \end{array}$$

where $\lambda_{\max} I - Q \in \mathcal{H}_+^n$, the desired result follows. □

For a given $z \in \mathbf{C}$ with $z = \gamma e^{\boldsymbol{i}\alpha}$ and $|z| = \gamma \le 1$, we have

$$\begin{aligned} F(z) &= \frac{1}{4\pi}\int_0^{2\pi} e^{\boldsymbol{i}\theta} \left(\arccos(-\gamma\cos(\theta-\alpha))\right)^2 d\theta \\ &= \frac{1}{4\pi} e^{\boldsymbol{i}\alpha} \int_0^{2\pi} e^{\boldsymbol{i}\theta} \left(\arccos(-\gamma\cos\theta)\right)^2 d\theta \\ &= \frac{1}{4\pi} e^{\boldsymbol{i}\alpha} \left[\int_0^{\pi} e^{\boldsymbol{i}\theta} \left(\arccos(-\gamma\cos\theta)\right)^2 d\theta - \int_0^{\pi} e^{\boldsymbol{i}\theta} \left(\arccos(\gamma\cos\theta)\right)^2 d\theta\right] \\ &= \frac{1}{2} e^{\boldsymbol{i}\alpha} \int_0^{\pi} e^{\boldsymbol{i}\theta} \left(\frac{\pi}{2} - \arccos(\gamma\cos\theta)\right) d\theta \\ &= \frac{1}{2} e^{\boldsymbol{i}\alpha} \int_0^{\pi} e^{\boldsymbol{i}\theta} \arcsin(\gamma\cos\theta) d\theta \\ &= \frac{1}{2} e^{\boldsymbol{i}\alpha} \int_0^{\pi} e^{\boldsymbol{i}\theta} \left(\gamma\cos\theta + \sum_{k=1}^{\infty} \frac{(2k)!}{4^k (k!)^2 (2k+1)} (\gamma\cos\theta)^{2k+1}\right) d\theta \\ &= \frac{\pi}{4}\gamma e^{\boldsymbol{i}\alpha} + \frac{\pi}{2}\sum_{k=1}^{\infty} \frac{((2k)!)^2}{2^{4k+1}(k!)^4(k+1)} \gamma^{2k+1} e^{\boldsymbol{i}\alpha} \\ &= \frac{\pi}{4} z + \frac{\pi}{2}\sum_{k=1}^{\infty} \frac{((2k)!)^2}{2^{4k+1}(k!)^4(k+1)} |z|^{2k} z, \end{aligned} \tag{9}$$

where the second to last step follows from the fact that

$$\int_0^{\pi} \sin\theta(\cos\theta)^{2k+1} d\theta = 0 \text{ and } \int_0^{\pi} (\cos\theta)^{2k+2} d\theta = \frac{(2k+1)(2k-1)\cdots 1}{(2k+2)(2k)\cdots 2}\pi,\ k = 0, 1, \ldots.$$

Clearly, if $Z \in \mathcal{H}_+^n$, then $Z^{\mathrm{T}} \in \mathcal{H}_+^n$. Furthermore, observe that the Hadamard product of any two positive semidefinite Hermitian matrices remains Hermitian positive semidefinite. Denote $A \circ B$ to be the Hadamard product of A and B, and denote $A^{(k)}$ to be

$$\overbrace{A \circ A \circ \cdots \circ A}^{k}.$$

It thus follows from (9) that

$$F(Z) = \frac{\pi}{4} Z + \frac{\pi}{2} \sum_{k=1}^{\infty} \frac{((2k)!)^2}{2^{4k+1}(k!)^4(k+1)} (Z^{\mathrm{T}} \circ Z)^{(k)} \circ Z \succeq \frac{\pi}{4} Z.$$

Since $Q \succeq 0$, we have

$$Q \bullet F(Z) \geq \frac{\pi}{4} Q \bullet Z.$$

Consider the following complex SDP relaxation for (CP):

$$\begin{array}{lll} \text{(CSDP)} & \max & Q \bullet Z \\ & \text{s.t.} & Z_{kk} = 1,\ k = 1, \ldots, n, \\ & & Z \succeq 0. \end{array}$$

Let the optimal value of (CP) be $v^*(CP)$, the optimal value of (CSDP) be $v^*(CSDP)$, and Z^* be an optimal solution. Suppose that a randomized solution z is generated by independently setting $z_k = e^{\boldsymbol{i} \operatorname{Arg} \xi_k}$ for each $k = 1, \ldots, n$, and $\xi \in \mathcal{N}_c(0, Z^*)$. Let the expected value of the randomized solution z be $v(H(C))$. Then

$$v(H(C)) \geq \frac{\pi}{4} v^*(CSDP) \geq \frac{\pi}{4} v^*(CP) \approx 0.7854 \cdot v^*(CP).$$

Since (CP) can be viewed as the limit of (P) as $m \to \infty$, it is interesting to observe that the approximation ratio for (CP), $\frac{\pi}{4}$, is indeed the limit of $\alpha_m = \frac{m^2(1-\cos\frac{2\pi}{m})}{8\pi}$ as $m \to \infty$. It is also interesting to compare this ratio with that of its real counterpart:

$$\begin{array}{lll} \text{(RP)} & \max & x^{\mathrm{T}} Q x \\ & \text{s.t.} & x_k^2 = 1,\ k = 1, \ldots, n, \end{array}$$

where Q is a real positive semidefinite matrix. Nesterov [11] showed that in this case the randomization solution based on the SDP relaxation

$$\begin{array}{lll} \text{(RSDP)} & \max & Q \bullet X \\ & \text{s.t.} & X_{kk} = 1,\ k = 1, \ldots, n, \\ & & X \succeq 0, \end{array}$$

has the following approximation ratio:

$$v(H(R)) \geq \frac{2}{\pi} v^*(RSDP) \geq \frac{2}{\pi} v^*(RP) \approx 0.6366 \cdot v^*(RP).$$

Therefore, the complex SDP relaxation for the complex quadratic optimization problem is more effective than the real SDP relaxation for its real counterpart, in the sense that the former has a slightly better approximation ratio.

We remark that as in the analysis of Nesterov [11], Ye [14], and Zhang [15] for the real case, we can extend all the approximation results to the following more general setting:

$$\begin{array}{ll} \max & z^{\mathrm{H}} Q z \\ \text{s.t.} & (|z_1|^2, |z_2|^2, \dots, |z_n|^2)^{\mathrm{T}} \in \mathcal{F}, \end{array}$$

where $\mathcal{F}$ is a closed convex set in $\Re^n$. The corresponding complex and convex SDP relaxation is

$$\begin{array}{ll} \max & Q \bullet Z \\ \text{s.t.} & \operatorname{diag} Z \in \mathcal{F}, \\ & Z \succeq 0. \end{array}$$

In particular, if $\mathcal{F}$ is a hypercube and $Q \succ 0$, then the above $\frac{\pi}{4}$-approximation result also follows from the matrix cube theorem of Ben-Tal, Nemirovski, and Roos [3]. However, our technique appears to be very different in nature.

It is also interesting to remark that if we regard (CP) as an equivalent real quadratic problem

$$\begin{array}{ll} \max & (u^{\mathrm{T}}, v^{\mathrm{T}}) \begin{pmatrix} \operatorname{Re} Q & -\operatorname{Im} Q \\ \operatorname{Im} Q & \operatorname{Re} Q \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} \\ \text{s.t.} & u_k^2 + v_k^2 = 1,\ k = 1, \dots, n, \end{array}$$

then the approximation ratio obtained that way would be $2/\pi$ instead of $\pi/4$. This shows that the complex SDP relaxation does have an advantage in this particular case.

3.4. Structured continuous complex quadratic optimization. In this section, we study a special case of (CP) with a sign structure on the object matrix, which is parallel to the original (real) max-cut model studied in [8]:

$$\begin{array}{lll} \text{(CPS)} & \max & z^{\mathrm{H}} Q z \\ & \text{s.t.} & |z_k| = 1,\ k = 1, \dots, n, \end{array}$$

where we assume that $Q = [q_{jl}]_{n \times n} \in \mathcal{S}_+^n$ and $q_{jl} \le 0$ for all $1 \le j < l \le n$. Using (9) we know that the expected value of the randomized solution based on the complex SDP relaxation is

$$\begin{aligned} v(H(C)) &= 2 \sum_{j<l} q_{jl} \operatorname{Re}\, F(Z^*_{jl}) + \sum_{j=1}^{n} q_{jj} \\ &= 2 \sum_{j<l} q_{jl} \left(\frac{\pi}{4} + \frac{\pi}{2} \sum_{k=1}^{\infty} \frac{((2k)!)^2}{2^{4k+1} (k!)^4 (k+1)} |Z^*_{jl}|^{2k} \right) \operatorname{Re}\, Z^*_{jl} \\ &\quad + \sum_{j=1}^{n} q_{jj}, \end{aligned} \tag{10}$$

where Z^* is the optimal solution of the complex SDP relaxation. Define the following real function on $y \in [0,1]$:

$$g(y) := \frac{\pi}{4} + \frac{\pi}{2} \sum_{k=1}^{\infty} \frac{((2k)!)^2}{2^{4k+1} (k!)^4 (k+1)} y^{2k}$$

We have $0 \leq g(y) \leq 1$ for all $y \in [0,1]$. Suppose that x is real and $|x| \leq y \leq 1$. Then,

$$\min_{|x|\leq y} \frac{1-g(y)x}{1-x} = \min_{|x|\leq y} \left(g(y) + \frac{1-g(y)}{1-x} \right) = \frac{1+g(y)y}{1+y}.$$

One computes that

$$\min_{0\leq y\leq 1} \frac{1+g(y)y}{1+y} \approx 0.9349 =: \beta.$$

Therefore,

$$1 - g(y)x \geq \beta - \beta x$$

for all $y \in [0,1]$ and $|x| \leq y$, or equivalently,

$$g(y)x \leq 1 - \beta + \beta x \tag{11}$$

for all $y \in [0,1]$ and $|x| \leq y$. Using (11), we have

$$\left(\frac{\pi}{4} + \frac{\pi}{2} \sum_{k=1}^{\infty} \frac{((2k)!)^2}{2^{4k+1}(k!)^4(k+1)} |Z_{jl}^*|^{2k} \right) \text{Re } Z_{jl}^* \leq 1 - \beta + \beta \text{Re } Z_{jl}^*. \tag{12}$$

Now we apply (12) in a componentwise fashion to (10) and obtain, thanks to the sign restriction, the following inequalities:

$$\begin{aligned} v(H(C)) &= 2\sum_{j<l} q_{jl} \left(\frac{\pi}{4} + \frac{\pi}{2} \sum_{k=1}^{\infty} \frac{((2k)!)^2}{2^{4k+1}(k!)^4(k+1)} |Z_{jl}^*|^{2k} \right) \text{Re } Z_{jl}^* + \sum_{j=1}^{n} q_{jj} \\ &\geq 2\sum_{j<l} q_{jl}(1 - \beta + \beta \text{Re } Z_{jl}^*) + \sum_{j=1}^{n} q_{jj} \\ &= (1-\beta)e^{\mathrm{T}}Qe + \beta Q \bullet Z^* \\ &\geq \beta v^*(CSDP) \\ &\geq \beta v^*(CPS). \end{aligned} \tag{13}$$

This yields an approximation ratio of 0.9349 for (CPS).

4. Concluding remarks. In this paper we discussed complex quadratic maximization models, denoted as (P) and (CP), in which the decision variables either take values as unit roots of the equation $z^m = 1$ or are assumed to have modulus 1. We established approximation ratios for randomization algorithms for these problems, based on the properties of the complex-valued normal distributions. In particular, the approximation ratio is $\frac{m^2(1-\cos\frac{2\pi}{m})}{8\pi}$ for (P) when $m \geq 3$, and is $\pi/4$ for (CP). If the off-diagonal elements of the objective matrix Q are nonpositive, then the approximation ratio is improved to 0.9349. Our approach is based on a probability analysis of the complex-valued normally distributed random variables. The same results can also be obtained by different approaches. For example, recently So, Zhang, and Ye [12] used Grothendieck's inequality and obtained the same $\frac{m^2(1-\cos\frac{2\pi}{m})}{8\pi}$ approximation bound for (P), and Ben-Tal, Nemirovski, and Roos [3] established a matrix cube theorem and obtained the $\pi/4$ approximation ratio for a model similar to (CP). Moreover, Ben-Tal, Nemirovski, and Roos [3] also suggested that the $\pi/4$

approximation ratio is a tight bound. However, it remains unknown whether or not $\alpha_m = \frac{m^2(1-\cos\frac{2\pi}{m})}{8\pi}$ is a tight bound for (P). Related to our models, Charikar and Wirth [5] discussed quadratic maximization models (the real case) in which Q is not assumed to be positive semidefinite; instead, the diagonals of Q are assumed to be all zeros. They proposed a randomized algorithm for such a quadratic maximization model and established an $\Omega(1/\log n)$-approximation ratio. We plan to extend our analysis to such models in the future.

Appendix A. Proof of Lemma 3.3.

Consider

$$\begin{aligned} F_m(z) &= \frac{m(2-\omega-\omega^{-1})}{8\pi^2}\sum_{j=0}^{m-1}\omega^j(\arccos(-\mathrm{Re}\ (\omega^{-j}z)))^2 \\ &= \frac{m(1-\cos\frac{2\pi}{m})}{4\pi^2}e^{\boldsymbol{i}\alpha}\sum_{j=0}^{m-1}e^{\boldsymbol{i}\theta_j}(\arccos(-\gamma\cos\theta_j))^2, \end{aligned}$$

where $z=\gamma e^{\boldsymbol{i}\alpha}$, $\omega=e^{\boldsymbol{i}\frac{2\pi}{m}}$, and $\theta_j=\frac{j}{m}2\pi-\alpha$ for $j=0,\dots,m-1$.

Since $\arccos(-x)=\frac{\pi}{2}-\arcsin(-x)$, we have

$$\begin{aligned} F_m(z) &= \frac{m(1-\cos\frac{2\pi}{m})}{4\pi^2}e^{\boldsymbol{i}\alpha}\sum_{j=0}^{m-1}e^{\boldsymbol{i}\theta_j}\left(\frac{\pi^2}{4}+\pi\arcsin(\gamma\cos\theta_j)+(\arcsin(\gamma\cos\theta_j))^2\right) \\ &= \frac{m(1-\cos\frac{2\pi}{m})}{4\pi^2}e^{\boldsymbol{i}\alpha}\sum_{j=0}^{m-1}(\pi e^{\boldsymbol{i}\theta_j}\arcsin(\gamma\cos\theta_j)+e^{\boldsymbol{i}\theta_j}(\arcsin(\gamma\cos\theta_j))^2) \\ &= \frac{m(1-\cos\frac{2\pi}{m})}{4\pi^2}e^{\boldsymbol{i}\alpha}\sum_{j=0}^{m-1}(\pi e^{\boldsymbol{i}\theta_j}\gamma\cos\theta_j+\pi e^{\boldsymbol{i}\theta_j}(\arcsin(\gamma\cos\theta_j)-\gamma\cos\theta_j) \\ &\quad + e^{\boldsymbol{i}\theta_j}(\arcsin(\gamma\cos\theta_j))^2). \end{aligned}$$

Set

$$\begin{aligned} I_1 &= \gamma e^{\boldsymbol{i}\alpha}\sum_{j=0}^{m-1}e^{\boldsymbol{i}\theta_j}\cos\theta_j, \\ I_2 &= e^{\boldsymbol{i}\alpha}\sum_{j=0}^{m-1}e^{\boldsymbol{i}\theta_j}(\arcsin(\gamma\cos\theta_j)-\gamma\cos\theta_j), \\ I_3 &= e^{\boldsymbol{i}\alpha}\sum_{j=0}^{m-1}e^{\boldsymbol{i}\theta_j}(\arcsin(\gamma\cos\theta_j))^2. \end{aligned}$$

Thus, we shall have

$$F_m(z)=\frac{m(1-\cos\frac{2\pi}{m})}{4\pi}\left(I_1+I_2+I_3/\pi\right). \tag{14}$$

Let us now treat these items one by one. First, we note that

$$\begin{aligned}
I_1 &= \gamma e^{\boldsymbol{i}\alpha} \sum_{j=0}^{m-1} e^{\boldsymbol{i}\theta_j} \cos\theta_j \\
&= \frac{\gamma e^{\boldsymbol{i}\alpha}}{2} \sum_{j=0}^{m-1} e^{\boldsymbol{i}\theta_j} (e^{\boldsymbol{i}\theta_j} + e^{-\boldsymbol{i}\theta_j}) \\
&= \frac{\gamma e^{\boldsymbol{i}\alpha}}{2} \left(m + \sum_{j=0}^{m-1} e^{\boldsymbol{i}\frac{4\pi}{m}j} e^{-2\boldsymbol{i}\alpha} \right) \\
&= \begin{cases} \frac{\gamma e^{\boldsymbol{i}\alpha}}{2} m = mz/2 & \text{if } m \geq 3, \\ \gamma e^{\boldsymbol{i}\alpha} + \gamma e^{-\boldsymbol{i}\alpha} = z + \bar{z} & \text{if } m = 2. \end{cases}
\end{aligned}$$

Let us denote $a_n = \frac{(2n)!}{2^{2n}(n!)^2(2n+1)}$, $n = 0, 1, \ldots$. Then we have the Taylor expansion $\arcsin(t) = \sum_{n=0}^{\infty} a_n t^{2n+1}$, and so

$$\begin{aligned}
I_2 &= e^{\boldsymbol{i}\alpha} \sum_{j=0}^{m-1} e^{\boldsymbol{i}\theta_j} \sum_{n=1}^{\infty} a_n \gamma^{2n+1} (\cos\theta_j)^{2n+1} \\
&= \sum_{n=1}^{\infty} \frac{a_n}{2^{2n+1}} \gamma^{2n+1} e^{\boldsymbol{i}\alpha} \sum_{j=0}^{m-1} e^{\boldsymbol{i}\theta_j} \left(e^{-\boldsymbol{i}\theta_j} + e^{\boldsymbol{i}\theta_j} \right)^{2n+1} \\
&= \sum_{n=1}^{\infty} \frac{a_n}{2^{2n+1}} \gamma^{2n+1} e^{\boldsymbol{i}\alpha} \sum_{j=0}^{m-1} \sum_{k=0}^{2n+1} \binom{2n+1}{k} e^{\boldsymbol{i}\theta_j (2n+2-2k)} \\
&= \sum_{n=1}^{\infty} \frac{a_n}{2^{2n+1}} \gamma^{2n+1} e^{\boldsymbol{i}\alpha} \sum_{k=0}^{2n+1} \binom{2n+1}{k} \left[\sum_{j=0}^{m-1} e^{\boldsymbol{i}\frac{2\pi}{m}(2n+2-2k)j} \right] e^{-\boldsymbol{i}\alpha(2n+2-2k)}.
\end{aligned}$$

Let us denote

$$b_k = \sum_{j=0}^{m-1} e^{\boldsymbol{i}\frac{2\pi}{m}kj},$$

where k is an integer number. Obviously, b_k is either 0 or m. In particular, if m is even and k is odd, then $b_k = 0$. We further obtain that

$$\begin{aligned}
I_2 &= \sum_{n=1}^{\infty} \frac{a_n}{2^{2n+1}} \gamma^{2n+1} e^{\boldsymbol{i}\alpha} \sum_{k=0}^{2n+1} \binom{2n+1}{k} b_{2n+2-2k} e^{-\boldsymbol{i}\alpha(2n+2-2k)} \\
&= \sum_{n=1}^{\infty} \frac{a_n}{2^{2n+1}} \sum_{k=0}^{2n+1} \binom{2n+1}{k} b_{2n+2-2k} \bar{z}^{2n+1-k} z^k.
\end{aligned}$$

Similarly, we have

$$\begin{aligned}
I_3 &= e^{\boldsymbol{i}\alpha}\sum_{j=0}^{m-1} e^{\boldsymbol{i}\theta_j}(\arcsin(\gamma\cos\theta_j))^2 \\
&= e^{\boldsymbol{i}\alpha}\sum_{j=0}^{m-1} e^{\boldsymbol{i}\theta_j}\sum_{s=0,t=0}^{\infty} a_s a_t \gamma^{2s+2t+2}(\cos\theta_j)^{2s+2t+2} \\
&= e^{\boldsymbol{i}\alpha}\sum_{s=0,t=0}^{\infty}\frac{a_s a_t}{2^{2s+2t+2}}\gamma^{2s+2t+2}\sum_{k=0}^{2s+2t+2}\begin{pmatrix} 2s+2t+2 \\ k \end{pmatrix}\sum_{j=0}^{m-1} e^{\boldsymbol{i}\theta_j(2s+2t+3-2k)} \\
&= e^{\boldsymbol{i}\alpha}\sum_{s=0,t=0}^{\infty}\frac{a_s a_t}{2^{2s+2t+2}}\gamma^{2s+2t+2}\sum_{k=0}^{2s+2t+2}\begin{pmatrix} 2s+2t+2 \\ k \end{pmatrix} b_{2s+2t+3-2k}e^{-\boldsymbol{i}\alpha(2s+2t+3-2k)} \\
&= \sum_{s=0,t=0}^{\infty}\frac{a_s a_t}{2^{2s+2t+2}}\sum_{k=0}^{2s+2t+2}\begin{pmatrix} 2s+2t+2 \\ k \end{pmatrix} b_{2s+2t+3-2k}\bar{z}^{2s+2t+2-k}z^k.
\end{aligned}$$

If m is even, then $I_3 = 0$ since $b_{2s+2t+3-2k} = 0$ in each term.

Observing that the Hadamard product of any two positive semidefinite Hermitian matrices remains Hermitian positive semidefinite, it follows from (14) that

$$\begin{aligned}
&F_m(Z) \\
&= \frac{m^2(1-\cos\frac{2\pi}{m})}{8\pi}Z + \frac{m(1-\cos\frac{2\pi}{m})}{4\pi}\sum_{n=1}^{\infty}\left[\frac{a_n}{2^{2n+1}}\sum_{k=0}^{2n+1}\begin{pmatrix} 2n+1 \\ k \end{pmatrix} b_{2n+2-2k}\right. \\
&\quad \left. Z^{(k)}\circ(Z^{\mathrm{T}})^{(2n+1-k)}\right] + \frac{m(1-\cos\frac{2\pi}{m})}{4\pi^2}\sum_{s=0,t=0}^{\infty}\left[\frac{a_s a_t}{2^{2s+2t+2}}\sum_{k=0}^{2s+2t+2}\begin{pmatrix} 2s+2t+2 \\ k \end{pmatrix}\right. \\
&\quad \left. b_{2s+2t+3-2k}Z^{(k)}\circ(Z^{\mathrm{T}})^{(2s+2t+2-k)}\right],
\end{aligned}$$

for $m \ge 3$; a similar expansion of $F_2(Z)$ can be obtained easily. Since $Z \succeq 0$ and $Z^{\mathrm{T}} \succeq 0$, we get

$$F_m(Z) \succeq \frac{m^2(1-\cos\frac{2\pi}{m})}{8\pi}Z,$$

for $m \ge 3$ and $F_2(Z) \succeq \frac{1}{\pi}(Z + Z^{\mathrm{T}}) = \frac{2}{\pi}\,\mathrm{Re}\,Z$. This completes the proof.

Acknowledgments. We would like to thank Tom Luo, Anthony So, Yinyu Ye, and Jiawei Zhang for stimulating discussions on the subject. We thank André Tits for drawing our attention to complex programming.

REFERENCES

[1] I. G. Abrahamson, *Orthant probability for the quadrivariate normal distribution*, Ann. Math. Statist., 35 (1964), pp. 1685–1703.

[2] H. H. Andersen, M. Højbjerre, D. Sørensen, and P. S. Eriksen, *Linear and Graphical Models for the Multivariate Complex Normal Distribution*, Lecture Notes in Statist. 101, Springer-Verlag, New York, 1995.

[3] A. Ben-Tal, A. Nemirovski, and C. Roos, *Extended matrix cube theorems with applications to μ-theory in control*, Math. Oper. Res., 28 (2003), pp. 497–523.

[4] D. Bertsimas and Y. Ye, *Semidefinite relaxations, multivarite normal distribution, and order statistics*, in Handbook of Combinatorial Optimization, Vol. 3, D. Z. Du and P. M. Pardalos, eds., Kluwer Academic Publishers, Boston, 1998, pp. 1–19.

[5] M. Charikar and A. Wirth, *Maximizing quadratic programs: Extending Grothendieck's inequality*, in Proceedings of the 45th Annual FOCS, 2004.

[6] A. Frieze and M. Jerrum, *Improved approximation algorithms for MAX-k-CUT and Max BISECTION*, Algorithmica, 18 (1997), pp. 67–81.

[7] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, 1979.

[8] M. X. Goemans and D. P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. ACM, 42 (1995), pp. 1115–1145.

[9] M. X. Goemans and D. P. Williamson, *Approximation algorithms for MAX-3-CUT and other problems via complex semidefinite programming*, J. Comput. System Sci., 68 (2004), pp. 442–470.

[10] Z. Q. Luo, X. D. Luo, and M. Kisialiou, *An efficient quasi-maximum likelihood decoder for PSK signals*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03), Vol. 6, 2003, pp. 561–564.

[11] Yu. Nesterov, *Semidefinite relaxation and nonconvex quadratic optimization*, Optim. Methods Softw., 9 (1998), pp. 141–160.

[12] A. So, J. Zhang, and Y. Ye, *On approximating complex quadratic optimization problems via semidefinite programming relaxations*, in Proceedings of the 11th Conference on Integer Programming and Combinatorial Optimization, Lecture Notes in Comput. Sci. 3509, M. Jünger and V. Kaibel, eds., Springer-Verlag, Berlin, 2005, pp. 125–135.

[13] O. Toker and H. Özbay, *On the complexity of purely complex μ computation and related problems in multidimensional systems*, IEEE Trans. Automat. Control, 43 (1998), pp. 409–414.

[14] Y. Ye, *Approximating quadratic programming with bound and quadratic constraints*, Math. Program., 84 (1999), pp. 219–226.

[15] S. Zhang, *Quadratic maximization and semidefinite relaxation*, Math. Program., 87 (2000), pp. 453–465.

SIAM J. OPTIM.
Vol. 16, No. 3, pp. 891–909

OPTIMALITY MEASURES FOR PERFORMANCE PROFILES[*]

ELIZABETH D. DOLAN[†], JORGE J. MORÉ[‡], AND TODD S. MUNSON[‡]

Abstract. We examine the importance of optimality measures when benchmarking a set of solvers and show that the scale-invariance requirements we impose lead to a convergence test for nonlinearly constrained optimization solvers that uses a mixture of absolute and relative error measures. We demonstrate that this convergence test is well behaved at any point where the constraints satisfy the Mangasarian–Fromovitz constraint qualification and also avoids the explicit use of a complementarity measure. Computational experiments explore the impact of this convergence test on the benchmarking process with performance profiles.

Key words. benchmarking, optimality conditions, performance profiles, scale invariance

AMS subject classifications. 90C30, 90C46

DOI. 10.1137/040608015

1. Introduction.

Benchmarking is essential when developing numerical software because this process reveals the strengths and weaknesses of the software. To obtain useful information from the benchmark, attention must be given to the convergence tests and tolerances used by competing solvers. In particular, a comparison between a solver that computes a highly accurate solution and another that computes an inaccurate solution can be misleading.

We propose a convergence test for benchmarking nonlinearly constrained optimization solvers and require that the approximate solution returned by each solver satisfies this convergence test. While we are primarily interested in convergence tests that can be used by solvers for the constrained optimization problem

$$\min \{ f(x) : l \le c(x) \le u \} , \tag{1.1}$$

our remarks apply to the benchmarking of general iterative solvers.

We could introduce a uniform convergence test in the benchmarking process by modifying all the solvers in the benchmark. This approach is not feasible, however, unless we have access to the source codes for all the solvers. Even with this access, adding a new convergence test requires detailed knowledge of the solver. In general, only the developers can reliably modify their codes. Furthermore, the cost of applying a convergence test may be significant if some of the required information is not readily available and must be computed, resulting in a noticeable increase in the total time taken to solve the benchmark problem.

An alternative approach is to compute and check a specific convergence test for all the solvers a posteriori. In this approach each solver is first used with default tolerances. If the approximate solution returned by the solver does not satisfy the

[*] Received by the editors May 11, 2004; accepted for publication (in revised form) July 8, 2005; published electronically January 12, 2006. This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under contract W-31-109-Eng-38, and by National Science Foundation grant DMI-0322580. T
http://www.siam.org/journals/siopt/16-3/60801.html

[†] Department of Computer Science, The University of North Carolina at Chapel Hill, Chapel Hill, NC 27599 (dolan@cs.unc.edu).

[‡] Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439 (more@mcs.anl.gov, tmunson@mcs.anl.gov).

a posteriori convergence test, then the native solver tolerances are reduced and the problem is solved again. This process is repeated until the a posteriori convergence test is satisfied or a time limit is exceeded. This approach guarantees that the approximate solutions returned by all the solvers in the benchmark satisfy the same convergence test. Moreover, this approach can be readily implemented because access to the source code for each solver is not required.

We impose two major requirements on the convergence test. The first requirement is that the accuracy be based only on the approximate solution returned by the solver; all other quantities needed to assess the level of accuracy, such as multiplier estimates, must be computed independently. The other requirement is scale invariance of the convergence test when either the function and constraints are scaled or when the variables are scaled.

Our convergence test for constrained optimization problems is based on the first-order optimality conditions but uses a mixture of absolute and relative error measures. Section 2 defines these measures and uses them to define τ-active constraints and the associated multiplier estimates. We have used the nonstandard term τ-active constraint because the term ϵ-active constraint is invariably related to absolute error measures.

We define a convergence test in section 3 in terms of measures for feasibility and stationarity that takes into account the relative size of the constraints. We show that this convergence test is well behaved at any point where the constraints satisfy the Mangasarian–Fromovitz constraint qualification, but may fail if this constraint qualification does not hold. In section 4 we examine the relationship of our convergence test to other tests commonly used in optimization software and show that our convergence test is always satisfied in a neighborhood of a Karush–Kuhn–Tucker (KKT) point.

Section 5 examines the scale-invariance properties of the convergence tests introduced in section 3. We show that these convergence tests are scale invariant under reasonable conditions. We also examine the scale-invariance properties of other convergence tests.

Section 6 describes a benchmarking process based on the convergence test introduced in section 3. We use performance profiles from [5] and Version 3.0 of COPS [6], the Constrained Optimization Problem Set, to evaluate the effect of the convergence tests. Performance profiles have been used in a wide variety of benchmarking studies (for example, [1, 2, 13, 14]), but in almost all cases the convergence criteria have not been specified in detail. This is certainly the case for benchmarking studies in the AMPL or GAMS modeling environments because in these cases there is no easy access to the source codes for the solvers. Our computational results in section 7 show the importance of using the same convergence test for all solvers in the benchmark because some of them have widely different notions of optimality with their default tolerances. Our results show that the trends in the performance profiles remain the same but that important differences arise with a uniform convergence test.

2. Approximate active sets and multipliers.

Given τ in $(0, 1)$ and an approximate solution x to the general optimization problem (1.1), we define measures of optimality in terms of the set of τ-active constraints and associated multiplier estimates.

The set of τ-active constraints at x consists of all constraints near the bounds l and u as measured by τ. The measure of nearness we propose involves a mixture of the absolute and relative errors. Given real numbers ξ_1 and ξ_2, we define

$$\delta[\xi_1,\xi_2] = \min\left\{ |\xi_1-\xi_2|, \frac{|\xi_1-\xi_2|}{|\xi_1|+|\xi_2|} \right\},$$

with $\delta[0,0]=0$. We extend this definition by continuity to set $\delta[\xi_1,\xi_2]=1$ if either ξ_1 or ξ_2 is infinite. Thus, $\delta[\xi_1,\xi_2]\in[0,1]$, provided one of the arguments of $\delta[\cdot,\cdot]$ is finite. Moreover,

$$\delta[\xi_1,\xi_2] \le \tau \quad \text{if and only if} \quad |\xi_1-\xi_2| \le \tau \max\{1, |\xi_1|+|\xi_2|\}.$$

In particular, $\delta[\xi_1,\xi_2]$ is the relative error whenever $|\xi_1|+|\xi_2|\ge 1$.

Given x, y in $\mathbb{R}^n$, we extend this definition of error to vectors by defining a vector-valued error measure $d : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}^n$ by

$$d[x,y] = (d_k[x,y]) = (\delta[x_k,y_k])$$

so that $d[x,y]$ is a vector where $d_k[x,y] = \delta[x_k,y_k]$ for each component k. This definition implies that $d[x,y]=|x-y|$ for vectors of modest size, that is, $\|x\|_\infty + \|y\|_\infty \le 1$. Moreover, if $\|\cdot\|$ is a monotone norm ($|x|\le|y|$ implies $\|x\|\le\|y\|$), then

$$\min\left\{ \|x-y\|,\ \frac{\|x-y\|}{\|x\|_\infty+\|y\|_\infty} \right\} \le \|d[x,y]\| \le \min\left\{ \|x-y\|, \left\| \left(\frac{|x_k-y_k|}{|x_k|+|y_k|} \right) \right\| \right\}. \tag{2.1}$$

This inequality shows that $\|d[x,y]\|$ is closely related to the absolute and relative error between x and y.

This definition of error between vectors can be modified in various ways. In particular, if we wish to introduce different levels of absolute and relative errors by requiring that

$$|\xi_1-\xi_2| \le \tau_a\tau \quad \text{or} \quad \frac{|\xi_1-\xi_2|}{|\xi_1|+|\xi_2|} \le \tau \tag{2.2}$$

for some $\tau_a \ge 0$, then we can define

$$\delta[\xi_1,\xi_2] = \min\left\{ \left(\frac{1}{\tau_a}\right)|\xi_1-\xi_2|, \frac{|\xi_1-\xi_2|}{(|\xi_1|+|\xi_2|)} \right\}. \tag{2.3}$$

The case where $\tau_a=0$ is equivalent to using a purely relative error test and can be obtained as the limit of $\delta[\xi_1,\xi_2]$ as τ_a converges to zero.

If we use the definition (2.3), then the inequality $\delta[\xi_1,\xi_2]\le\tau$ is equivalent to (2.2). Moreover, we now have

$$\delta[\xi_1,\xi_2] \le \tau \quad \text{if and only if} \quad |\xi_1-\xi_2| \le \tau \max\{\tau_a, |\xi_1|+|\xi_2|\}.$$

This equivalence shows that the test $\delta[\xi_1,\xi_2]\le\tau$ reduces to the relative convergence test in (2.2), unless $|\xi_1|+|\xi_2|\le\tau_a$, and then becomes the absolute convergence test in (2.2). We can emphasize the relative error in (2.3) by choosing $\tau_a\le\tau$.

We illustrate definition (2.3) by considering two vectors with components of different magnitudes. Consider, for example, vectors

$$x = \begin{pmatrix} \alpha_1\mu \\ \beta_1 \end{pmatrix}, \qquad y = \begin{pmatrix} \alpha_2\mu \\ \beta_2 \end{pmatrix},$$

where α_s and β_s lie in $[1,2]$ for $s=1,2$. For these two vectors we want to compare the componentwise condition $\|d[x,y]\|_\infty \le \epsilon$ with the condition

$$\|x-y\|_\infty \le \epsilon\left(\|x\|_\infty + \|y\|_\infty\right). \tag{2.4}$$

Note that $\|d[x,y]\|_\infty \le \epsilon$ if and only if

$$|\alpha_1-\alpha_2| \le \epsilon \max\{\tau_a, |\alpha_1|+|\alpha_2|\}, \qquad |\beta_1-\beta_2| \le \epsilon \max\{\tau_a, |\beta_1|+|\beta_2|\},$$

independent of the size of μ. On the other hand, if μ is large, then

$$\|x-y\|_\infty \le \epsilon\left(\|x\|_\infty + \|y\|_\infty\right) \quad \text{if and only if} \quad |\alpha_1-\alpha_2| \le \epsilon\left(|\alpha_1|+|\alpha_2|\right);$$

while if μ is small, then

$$\|x-y\|_\infty \le \epsilon\left(\|x\|_\infty + \|y\|_\infty\right) \quad \text{if and only if} \quad |\beta_1-\beta_2| \le \epsilon\left(|\beta_1|+|\beta_2|\right).$$

Comparing $\|d[x,y]\|_\infty \le \epsilon$ with (2.4) shows that $\|d[x,y]\|_\infty \le \epsilon$ leads to restrictions on all components, while (2.4) restricts only the larger components.

The merits of the componentwise error measure $\|d[x,y]\|_\infty$ depend on the application. If the user wants the same accuracy in all components, then a componentwise error measure is preferable. However, if the user cares mainly about the larger components, then a componentwise error measure could lead to excessive accuracy in the small components.

Componentwise error measures have not received attention in the optimization community but they have been used in error analysis and perturbation analysis. See, for example, Higham [11, Chapter 7]. In this work we propose the use of componentwise error measures for convergence tests. We show in section 5 that an advantage of (2.3) is that it leads to the scale invariance of convergence tests based on $\delta[\cdot,\cdot]$ if $\tau_a = 0$. We set $\tau_a = 1$ in the remainder of this paper, but all results can be easily modified for any $\tau_a \ge 0$.

Given τ in $(0,1)$ and the definition $d : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}^n$ of error between vectors, we define the set of τ-active constraints at x by

$$\mathcal{A}_\tau(x) = \{k : \min\{d_k[c(x),l], d_k[c(x),u]\} \le \tau\}. \tag{2.5}$$

In general, τ is related to the expected accuracy of the optimization algorithm because the set $\mathcal{A}_\tau(x)$ contains all constraints that are nearly active as measured by τ. Moreover, for τ sufficiently small, $\mathcal{A}_\tau(x)$ is the set $\mathcal{A}_0(x)$ of active constraints at x.

This definition of τ-active constraints for the optimization problem (1.1) reduces to standard notions for optimization problems in generic form. For example, if we consider

$$\min\{f(x) : c(x) \le 0\}, \tag{2.6}$$

then $d_k[c(x),0] = \min\{|c_k(x)|, 1\}$, and thus $\mathcal{A}_\tau(x) = \{k : |c_k(x)| \le \tau\}$. The more general definition (2.5) is needed for dealing with problems that have not been put into this standard form.

We measure optimality by computing multiplier estimates explicitly with the requirement that the multipliers lie in the cone $S_\tau(x)$ associated with $\mathcal{A}_\tau(x)$, where

$$S_\tau(x) = \left\{ v : \begin{array}{lll} v_k \text{ free} & \text{if} & d_k[c(x),l] \le \tau,\ d_k[c(x),u] \le \tau, \\ v_k \ge 0 & \text{if} & d_k[c(x),l] \le \tau,\ d_k[c(x),u] > \tau, \\ v_k \le 0 & \text{if} & d_k[c(x),l] > \tau,\ d_k[c(x),u] \le \tau, \\ v_k = 0 & \text{if} & d_k[c(x),l] > \tau,\ d_k[c(x),u] > \tau. \end{array} \right. \tag{2.7}$$

If the kth constraint is an equality constraint and is τ-active, then $l_k = u_k$, and thus the kth component is free in the cone $S_\tau(x)$. Moreover, the kth component is zero if the kth constraint is not τ-active. The following result records some useful properties of the cone $S_\tau(x)$.

THEOREM 2.1. *If the cone $S_\tau(\cdot)$ is defined by* (2.7) *and $x^* \in \mathbb{R}^n$, then the following hold:*

(a) *$S_{\tau_1}(x) \subset S_{\tau_2}(x)$ for $\tau_1 \le \tau_2$.*

(b) *If $\{(x_k, v_k, \tau_k)\}$ converges to (x^*, v^*, τ^*) with $v_k \in S_{\tau_k}(x_k)$, then $v^* \in S_{\tau^*}(x^*)$.*

(c) *For any $\tau > 0$ we have $S_0(x^*) \subset S_\tau(x)$ for x in some neighborhood $N(x^*)$.*

Proof. The proof of the first part is a consequence of the definition of the cone $S_\tau(x)$. We prove the second part by noting that if $v^* \notin S_{\tau^*}(x^*)$, then there is an index p such that

$$v_p^* > 0, \quad d_p[c(x^*), l] > \tau^* \qquad \text{or} \qquad v_p^* < 0, \quad d_p[c(x^*), u] > \tau^*.$$

We now show that this situation is not possible. Consider only the case where $v_p^* > 0$ since the case where $v_p^* < 0$ is similar. If $v_p^* > 0$, then the pth component of v_k is positive for all k sufficiently large, and since $v_k \in S_{\tau_k}(x_k)$, we have $d_p[c(x_k), l] \le \tau_k$. Hence, $d_p[c(x^*), l] \le \tau^*$, a contradiction.

We turn to the proof of the last part of this result. First, note that definition (2.7) shows that $S_\tau(x)$ is the product of closed intervals $I_k(x, \tau)$, where each interval is associated with a constraint. For example, if $d_k[c(x), l] \le \tau$ and $d_k[c(x), u] \le \tau$, then $I_k(x, \tau) = \mathbb{R}$. In general $I_k(x, \tau)$ is either $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{R}_-$, or $\{0\}$. We prove that $I_k(x^*, 0) \subset I_k(x, \tau)$.

Assume, for example, that $I_k(x^*, 0) = \mathbb{R}_+$. In this case, we have $d_k[c(x^*), l] = 0$ and $d_k[c(x^*), u] > 0$. Now choose $N(x^*)$ such that $d_k[c(x), l] \le \tau$ for all $x \in N(x^*)$. Then

$$I_k(x, \tau) = \begin{cases} \mathbb{R}_+ & \text{if} \quad d_k[c(x), l] \le \tau, \ d_k[c(x), u] > \tau, \\ \mathbb{R} & \text{if} \quad d_k[c(x), l] \le \tau, \ d_k[c(x), u] \le \tau. \end{cases}$$

Hence, $I_k(x^*, 0) \subset I_k(x, \tau)$ as desired. The proof that $I_k(x^*, 0) \subset I_k(x, \tau)$ in other cases is similar. □

We motivate the definition of multiplier estimates by first recalling that a KKT pair (x^*, λ^*) for the optimization problem (1.1) satisfies

$$\nabla f(x^*) = \nabla c(x^*) \lambda^*, \qquad \lambda^* \in S_0(x^*),$$

where λ^* are the multipliers. We determine multiplier estimates via the optimization problem

$$\min \left\{ \|\nabla f(x) - \nabla c(x) v\| : v \in S_\tau(x) \right\}, \tag{2.8}$$

where $\|\cdot\|$ is an arbitrary norm and $\tau \ge 0$. Note that computing multiplier estimates via the bound-constrained problem (2.8) has been proposed by others, at least in the case $\tau = 0$. See, for example, [10, page 250] and [3, page 474].

If we let $\lambda(x, \tau)$ be the multiplier estimates obtained by solving (2.8), then we can measure optimality via the residual

$$r(x, \tau) = \nabla f(x) - \nabla c(x) \lambda(x, \tau). \tag{2.9}$$

A short computation shows that if x is feasible and τ is sufficiently small, then $r(x, \tau) = 0$ if and only if $(x, \lambda(x, \tau))$ is a KKT pair. If we consider the special case of bound-constrained optimization problems where the constraints are $c(x) = x$, then

$$r(x,\tau) = \begin{cases} 0 & \text{if} \quad d_k[x,l] \le \tau, \ d_k[x,u] \le \tau, \\ \min(0, \partial_k f(x)) & \text{if} \quad d_k[x,l] \le \tau, \ d_k[x,u] > \tau, \\ \max(0, \partial_k f(x)) & \text{if} \quad d_k[x,l] > \tau, \ d_k[x,u] \le \tau, \\ \partial_k f(x) & \text{if} \quad d_k[x,l] > \tau, \ d_k[x,u] > \tau \end{cases}$$

for any l_p norm ($p < \infty$) in (2.8), where $\partial_k f(x)$ denotes the partial derivative of f with respect to the kth argument. This expression shows that $-r(x,0)$ agrees with the projected gradient as used, for example, by Chin and Moré [12, page 1105]. Thus, $-r(x,0)$ can be interpreted as a generalization of the projected gradient to nonlinearly constrained optimization problems.

The choice of the l_2 norm in the computation of the multipliers in (2.8) leads to a bound-constrained least squares problem. We prefer to use the l_∞ norm and thus define $\lambda(x,\tau)$ as a solution of

$$\min \left\{ \|y\|_\infty : y = \nabla f(x) - \nabla c(x) v, \ v \in S_\tau(x) \right\}. \tag{2.10}$$

This problem can be formulated as a linear programming problem, and thus $\lambda(x,\tau)$ can be readily computed by several solvers. The solution $\lambda(x,\tau)$ to (2.10) is a set of multiplier estimates for the original optimization problem.

3. Optimality measures.

We define a convergence test for the optimization problem (1.1) in terms of measures of feasibility, complementarity, and stationarity that takes into account the relative size of the constraints. Given tolerances $\tau_1, \ldots, \tau_p$ and measures of optimality $\nu_i : \mathbb{R}^n \mapsto \mathbb{R}_+$, a convergence test defines a set

$$\mathcal{C}(\tau) = \{ x \in \mathbb{R}^n : \nu_i(x) \le \tau_i, \ 1 \le i \le p \}$$

of acceptable points. A minimal requirement on the convergence test is that $\mathcal{C}(0)$ contain only KKT points.

The standard measure of feasibility for constraints of the form $l \le c(x) \le u$ can be written in the form

$$\| \text{mid}\{ c(x) - l, 0, c(x) - u \} \|,$$

where mid$\{\cdot\,,\cdot,\cdot\}$ denotes the argument in the middle, that is, the median of the three arguments. Thus, in particular, mid$\{\alpha, \beta, \gamma\}$ is γ if $\alpha \le \gamma \le \beta$. We introduce the relative size of the constraints by defining the *feasibility measure*

$$\nu_f(x) = \|v(x)\|,$$

where $\|\cdot\|$ is any norm and

$$v_k(x) = \begin{cases} 0 & \text{if} \quad l_k \le c_k(x) \le u_k, \\ \min(d_k[c(x),l], d_k[c(x),u]) & \quad\ \text{otherwise.} \end{cases}$$

Note that $\nu_f(x) = 0$ if and only if x is feasible. Moreover, if $\|\cdot\|$ is a monotone norm, then

$$\nu_f(x) \le \| \text{mid}\{ c(x) - l, 0, c(x) - u \} \|,$$

and thus $\nu_f(x)$ is bounded above by the norm of the constraint violation. We define a vector $x \in \mathbb{R}^n$ to be τ-feasible if $\nu_f(x) \le \tau$.

In most cases we use the l_∞ norm. For this norm a τ-feasible vector is precisely a vector that satisfies

$$l_k \le c_k(x) \le u_k \quad \text{or} \quad \min\{d_k[c(x), l], d_k[c(x), u]\} \le \tau, \qquad 1 \le k \le n.$$

If we consider the generic optimization problem (2.6) and any $\tau \in (0, 1)$, then x is τ-feasible if and only if $\|c(x)_+\| \le \tau$. The equivalence of norms in finite dimensions shows that these results hold for any norm, provided τ is sufficiently small. Finally, $0 \le \nu_f(x) \le 1$ with the l_∞ norm.

An advantage of computing the multipliers by either (2.8) or (2.10) is that all the multipliers of the τ-active constraints have the proper sign. Moreover, $\lambda_k(x) = 0$ if the kth constraint is not τ-active. Hence, complementarity should be defined in terms of the constraint violation for the τ-active constraints. We define

$$\nu_c(x, \tau) = \|w(x, \tau)\|$$

as a *measure of complementarity*, where

$$w_k(x, \tau) = \begin{cases} \min(d_k[c(x), l], d_k[c(x), u]) & k \in \mathcal{A}_\tau(x), \\ 0 & \text{otherwise}. \end{cases}$$

The definition of $\mathcal{A}_\tau(x)$ implies that $\nu_c(x, \tau) \le \tau$ when we use the l_∞ norm, and thus the complementarity measure ν_c is never large.

An optimization algorithm should deliver approximate solutions that are τ-feasible, that is, $\nu_f(x) \le \tau$. For τ-feasible vectors in the l_∞ norm we have

$$\nu_f(x) \le \nu_c(x, \tau) \le \tau,$$

and thus the complementarity measure dominates for τ-feasible vectors.

The standard method for measuring stationarity uses some norm of the difference between $\nabla f(x)$ and $\nabla c(x)\lambda(x, \tau)$, typically with $\tau = 0$. However, in order to take into account the relative size of $\nabla f(x)$, we use the *stationarity measure*

$$\nu_s(x, \tau) = \|d[\nabla f(x), \nabla c(x)\lambda(x, \tau)]\|.$$

A short computation shows that if x is feasible and τ is sufficiently small, then $\nu_s(x, \tau) = 0$ if and only if $(x, \lambda(x, \tau))$ is a KKT pair. Also note that if $\|\cdot\|$ is a monotone norm, then the definition of the function $d : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$ implies that

$$\nu_s(x, \tau) \le \|\nabla f(x) - \nabla c(x)\lambda(x, \tau)\|$$

so that $\nu_s(x, \tau)$ is bounded above by a standard measure of stationarity.

Table 3.1 summarizes all the measures of optimality for an optimization problem. We have defined these measures in terms of an arbitrary norm, but we use the l_∞ norm. We define a convergence test in terms of tolerances τ_f and τ_s by computing multipliers via (2.10) with the τ-active set determined by (2.5) with $\tau = \tau_f$, and requiring that

$$\nu_f(x) \le \tau_f, \qquad \nu_s(x, \tau_f) \le \tau_s. \tag{3.1}$$

The definition of ν_c guarantees that $\nu_c(x, \tau_f) \le \tau_f$ in the l_∞ norm, and thus it is not necessary to require a test on complementarity. This statement seems to be incorrect

Table 3.1
Measures of optimality.

Feasibility	$\nu_f(x)$
Complementarity	$\nu_c(x,\tau)$
Stationarity	$\nu_s(x,\tau)$

at first sight but is a consequence of using a set of multipliers that are zero if the constraint is not τ-active but otherwise have the proper sign.

The following result shows that the optimality measures in Table 3.1 behave appropriately if we consider a sequence of tolerances that converge to zero. We assume that we have a sequence $\{x_k\}$ that converges to a point x^* and that $f : \mathbb{R}^n \mapsto \mathbb{R}$ and $c : \mathbb{R}^n \mapsto \mathbb{R}^m$ are continuously differentiable in a neighborhood of x^*. We also assume that x^* satisfies the Mangasarian–Fromovitz constraint qualification at x^* in the sense that

$$\nabla c(x^*)v = 0, \quad v \in S_0(x^*) \qquad \Longrightarrow \qquad v = 0,$$

where $S_0(x^*)$ is the cone (2.7) at $\tau = 0$. This constraint qualification reduces to the classical Mangasarian–Fromovitz constraint qualification for the generic optimization problem (2.6).

Theorem 3.1. *Assume that $\tau_a > 0$ in* (2.2) *and that $f : \mathbb{R}^n \mapsto \mathbb{R}$ and $c : \mathbb{R}^n \mapsto \mathbb{R}^m$ are continuously differentiable in a neighborhood of x^*. Let $\{\tau_k\}$ be a sequence of tolerances that converges to zero, and let $\{x_k\}$ be a sequence that converges to x^*. If $\{\nu_f(x_k)\}$ converges to zero, then x^* is feasible. Moreover, if $\{\nu_s(x_k,\tau_k)\}$ converges to zero and the constraints satisfy the Mangasarian–Fromovitz constraint qualification at x^*, then x^* is a KKT point of the optimization problem* (1.1).

Proof. Since $\tau_a > 0$, the definition (2.2) of $\delta[\cdot,\cdot]$ implies that $d[\cdot,\cdot]$ preserves convergent sequences; that is, $\{d[y_k, y^*]\}$ converges to zero if and only if $\{y_k\}$ converges to y^*. Hence, the definition of ν_f shows that x^* is feasible.

We now assume that $\{\nu_s(x_k,\tau_k)\}$ converges to zero and show that x^* is a KKT point of the optimization problem (1.1) if the constraints satisfy the Mangasarian–Fromovitz constraint qualification at x^*. We first show that $\{\lambda(x_k,\tau_k)\}$ is bounded by noting that since $S_{\tau_k}(x_k)$ is a cone,

$$\|\nabla f(x_k) - \nabla c(x_k)\lambda(x_k,\tau_k)\| \le \|\nabla f(x_k)\|,$$

and thus, $\{\nabla c(x_k)\lambda(x_k,\tau_k)\}$ is bounded. Since $\{x_k\}$ converges to x^* and the constraints satisfy the Mangasarian–Fromovitz constraint qualification at x^*, the sequence $\{\lambda(x_k,\tau_k)\}$ is bounded. We also note that since $\lambda(x_k,\tau_k)$ belongs to the cone $S_{\tau_k}(x_k)$, Theorem 2.1 shows that any limit point λ^* of $\{\lambda(x_k,\tau_k)\}$ is a valid set of multipliers for x^*, that is, $\lambda^* \in S_0(x^*)$.

We have shown that $\{\lambda(x_k,\tau_k)\}$ is bounded and that any limit point of this sequence is a valid set of multipliers for x^*. Now note that (2.1) guarantees that there is a constant $\sigma > 0$ such that

$$\sigma\|\nabla f(x_k) - \nabla c(x_k)\lambda(x_k,\tau_k)\| \le \nu_s(x_k,\tau_k).$$

Since $\{\nu_s(x_k,\tau_k)\}$ converges to zero, this inequality shows that any limit point λ^* of the sequence $\{\lambda(x_k,\tau_k)\}$ satisfies $\nabla f(x^*) = \nabla c(x^*)\lambda^*$, and since $\lambda^* \in S_0(x^*)$, we have shown that x^* is a KKT point of the optimization problem (1.1). □

We can generalize Theorem 3.1 by noting that the proof of this result shows that ν_s is lower semicontinuous, that is,

$$\liminf_{(x,\tau)\to(x^*,0)} \nu_s(x,\tau) \geq \nu_s(x^*,0).$$

This property is of interest because simple examples show that ν_s is not continuous in (x,τ), although $\nu_s(\cdot,\tau)$ is certainly continuous. For example, if $\{x_k\}$ is a sequence that converges to a KKT point x^* such that x_k is not on the boundary of the feasible set, then we can choose τ_k so that $\{\tau_k\}$ converges to zero and $S_{\tau_k}(x_k) = \{0\}$. In this case $\{\nu_s(x_k,\tau_k)\}$ is bounded away from zero if $\nabla f(x^*) \neq 0$, but $\nu_s(x^*,0) = 0$.

The assumption that the constraints satisfy the Mangasarian–Fromovitz constraint qualification at x^* is essential for Theorem 3.1. Consider, for example, the optimization problem in $\mathbb{R}$,

$$\min\left\{\xi : \tfrac{1}{2}\xi^2 \geq 0\right\}.$$

If $\{\xi_k\}$ is any monotone sequence that converges to zero and $\tau_k = \xi_k$, then the multipliers determined by (2.10) are $\lambda(\xi) = 1/\xi$. Thus, in this case, $\{\nu_s(\xi_k,\tau_k)\}$ converges to zero, but $\xi^* = 0$ is not a KKT point. Of course, in this case the Mangasarian–Fromovitz constraint qualification fails and the multiplier estimates are unbounded.

4. Convergence tests. We have defined the convergence test (3.1) in terms of tolerances τ_f and τ_s. In this section we explore the relationship between the optimality conditions in Table 3.1 and the convergence tests used in optimization solvers.

Given a set of input values (for example, an approximate solution, multiplier estimates, tolerances, ...), an optimization solver generates a sequence of iterates $x_0, x_1, \ldots, x(\tau)$, where τ is the vector of tolerances. In some cases the tolerances are used only to determine when to stop the iteration process and return an approximate solution $x(\tau)$. In this case only $x(\tau)$ is dependent on the tolerances. However, the solver could also use the input tolerances to dictate how the solver behaves in certain situations. In this case all the iterates are potentially dependent on the input tolerances. Our discussion of convergence tests in this section applies to both kinds of optimization algorithms.

We assume infinite-precision arithmetic in all of our discussions since a discussion of convergence behavior under rounding error is outside the scope of this study. We note only that rounding errors may cause the solver to fail if the tolerances are too small or the computation of the function has too much noise.

In our benchmarking results in section 7 we study the performance of optimization solvers as the tolerances are gradually decreased. We now show that if some subsequence of the approximate solutions $x(\tau)$ generated by the solver converges to a KKT point, then there is an approximate solution $x(\tau)$ that satisfies the convergence test (3.1).

THEOREM 4.1. *Let $\tau > 0$ be given, and assume that $f : \mathbb{R}^n \mapsto \mathbb{R}$ and $c : \mathbb{R}^n \mapsto \mathbb{R}^m$ are continuously differentiable in a neighborhood of a KKT point x^* of problem* (1.1). *If $\{x_k\}$ is a sequence that converges to x^*, then $\{\nu_f(x_k)\}$ and $\{\nu_s(x_k,\tau)\}$ converge to zero.*

Proof. If $\{x_k\}$ is a sequence that converges to a feasible point x^* of the optimization problem (1.1), then clearly $\{\nu_f(x_k)\}$ converges to zero.

We now show that $\{\nu_s(x_k,\tau)\}$ converges to zero. We have already noted that if $\|\cdot\|$ is a monotone norm, then the definition of the function $d : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$ implies that

$$\nu_s(x_k,\tau) \le \|\nabla f(x_k) - \nabla c(x_k)\lambda(x_k,\tau)\|.$$

Moreover, the definition (2.10) of the multiplier $\lambda(x_k,\tau)$ shows that

$$\|\nabla f(x_k) - \nabla c(x_k)\lambda(x_k,\tau)\| \le \|\nabla f(x_k) - \nabla c(x_k)\lambda\|, \qquad \lambda \in S_\tau(x_k).$$

Hence, these two inequalities show that

$$\nu_s(x_k,\tau) \le \|\nabla f(x_k) - \nabla c(x_k)\lambda\|, \qquad \lambda \in S_\tau(x_k).$$

Now note that if λ^* is a multiplier associated with the KKT point x^*, then $\lambda^* \in S_0(x^*)$, and thus Theorem 2.1 guarantees that for all k sufficiently large we have $\lambda^* \in S_\tau(x_k)$. Hence, the previous inequality implies that

$$\nu_s(x_k,\tau) \le \|\nabla f(x_k) - \nabla c(x_k)\lambda^*\|.$$

Thus, since $\{x_k\}$ converges to x^* and λ^* is a multiplier associated with x^*, we must have that $\{\nu_s(x_k,\tau)\}$ converges to zero as desired. □

We now relate the optimality conditions in Table 3.1 to convergence tests used in optimization algorithms. Consider, for example, the generic optimization problem

$$\min\left\{ f(x) : c(x) \le 0 \right\}.$$

Given tolerances τ_f, τ_c, and τ_s, assume that the convergence tests are

$$c_k(x) \le \tau_f, \qquad \pi_k(x) \le \tau_c, \qquad |\pi_k(x)| \le \tau_c \quad \text{if} \quad |c_k(x)| > \tau_f \tag{4.1}$$

on the approximate solution x and multiplier estimate $\pi(x)$, and

$$\|\nabla f(x) - \nabla c(x)\pi(x)\|_\infty \le \tau_s \tag{4.2}$$

on the residual (2.9). These are suitable convergence tests in the sense that if all the tolerances are set to zero, then we recover the KKT conditions.

An important difference between these convergence tests and the optimality measures in Table 3.1 is that the multiplier estimates $\pi_k(x)$ are not guaranteed to be nonpositive. However, if we define

$$\lambda_k(x) = \begin{cases} \min(\pi_k(x), 0) & \text{if} \quad k \in \mathcal{A}_{\tau_f}(x), \\ 0 & \qquad\text{otherwise,} \end{cases}$$

then $\lambda_k(x) \le 0$ are multiplier estimates with

$$|\lambda_k(x) - \pi_k(x)| \le \tau_c. \tag{4.3}$$

This estimate holds if $k \in \mathcal{A}_{\tau_f}(x)$ and $\pi_k(x) \le 0$ because then $\lambda_k(x) = \pi_k(x)$. If $k \notin \mathcal{A}_{\tau_f}(x)$ or $\pi_k(x) > 0$, then $\lambda_k(x) = 0$. Moreover, in either case, (4.1) implies that $|\pi_k(x)| \le \tau_c$. Hence, (4.3) also holds in this case.

The estimate (4.3) shows that the residual (2.9) is bounded in terms of the tolerances and the problem data. Indeed, a direct consequence of (4.2) and (4.3) is that

$$\|\nabla f(x) - \nabla c(x)\lambda(x)\|_\infty \le \tau_s + \|\nabla c(x)\|_\infty \tau_c.$$

Hence, we have shown that if the convergence tests (4.1) and (4.2) hold, then

$$\nu_f(x) \le \tau_f, \qquad \nu_s(x, \tau_f) \le \tau_s + \|\nabla c(x)\|_\infty \tau_c. \tag{4.4}$$

This is an important observation because (4.1) and (4.2) are closely related to convergence tests used by optimization solvers such as SNOPT and KNITRO. For example, instead of (4.1), SNOPT [9] requires that

$$c_k(x) \le \tau_f, \qquad \pi_k(x) \le \tau_c, \qquad |c_k(x)\pi_k(x)| \le \tau_c. \tag{4.5}$$

This is a stronger convergence test than (4.1) because if (4.5) holds, then

$$|\pi_k(x)| \le \frac{\tau_c}{|c_k(x)|},$$

and thus $|\pi_k(x)|$ will be forced to be much smaller than τ_c if $|c_k(x)|$ is much larger than τ_f. We also note that (4.5) not only implies (4.1) but also implies a bound on the multipliers of the τ-active constraints. Assume, for example, that the kth constraint is τ-active with

$$|c_k(x)| = \sigma \tau_f, \qquad \sigma \in (0, 1).$$

Under this assumption, (4.5) implies that

$$|\pi_k(x)| \le \frac{1}{\sigma} \frac{\tau_c}{\tau_f}.$$

Thus, for problems with large multipliers, this bound shows that τ_f may have to be relatively small in order to satisfy (4.5).

Similar remarks apply to the optimization solver KNITRO [18]. The convergence test in KNITRO replaces (4.1) by

$$c_k(x) \le \tau_f, \qquad \pi_k(x) < 0, \qquad |c_k(x)\pi_k(x)| \le \tau_c. \tag{4.6}$$

Thus, the only difference between the convergence test in SNOPT and KNITRO is that KNITRO guarantees that the multipliers are negative. For the KNITRO convergence test we can show that if

$$\lambda_k(x) = \begin{cases} \pi_k(x) & \text{if} \quad k \in \mathcal{A}_{\tau_f}(x), \\ 0 & \qquad \text{otherwise}, \end{cases}$$

then (4.3) holds. Hence, (4.4) also holds.

The convergence tests in SNOPT and KNITRO require the user to choose tolerances τ_P and τ_D. SNOPT sets

$$\tau_f = \tau_P(1 + \|x\|), \qquad \tau_c = \tau_s = \tau_D(1 + \|\pi(x)\|),$$

while KNITRO sets

$$\tau_f = \max\{\tau_P \max(1, \|c(x_0)_+\|_\infty), \tau_0\}, \qquad \tau_c = \tau_s = \max\{\tau_D \max(1, \|\nabla f(x)\|_\infty), \tau_0\}$$

for some absolute tolerance $\tau_0 \ge 0$. An important difference between these tests and (3.1) is that there is no explicit test of the complementarity error in (3.1). Another difference is that with (3.1) it is readily apparent when the solution is not sufficiently accurate because in these cases one of the measures in Table 3.1 is above the required tolerance (but less than one). On the other hand, with tests that are not scale invariant, large or small values for an optimality measure may not reflect the accuracy of the approximate solution returned by the solver.

There are other convergence tests in the literature, but they invariably require multiplier estimates. The GAMS Examiner [7], for example, checks the approximate solution returned by solvers in the GAMS modeling language. The test implemented requires an approximate solution and multiplier estimates. Moreover, the primal and dual feasibility tests use absolute error measures that are not scale invariant.

5. Scale invariance. We now examine the invariance properties of convergence tests when the general optimization problem (1.1) is transformed into an equivalent optimization problem

$$\min \left\{ \hat{f}(x) : \hat{l} \le \hat{c}(x) \le \hat{u} \right\} . \tag{5.1}$$

Scale invariance is a desirable attribute of a convergence test because the choice of tolerances can be made on the basis of the desired accuracy.

We explore scale invariance under transformations (for example, $x \mapsto Sx$, where S is a nonsingular diagonal matrix) that change the units of the problem since, in our experience, most users want the ability to choose the units in the problem formulation and yet retain the same behavior in the optimization algorithm. If the optimization algorithm is scale invariant under the transformation $x \mapsto Sx$ in (1.1), then the iterates $\{x_k\}$ for the optimization problem (1.1) will undergo the same transformation, that is, $x_k = S\hat{x}_k$, where $\{\hat{x}_k\}$ are the iterates for the optimization problem (5.1). Invariance under other transformations (for example, general affine transformations) has been studied in the literature [4].

In the remainder of this section we restrict the discussion to the scale invariance of convergence tests. We first consider the change of scale $\hat{f} = \alpha f$, $\hat{c} = \beta c$, where f is scaled by $\alpha > 0$, and the constraints c are scaled by $\beta > 0$. With this change of scale we must also scale the bounds l and u in the optimization problem (1.1) by β so that $\hat{l} = \beta l$ and $\hat{u} = \beta u$. The optimization problems (1.1) and (5.1) are equivalent under this change of scale in the sense that they have the same solutions.

We also consider the change of scale $x \mapsto Sx$ in (1.1), where S is a nonsingular diagonal matrix. In this case we have $\hat{f}(x) = f(Sx)$ and $\hat{c}(x) = c(Sx)$ in (5.1). With this change of scale any minimizer x^* of the optimization problem (1.1) generates a minimizer $\hat{x}^*$ of (5.1) via $x^* = S\hat{x}^*$. The converse of this statement also holds. Thus, both optimization problems (1.1) and (5.1) have the same solution sets.

We explore the scale invariance of convergence tests under the assumption that all absolute tolerances are set to zero. For the optimality measures in Table 3.1, this means that $\tau_a = 0$ in the definition (2.3) of $\delta[\cdot,\cdot]$. Under this assumption,

$$d[Sx, Sy] = d[x, y] \tag{5.2}$$

for all nonsingular diagonal matrices S and vectors x and y. If $\tau_a > 0$, then (5.2) holds if

$$|s_k| \left(|x_k| + |y_k| \right) \ge \tau_a, \qquad 1 \le k \le n.$$

Hence, scale invariance of $d[\cdot,\cdot]$ holds if the scaled variables for at least one of the vectors is above the absolute tolerance level, that is, $|s_k||x_k| \ge \tau_a$ or $|s_k||y_k| \ge \tau_a$.

The change of scale, $\hat{f} = \alpha f$, $\hat{c} = \beta c$, in f and c implies that $\hat{\lambda}(x) = (\alpha/\beta)\lambda(x)$ for the multiplier defined by (2.10), and hence

$$\nabla \hat{f}(x) = \alpha \nabla f(x), \qquad \nabla \hat{c}(x)\hat{\lambda}(x) = \alpha \nabla c(x)\lambda(x).$$

Thus, (5.2) shows that both $\nu_f(x)$ and $\nu_s(x)$ are invariant under this change of scale.

The scale-invariance properties (5.2) of $d[\cdot,\cdot]$ also show that the set of τ-active constraints is invariant under the change of scale $x \mapsto Sx$, where S is a nonsingular diagonal matrix. On the other hand, the multipliers $\lambda(x)$ defined by (2.10) are not invariant under this change of scale because

$$\nabla \hat{f}(x) = S\nabla f(x), \qquad \nabla \hat{c}(x) = S\nabla c(x).$$

However, if we modify the norm in (2.10) and consider

$$\min\left\{\|D^{-1}y\|_\infty : y = \nabla f(x) - \nabla c(x)v,\ v \in S_\tau(x)\right\},$$

where D is determined from the problem data, then we can have scale invariance. For example, if

$$d_k = \max\left\{|\partial_k f(x_0)|, \|\partial_k c(x_0)\|\right\},$$

then $\lambda(x)$ is invariant under this change of scale. If we take into account (5.2), then we have shown that with this modification all the optimality measures in Table 3.1 are invariant.

An important observation is that the multipliers $\lambda(x)$ defined by (2.10) are scale invariant under the change of scale $x \mapsto Sx$ at any x such that $\nabla f(x) = \nabla c(x)v$ for some $v \in S_\tau(x)$. This holds because in this case $\lambda(x,\tau)$ satisfies

$$\nabla f(x) = \nabla c(x)\lambda(x,\tau),$$

and thus $\lambda(x,\tau)$ is unchanged by the change of scale $x \mapsto Sx$. Thus, we are at least guaranteed scale invariance at KKT points.

An analysis of the scale-invariance properties of the convergence tests (4.1) and (4.2) requires that we specify how the multiplier estimate $\pi(x)$ depends on the change of scale. If we consider the change of scale where $\hat{f} = \alpha f$ and $\hat{c} = \beta c$, and assume that $\hat{\pi}(x) = (\alpha/\beta)\pi(x)$ under this change of scale, then (4.2) shows that this convergence test is scale invariant if the tolerances are scaled by the appropriate problem data. On the other hand, if we consider the change of scale $x \mapsto Sx$, where S is a nonsingular diagonal matrix, then the convergence test (4.2) is not generally scale invariant unless (4.2) is modified to use a norm scaled by the problem data.

The scale-invariance properties of the convergence test (4.1) are shared by (4.6), but this is not the case for (4.5). Indeed, if we consider the change of scale where $\hat{f} = \alpha f$ and $\hat{c} = \beta c$, and assume that $\hat{\pi}(x) = (\alpha/\beta)\pi(x)$, then $\hat{c}(x)\hat{\pi}(x) = \alpha c(x)\pi(x)$. Hence, (4.5) shows that τ_c must be scaled by both α/β and α. Since τ_c cannot absorb two different changes of scale, the convergence test (4.5) is not scale invariant.

6. Benchmarking with COPS.

We use performance profiles [5] and COPS [6] (Version 3.0) to evaluate the effect of the optimality measures in Table 3.1. The COPS benchmark collection provides a selection of difficult nonlinearly constrained optimization problems from applications in optimal design, fluid dynamics, parameter

TABLE 6.1
Problem data (minimum, quartiles, maximum) for the COPS *benchmark.*

Problem parameter	min	q_1	q_2	q_3	max
Variables	98	1146	2500	4398	19241
Equality constraints	0	0	100	1995	7995
General constraints	0	0	0	41	20093
Total constraints	21	1177	2402	4001	20496

estimation, mesh smoothing, computational chemistry, and optimal control, among others. Moreover, each application has a short description of the formulation of the application as an optimization problem.

Version 3.0 of COPS has 22 different applications. For each of these applications we use three to five instances of the application obtained by varying a parameter in the application, for example, the number of grid points in a discretization. Table 6.1 gives the quartiles for four problem parameters: the number of variables, the number of equality and general inequality constraints (excluding bounds), and the total number of constraints (including bounds).

Our benchmarking results are done with a set of problems and solvers. We have used COPS, but we could have also used a selection of the engineering problems provided by Vanderbei [16]. Also note that timing data refers to a particular computing environment (machine, compiler, libraries). Hence, our conclusions could change if the problems, solvers, or computing environment changes. On the other hand, the use of performance profiles tends to minimize the effect of these issues, as noted in [5].

We also note that the solvers for constrained optimization problems invariably have different requirements. Some of the solvers use second-order information, while others (for example, MINOS and SNOPT) use only first-order information. The use of second-order information can reduce the number of iterations, but the cost per iteration usually increases. In addition, obtaining second-order information is more costly and may not even be possible. Memory requirements can also play an important role. In particular, solvers that use direct linear equation solvers are often more efficient in terms of computing time, provided there is enough memory. Moreover, some of the solvers are designed for problems with a modest number of degrees of freedom.

The script for generating the timing data sends a problem to each solver successively, so as to minimize the effect of fluctuation in the machine load. The script tracks the wall-clock time from the start of the AMPL process to the end of the solve. Any process that runs 30 minutes is declared unsuccessful. We cycle through all the problems, recording the wall-clock time as well as the combination of AMPL system time (to interpret the model and compute varying amounts of derivative information required by each solver) and solver time for each model variation. We have verified that the AMPL time results we present can be reproduced to within 10 percent accuracy.

7. Computational experiments.

We now investigate how performance profiles behave when the convergence test (3.1) is enforced on all the solvers. We also describe some of the computational experiments that we have done with an analyzer that computes the optimality measures in Table 3.1 for optimization problems in the AMPL or GAMS modeling language.

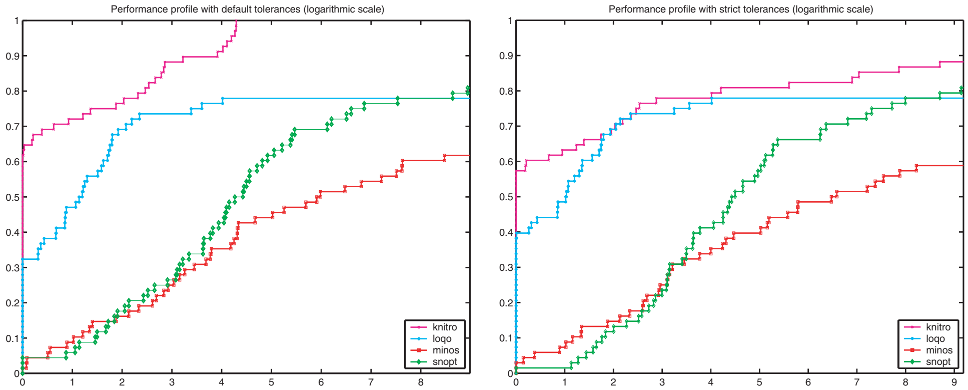


FIG. 7.1. *Performance profiles* ($\log_2$ *scale) when no optimality checks are made (left) and when the convergence test* (7.1) *is enforced (right).*

Figure 7.1 displays performance profiles for the two experiments performed with the COPS 3.0 test set [6]. The following solvers were used for the experiments:

KNITRO 3.0 [18], ASL (20020905)	LOQO 6.02 [17], ASL (20020221)
MINOS 5.5 [15], ASL (20020614)	SNOPT 6.1 [8], ASL (20020614)

All the computations were performed on an Intel Pentium 4 1.8 GHz CPU with 512 MB of RAM and a 256 KB cache, running Red Hat Linux 7.3. Furthermore, a time limit of 30 minutes was imposed on the solvers for each problem in the test set. A failure is reported when the time limit expires.

The first experiment does not check the optimality measures, trusting the optimization solver to report optimality. In the second experiment, if the convergence test proposed in section 3,

$$\nu_f(x) \le \tau_f, \quad \nu_s(x, \tau_f) \le \tau_s, \qquad \tau_f = \tau_s = 10^{-6}, \tag{7.1}$$

is not satisfied, then the convergence tolerances used by the solvers are reduced and the problem is solved again. This procedure is stopped when the tolerances provided to the solver reach 10^{-16}. The method used to reduce convergence tolerances depends on the solver. All solvers that we tested have feasibility and optimality tolerances that can be set by the user. However, the names and meanings of the parameters change with the solver.

We expect that a solver will take longer to compute a solution when the termination tolerances are reduced. However, if the behavior of the solver depends on the convergence tolerances, then the solver can take less time to compute a more accurate solution. In particular, the iterates explored by the solver can change dramatically when the internal tolerances are modified. The performance profiles use the time taken to solve the given model with the tightened tolerances.

The analyzer computes the optimality measures in Table 3.1 for optimization problems in the AMPL or GAMS modeling language. The main computational task in determining the optimality measures is to compute multipliers by setting up and solving the linear program (2.10) with MINOS. All the data for (2.10) are written to a file with at least 15 digits of accuracy. The optimality and feasibility tolerances for MINOS are set to 10^{-14} when computing the multipliers. In all tests, MINOS indicates that an optimal solution to the linear program was found.

The solvers reported optimal solutions that satisfied the convergence test (7.1) on 109 (40%) of the 272 problem instances. In two cases, one of the solvers reported that a nonoptimal solution had been found, when in fact our optimality measures were within the desired tolerances.

The major conclusion that can be drawn from Figure 7.1 is that performance profiles do indeed change when a consistent convergence test is used. The trends in the plots remain the same, but the magnitude of the differences, especially at the beginning of the plots, can change significantly.

The effect of convergence criteria on solver performance can also be seen in Figure 7.2. In this figure we plot the performance metric

$$p(x) = -\log_{10}(\max(\nu_f(x), \nu_s(x, \tau_f))) \qquad (7.2)$$

for all the solvers. The white bars indicate the first experiment, where the convergence test (7.1) is not enforced, while the black bars indicate the second experiment, where the convergence test is enforced.

The heights of the bars in Figure 7.2 give the levels of accuracy reached. The definition of the measures ν_f and ν_s shows that we can expect to have the performance metric $p(x) \in [0, 16]$ on computations with 16 decimal digits. Problems where $p(x)$ is near zero have not been solved accurately. If $p(x) < 6$, then the convergence test (7.1) is not satisfied.

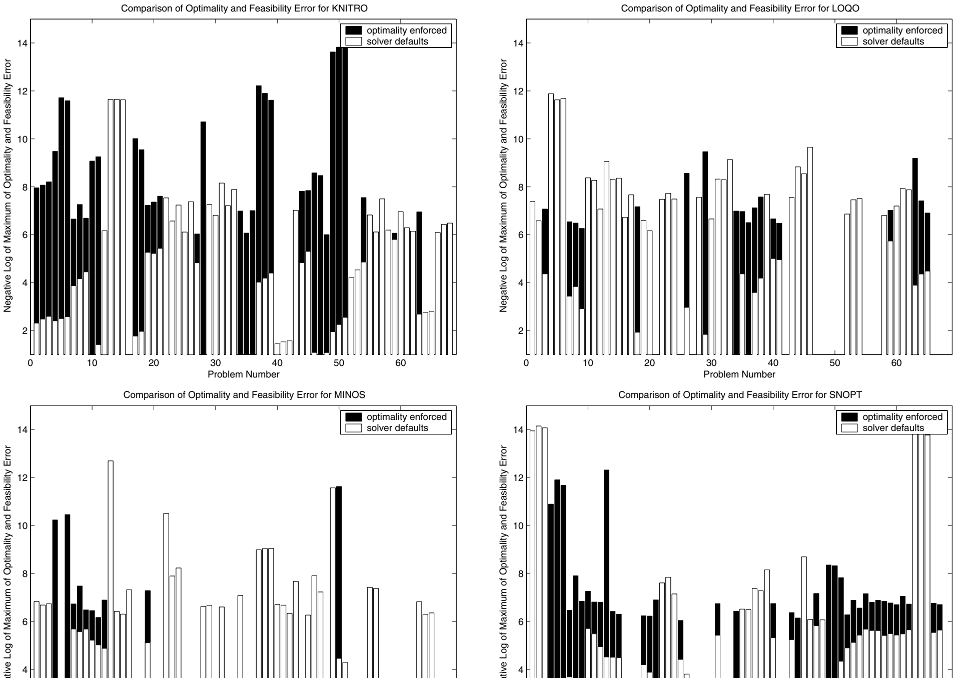


FIG. 7.2. *Graph of the performance metric* (7.2) *for* KNITRO *(upper left),* LOQO *(upper right),* MINOS *(lower left), and* SNOPT *(lower right) when no optimality checks are made (white) and when the convergence test* (7.1) *is enforced (black).*

If only a white bar is shown for a problem, the solver either satisfied the convergence test with the default tolerances, and no refinement was needed, or reported a failure. If the solver reports a failure, then the white bar will not reach a tolerance of $\tau_f = \tau_s = 10^{-6}$. In these cases, the iterative reduction in the tolerances was stopped. Those models with no bar either encountered the time limit imposed during the testing or reported a failure.

A black bar in Figure 7.2 indicates that for the particular problem the default tolerances had to be reduced to meet the convergence test (7.1). As can be seen, all solvers failed to satisfy (7.1) initially in several cases but managed to satisfy the convergence test as the tolerances were reduced.

Figure 7.2 clearly shows that the default convergence test for MINOS tends to agree with (7.1) in most cases. This code was able to satisfy the convergence test with the default tolerances for most of the problems. On the other hand, these results show that KNITRO tended to perform poorly when measured with the metric (7.2).

The main reason why solvers fail to satisfy (7.1) with their default tolerances is that, as noted at the end of section 3, solvers tend to scale τ_f and τ_s based on the size of problem data, for example, x, the multipliers $\pi(x)$, or the constraints $c(x)$. This scaling can increase the values of τ_f and τ_s, and thus lead to a weaker convergence test. Assume, for example, that $\tau_P = \tau_D = 10^{-6}$ in the convergence tests of SNOPT and KNITRO. If $\|x\| = 10^4$ and $\|c(x_0)_+\|_\infty = 10^4$, then both solvers set $\tau_f = 10^{-2}$. This explains why some of the white bars are below the 10^{-3} level.

Another reason for the large values of ν_f and ν_s obtained by the solvers with their default tolerances is that ν_f and ν_s examine the accuracy in all of the components, while other measures examine the accuracy in the largest components. Consider, for example, a case where

$$\nabla f(x) = \begin{pmatrix} \alpha \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \qquad \nabla c(x)\pi(x) = \begin{pmatrix} \alpha(1+\tau_s) \\ 1+\alpha\tau_s \\ \vdots \\ 1+\alpha\tau_s \end{pmatrix}$$

for some $\alpha \geq 1$. In this case,

$$\|\nabla f(x) - \nabla c(x)\pi(x)\|_\infty \leq \tau_s \min\Big\{ \|\nabla f(x)\|_\infty, \|\nabla c(x)\pi(x)\|_\infty \Big\},$$

and thus the relative error between $\nabla f(x)$ and $\nabla c(x)\pi(x)$ is small. However, it is also clear that the relative error between components $2, \ldots, n$ can be large. In fact,

$$\nu_s(x, \tau_f) = \frac{\alpha\tau_s}{2+\alpha\tau_s},$$

and thus the error measured by ν_s can be arbitrarily close to one.

8. Concluding remarks. We have shown that the convergence test (3.1) is scale invariant when absolute tolerances are set to zero and behaves satisfactorily at any point where the constraints satisfy the Mangasarian–Fromovitz constraint qualification. We have also demonstrated that this test does not need to use the multipliers given by (2.10), but can use the projection of any set of multipliers into the cone (2.7) associated with $\mathcal{A}_\tau(x)$. This approach avoids an explicit test on complementarity.

Our computational experiments have shown that the use of this convergence test on the benchmarking process can have a significant effect on performance profiles.

These experiments have also shown that an additional advantage of the measures associated with this convergence test is that solutions of low accuracy will have either ν_f or ν_s close to one, while high accuracy solutions will satisfy (7.1) for the appropriate values of τ_f and τ_s. This is clearly seen in Figure 7.2.

Our strategy of decreasing the solver tolerances until a uniform convergence test is satisfied or a time limit is exceeded provides a benchmarking process that forces all solvers to satisfy the same convergence test. Our benchmarking results used a particular set of tolerance values ($\tau_f = \tau_s = 10^{-6}$) and did not address the issue of how results change if these values are changed. We chose the tolerance values in (7.1) to reflect our views of reasonable tolerances. We expect the results to remain basically unchanged if τ_f and τ_s are chosen in the interval $(10^{-7}, 10^{-5})$, but at present this is a conjecture. We do expect optimization solvers to behave unpredictably if tolerances are chosen too small or too large, but we have not benchmarked solvers in these situations.

Our computational results in section 7 noted that computing times generally increased as tolerances were reduced, but that this was not always the case. The increase in computing time is guaranteed if the optimization solver uses only the tolerances to determine when to stop the iteration process and return an approximate solution. However, if the optimization solver uses the input tolerances to dictate how the solver behaves in certain situations, then the computing time could change significantly if the tolerances are changed.

If computing times for a solver change significantly as tolerances are changed, then the results of a benchmarking study that uses computing time to measure performance could be claimed to be invalid unless an attempt is made to incorporate this variability in computing times into the conclusions. We do not agree with this claim. Instead, we repeat the cautions mentioned in section 6 that computing time in a benchmarking study refers to a particular computing environment (machine, compiler, libraries) and that the conclusions of the study may change if the problems, solvers, or computing environment changes.

Acknowledgments. We are grateful for the careful reading and suggestions of the referees and associate editor, Margaret Wright. Their attention to detail is noteworthy and led to clarifications on various aspects of this work.

REFERENCES

[1] H. Y. Benson, D. F. Shanno, and R. J. Vanderbei, *A comparative study of large-scale nonlinear optimization algorithms*, in High Performance Algorithms and Software for Nonlinear Optimization, G. Di Pillo and A. Murli, eds., Kluwer Academic, Dordrecht, The Netherlands, 2003, pp. 95–128.

[2] E. G. Birgin, R. A. Castillo, and J. M. Martínez, *Numerical Comparison of Augmented Lagrangian Algorithms for Nonconvex Problems*, preprint, University of São Paulo, 2003.

[3] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust Region Methods*, MPS-SIAM Ser. Optim., SIAM, Philadelphia, 2000.

[4] P. Deuflhard, *Newton Methods for Nonlinear Problems*, Springer Ser. Comput. Math. 35, Springer, New York, 2004.

[5] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Programming, 91 (2002), pp. 201–213.

[6] E. D. Dolan, J. J. Moré, and T. S. Munson, *Benchmarking Optimization Software with COPS* 3.0, Technical Memorandum ANL/MCS-TM-273, Argonne National Laboratory, Argonne, IL, 2004.

[7] *General Algebraic Modeling System (GAMS) Solvers*, www.gams.com/solvers.

[8] P. E. Gill, W. Murray, and M. A. Saunders, *User's Guide for SNOPT* 5.3*: A Fortran Package for Large-Scale Nonlinear Programming*, Report NA97-5, University of California, San Diego, CA, 1997.

[9] P. E. Gill, W. Murray, and M. A. Saunders, *SNOPT: An SQP algorithm for large-scale constrained optimization*, SIAM J. Optim., 12 (2002), pp. 979–1006.

[10] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, New York, 1981.

[11] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.

[12] C.-J. Lin and J. J. Moré, *Newton's method for large bound-constrained optimization problems*, SIAM J. Optim., 9 (1999), pp. 1100–1127.

[13] H. D. Mittelmann and A. Pruessner, *A Server for Automated Performance Analysis of Benchmarking Data*, preprint, Performance World, 2003. Available online at www.gamsworld.org/performance.

[14] J. L. Morales, J. Nocedal, R. A. Waltz, and H. Liu, *Assessing the potential of interior point methods for nonlinear optimization*, in High Performance Algorithms and Software for Nonlinear Optimization, L. T. Biegler, O. Ghattas, M. Heinkenschloss, and B. Van Bloemen Waanders, eds., Springer-Verlag, New York, 2003, pp. 167–183.

[15] B. A. Murtagh and M. A. Saunders, *MINOS* 5.5 *User's Guide*, Report SOL 83-20R, Stanford University, Stanford, CA, 1983; revised 1998.

[16] R. J. Vanderbei, *Nonlinear Optimization Models*, www.sor.princeton.edu/~rvdb/ampl/nlmodels.

[17] R. J. Vanderbei, *LOQO User's Manual—Version* 4.05, 2000.

[18] R. Waltz and J. Nocedal, *KNITRO User's Manual—Version* 3.1, Tech. Report 5, Optimization Technology Center, Northwestern University, Evanston, IL, 2003. Available online at http://www.princeton.edu/~rvdb/tex/loqo/loqo405.pdf.

SIAM J. Optim.
Vol. 16, No. 3, pp. 910–937

GLOBAL SEARCH BASED ON EFFICIENT DIAGONAL PARTITIONS AND A SET OF LIPSCHITZ CONSTANTS[*]

YAROSLAV D. SERGEYEV[†] AND DMITRI E. KVASOV[‡]

Abstract. In the paper, the global optimization problem of a multidimensional "black-box" function satisfying the Lipschitz condition over a hyperinterval with an unknown Lipschitz constant is considered. A new efficient algorithm for solving this problem is presented. At each iteration of the method a number of possible Lipschitz constants are chosen from a set of values varying from zero to infinity. This idea is unified with an efficient diagonal partition strategy. A novel technique balancing usage of local and global information during partitioning is proposed. A new procedure for finding lower bounds of the objective function over hyperintervals is also considered. It is demonstrated by extensive numerical experiments performed on more than 1600 multidimensional test functions that the new algorithm shows a very promising performance.

Key words. global optimization, black-box functions, derivative-free methods, partition strategies, diagonal approach

AMS subject classifications. 65K05, 90C26, 90C56

DOI. 10.1137/040621132

1. Introduction.

Many decision-making problems arising in various fields of human activity (technological processes, economic models, etc.) can be stated as global optimization problems (see, e.g., [7, 26, 33]). Objective functions describing real-life applications are very often multiextremal, nondifferentiable, and hard to evaluate. Numerical techniques for finding solutions to such problems have been widely discussed in the literature (see, e.g., [5, 14, 15, 26, 33]).

In this paper, the Lipschitz global optimization problem is considered. This type of optimization problem is sufficiently general both from theoretical and applied points of view. In fact, it is based on a rather natural assumption that any limited change in the parameters of the objective function yields some limited changes in the characteristics of the object's performance. The knowledge of a bound on the rate of change of the objective function, expressed by the Lipschitz constant, allows one to construct global optimization algorithms and to prove their convergence (see, e.g., [14, 15, 26, 33]).

Mathematically, the global optimization problem considered in the paper can be formulated as minimization of a multidimensional multiextremal "black-box" function that satisfies the Lipschitz condition over a domain $D \subset \mathbb{R}^N$ with an unknown constant L, i.e., finding the value f^* and points x^* such that

$$f^* = f(x^*) = \min_{x \in D} f(x), \tag{1.1}$$

$$|f(x') - f(x'')| \le L\|x' - x''\|, \quad x', x'' \in D, \quad 0 < L < \infty, \tag{1.2}$$

[*] Received by the editors December 20, 2004; accepted for publication (in revised form) July 28, 2005; published electronically January 26, 2006. This work was partially supported by the following grants: FIRB RBAU01JYPN, FIRB RBNE01WBBB, and RFBR 04-01-00455-a.
http://www.siam.org/journals/siopt/16-3/62113.html

[†] Dipartimento di Elettronica, Informatica e Sistemistica, Università della Calabria, Via P.Bucci, Cubo 41C – 87036 Rende (CS), Italy, and N.I. Lobachevski University of Nizhni Novgorod, Russia (yaro@si.deis.unical.it).

[‡] Dipartimento di Statistica, Università di Roma "La Sapienza," P.le A. Moro 5 – 00185 Roma, Italy, and N.I. Lobachevski University of Nizhni Novgorod, Russia (kvadim@si.deis.unical.it).

where

$$D = [a, b\,] = \{x \in \mathbb{R}^N : a(j) \leq x(j) \leq b(j),\, 1 \leq j \leq N\}, \tag{1.3}$$

a, b are given vectors in $\mathbb{R}^N$, and $\| \cdot \|$ denotes the Euclidean norm.

The function $f(x)$ is supposed to be nondifferentiable. Hence, optimization methods using derivatives cannot be used for solving problem (1.1)–(1.3). It is also assumed that evaluation of the objective function at a point (also referred to as a function trial) is a time-consuming operation.

Numerous algorithms have been proposed (see, e.g., [5, 14, 15, 18, 21, 23, 26, 27, 29, 32, 33]) for solving problem (1.1)–(1.3). One of the main questions to be considered in this occasion is, How can the Lipschitz constant L be specified? There are several approaches to specify the Lipschitz constant. First of all, it can be given a priori (see, e.g., [14, 15, 23, 27, 32]). This case is very important from the theoretical viewpoint but is not frequently encountered in practice. The more promising and practical approaches are based on an adaptive estimation of L in the course of the search. In such a way, algorithms can use either a global estimate of the Lipschitz constant (see, e.g., [21, 26, 33]) valid for the whole region D from (1.3), or local estimates L_i valid only for some subregions $D_i \subseteq D$ (see, e.g., [20, 24, 29, 30, 33]).

Since the Lipschitz constant has a significant influence on the convergence speed of the Lipschitz global optimization algorithms, the problem of its specifying is of the great importance. In fact, accepting too high a value of L for a concrete objective function means assuming that the function has complicated structure with sharp peaks and narrow attraction regions of minimizers within the whole admissible region. Thus, too high a value of L (if it does not correspond to the real behavior of the objective function) leads to a slow convergence of the algorithm to the global minimizer.

Global optimization algorithms using in their work a global estimate of L (or some value of L given a priori) do not take into account local information about behavior of the objective function over every small subregion of D. As has been demonstrated in [29, 30, 33], estimating local Lipschitz constants allows us to significantly accelerate the global search. Naturally, balancing between local and global information must be performed in an appropriate way to avoid the missing of the global solution.

Recently, an interesting approach unifying usage of local and global information during the global search has been proposed in [18]. At each iteration of this new algorithm, called DIRECT, instead of only one estimate of the Lipschitz constant a set of possible values of L is used.

Like many Lipschitz global optimization algorithms, DIRECT tries to find the global minimizer by partitioning the search hyperinterval D into smaller hyperintervals D_i using a particular partition scheme described in [18]. The objective function is evaluated only at the central point of a hyperinterval. Each hyperinterval D_i of a current partition of D is characterized by a lower bound of the objective function over this hyperinterval. It is calculated similarly to [27, 32] taking into account the Lipschitz condition (1.2). A hyperinterval D_i is selected for a further partitioning if and only if for some value $\tilde{L} > 0$ (which estimates the unknown constant L) it has the smallest lower bound of $f(x)$ with respect to the other hyperintervals. By changing $\tilde{L}$ from zero to infinity, at each iteration DIRECT selects several "potentially optimal" hyperintervals (see [10, 13, 18]) in such a way that for a particular estimate of the Lipschitz constant the objective function could have the smallest lower bound over every potentially optimal hyperinterval.

Due to its simplicity and efficiency, DIRECT has been widely adopted in practical applications (see, e.g., [1, 2, 3, 4, 10, 13, 22, 35]). In fact, DIRECT is a derivative-free deterministic algorithm which does not require multiply runs. It has only one parameter, which is easy to set (see [18]). The center-sampling partition strategy of DIRECT reduces the computational complexity in high-dimensional spaces, allowing DIRECT to demonstrate good performance results (see [11, 18]).

However, some aspects which can limit the applications of DIRECT have been pointed out by several authors (see, e.g., [4, 6, 13, 17]). First of all, it is difficult to apply for DIRECT some meaningful stopping criterion, such as, for example, stopping on achieving a desired accuracy in solution. This happens because DIRECT does not use a single estimate of the Lipschitz constant but a set of possible values of L. Although several attempts at introducing a reasonable criterion of arrest have been made (see, e.g., [2, 4, 10, 13]), termination of the search process caused by exhaustion of the available computing resources (such as maximal number of function trials) remains the most interesting for practical engineering applications.

Another important observation regarding DIRECT is related to the partition and sampling strategies adopted by the algorithm (see [18]) which simplicity turns into some problems. As has been outlined in [4, 6, 22], DIRECT is quick to locate regions of local optima but slow to converge to the global one. This can happen for several reasons. The first one is a redundant (especially in high dimensions, see [17]) partition of hyperintervals along all longest sides. The next cause of DIRECT's slow convergence can be excessive partition of many small hyperintervals located in the vicinity of local minimizers which are not global ones. Finally, DIRECT—like all center-sampling partitioning schemes—uses relatively poor information about behavior of the objective function $f(x)$. This information is obtained by evaluating $f(x)$ only at one central point of each hyperinterval without considering the adjacent hyperintervals. Due to this fact, DIRECT can manifest slow convergence (as has been highlighted in [16]) in cases when the global minimizer lies at the boundary of the admissible region D from (1.3).

There are several modifications to the original DIRECT algorithm. For example, in [17], partitioning along only one long side is suggested to accelerate convergence in high dimensions. The problem of stagnation of DIRECT near local minimizers (emphasized, e.g., in [6]) can be attacked by changing the parameter of the algorithm (see [18]) preventing DIRECT from being too local in its orientation (see [6, 10, 13, 17, 18]). But in this case the algorithm becomes too sensitive to tuning such a parameter, especially for difficult black-box global optimization problems (1.1)–(1.3). In [1, 35], another modification to DIRECT, called "aggressive DIRECT," has been proposed. It subdivides all hyperintervals with the smallest function value for each hyperinterval size. This results in more hyperintervals partitioned at every iteration, but the number of hyperintervals to be subdivided grows significantly. In [10, 11], the opposite idea which is more biased toward local improvement of the objective function has been studied. Results obtained in [10, 11] demonstrate that this modification seems to be more suitable for low-dimensional problems with a single global minimizer and a few local minimizers.

The goal of this paper is to present a new algorithm which would be oriented (in contrast with the algorithm from [10, 11]) on solving "difficult" multidimensional multiextremal black-box problems (1.1)–(1.3). It uses a new technique for selection of hyperintervals to be subdivided which is unified with a new diagonal partition strategy. A new procedure for estimation of lower bounds of the objective function over hyperintervals is combined with the idea (introduced in DIRECT) of usage of a set of Lipschitz constants instead of a unique estimate. As demonstrated by extensive numerical results, application of the new algorithm to minimizing hard multidimensional black-box functions leads to significant improvements.

The paper is organized as follows. In section 2, a theoretical background of the new algorithm—a new partition strategy, a new technique for lower bounding of the objective function over hyperintervals, and a procedure for selection of "nondominated" hyperintervals for eventual partitioning—is presented. Section 3 is dedicated to the description of the algorithm and to its convergence analysis. Finally, section 4 contains results of numerical experiments executed on more than 1600 test functions.

2. Theoretical background.

This section consists of the following three parts. First, a new partition strategy developed in the framework of diagonal approach is described. The second part presents a new procedure for estimation of lower bounds of the objective function over hyperintervals. The third part is dedicated to the description of a procedure for determining nondominated hyperintervals—hyperintervals that have the smallest lower bound for some particular estimate of the Lipschitz constant.

2.1. Partition strategy.

In global optimization algorithms, various techniques for adaptive partition of the admissible region D into a set of hyperintervals D_i are used (see, e.g., [14, 18, 26, 31]) for solving (1.1)–(1.3). A current partition $\{D^k\}$ of D in the course of an iteration $k \geq 1$ of an algorithm can be represented as

$$D = \cup_{i=1}^{m(k)+\Delta m(k)} D_i, \quad D_i \cap D_j = \delta(D_i) \cap \delta(D_j), \quad i \neq j. \tag{2.1}$$

Here, $\delta(D_i)$ denotes the boundary of D_i, $m(k)$ is the number of hyperintervals at the beginning of the iteration k, and $\Delta m(k)$ is the current number of new hyperintervals produced during the kth iteration. For example, if only one new hyperinterval is generated at every iteration, then $\Delta m(k) = 1$.

Over each hyperinterval $D_i \in \{D^k\}$, approximation of $f(x)$ is based on results obtained by evaluating $f(x)$ at some points $x \in D$. For example, DIRECT [18] involves partitioning with evaluation of $f(x)$ at the central points of hyperintervals (note that for DIRECT the number $\Delta m(k)$ in (2.1) can be greater than 1).

In this paper, the diagonal approach proposed in [25, 26] is considered. In this approach, the function $f(x)$ is evaluated only at two vertices a_i and b_i of the main diagonals of each hyperinterval D_i independently of the problem dimension (recall that each evaluation of $f(x)$ is a time-consuming operation).

Among attractions of the diagonal approach there are the following two. First, the objective function is evaluated at two points at each hyperinterval. Thus, diagonal algorithms obtain more complete information about the objective function than center-sampling methods. Second, many efficient one-dimensional global optimization algorithms can be easily extended to the multivariate case by means of the diagonal scheme (see, e.g., [20, 21, 24, 25, 26]).

As shown in [21, 31], diagonal global optimization algorithms based on widely used partition strategies (such as bisection or partition 2^N used in [25, 26]) produce many redundant trials of the objective function. This redundancy slows down the algorithm execution because of high time required for the evaluations of $f(x)$. It also increases the computer memory allocated for storing the redundant information.

The new partition strategy proposed in [31] (see also [21]) overcomes these drawbacks of conventional diagonal partition strategies. We start its description by a two-dimensional example in Figure 1. In this figure, partitions of the admissible region D produced by the algorithm at the initial iterations are presented. We suppose

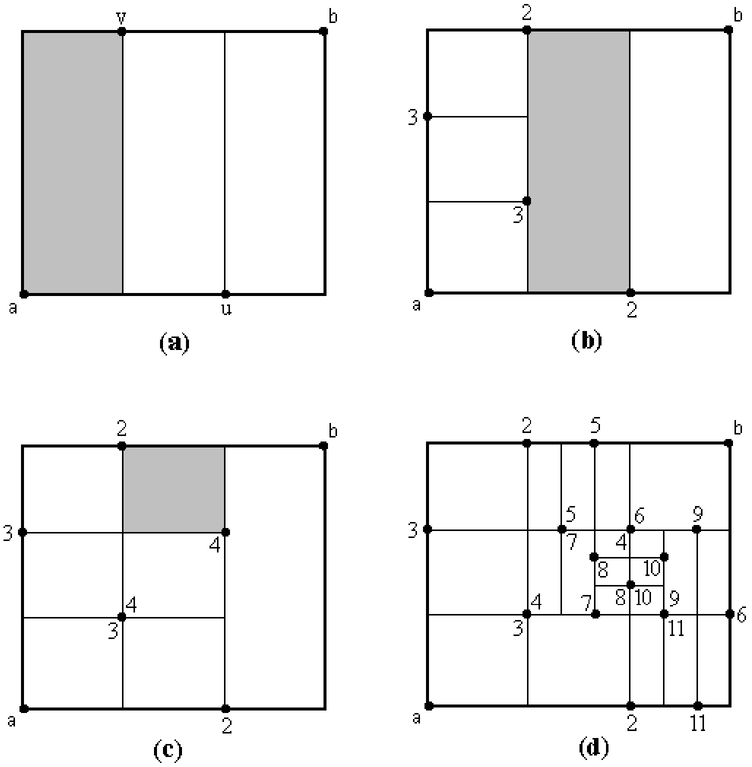


FIG. 1. *An example of subdivisions by a new partition strategy.*

just for simplicity that at each iteration only one hyperinterval can be subdivided. Trial points of $f(x)$ are represented by black dots. The numbers around these dots indicate iterations at which the objective function is evaluated at the corresponding points. The terms "interval" and "subinterval" will be used to denote two-dimensional rectangular domains.

In Figure 1(a) the situation after the first two iterations is presented. At the first iteration, the objective function $f(x)$ is evaluated at the vertices a and b of the search domain $D = [a, b]$. At the next iteration, the interval D is subdivided into three subintervals of equal area (equal volume in general case). This subdivision is performed by two lines (hyperplanes) orthogonal to the longest edge of D and passing through points u and v (see Figure 1(a)). The objective function is evaluated at both points u and v.

Suppose that the interval shown in grey in Figure 1(a) is chosen for the further partitioning. Thus, at the third iteration, three smaller subintervals appear (see Figure 1(b)). It seems that a trial point of the third iteration is redundant for the interval (shown in grey in Figure 1(b)) selected for the next splitting. But in reality, Figure 1(c) demonstrates that one of the two points of the fourth iteration coincides with the point 3 at which $f(x)$ has already been evaluated. Therefore, there is no need to evaluate $f(x)$ at this point again, since the function value obtained at the previous iteration can be used. This value can be stored in a special vertex database and is simply retrieved when it is necessary without reevaluation of the function. For example, Figure 1(d) illustrates the situation after 11 iterations. Among 22 points at which the objective function is to be evaluated, there are 5 repeated points. That is, $f(x)$ is evaluated 17 rather than 22 times. Note also that the number of generated intervals (equal to 21) is greater than the number of trial points (equal to 17). Such a difference becomes more pronounced in the course of further subdivisions (see [21]).

Let us now describe the general procedure of hyperinterval subdivision. Without loss of generality, hereafter we assume that the admissible region D in (1.3) is an N-dimensional hypercube. Suppose that at the beginning of an iteration $k \geq 1$ of the algorithm the current partition $\{D^k\}$ of $D = [a, b]$ consists of $m(k)$ hyperintervals and $\Delta m(k) \geq 0$ new hyperintervals have been already obtained. Let a hyperinterval $D_t = [a_t, b_t]$ be chosen for partitioning too. The operation of partitioning the selected hyperinterval D_t is performed as follows (we omit the iteration index in the formulae).

Step 1. Determine points u and v by the following formulae:

$$u = \left(a(1), \ldots, a(i-1), a(i) + \frac{2}{3}(b(i) - a(i)), a(i+1), \ldots, a(N)\right), \tag{2.2}$$

$$v = \left(b(1), \ldots, b(i-1), b(i) + \frac{2}{3}(a(i) - b(i)), b(i+1), \ldots, b(N)\right), \tag{2.3}$$

where $a(j) = a_t(j),\ b(j) = b_t(j),\ 1 \leq j \leq N$, and i is given by the equation

$$i = \arg\min \max_{1 \leq j \leq N} |b(j) - a(j)|. \tag{2.4}$$

Get (evaluate or read from the vertex database) the values of the objective function $f(x)$ at the points u and v.

Step 2. Divide the hyperinterval D_t into three hyperintervals of equal volume by two parallel hyperplanes that are perpendicular to the longest edge i of D_t and pass through the points u and v.

The hyperinterval D_t is thus substituted by three new hyperintervals with indices $t' = t$, $m + \Delta m + 1$, and $m + \Delta m + 2$ determined by the vertices of their main diagonals

$$a_{t'} = a_{m+\Delta m+2} = u, \qquad b_{t'} = b_{m+\Delta m+1} = v, \tag{2.5}$$

$$a_{m+\Delta m+1} = a_t, \qquad b_{m+\Delta m+1} = v, \tag{2.6}$$

$$a_{m+\Delta m+2} = u, \qquad b_{m+\Delta m+2} = b_t. \tag{2.7}$$

Step 3. Augment the number of hyperintervals generated during the iteration k:

$$\Delta m = \Delta m(k) := \Delta m(k) + 2. \tag{2.8}$$

The existence of a special indexation of hyperintervals establishing links between hyperintervals generated at different iterations has been theoretically demonstrated in [31]. It allows one to store information about vertices and the corresponding values of $f(x)$ in a special database, thereby avoiding redundant evaluations of $f(x)$. The objective function value at a vertex is calculated only once, stored in the database, and read when required. The new partition strategy generates trial points in such a way that one vertex where $f(x)$ is evaluated can belong to several (up to 2^N)

hyperintervals (see, for example, a trial point at the 8th iteration in Figure 1(d)). Since the time-consuming operation of evaluation of the function is replaced by a significantly faster operation of reading (up to 2^N times) the function values from the database, the new partition strategy considerably speeds up the search and also leads to saving computer memory. It is particularly important that the advantage of the new scheme increases with the growth of the problem dimension (see [21, 31]).

The new strategy can be viewed also as a procedure generating a series of curves similar to space-filling curves—adaptive diagonal curves. Each of these curves is constructed on the main diagonals of hyperintervals obtained during subdivision of D. The objective function is approximated over the multidimensional region D by evaluating $f(x)$ at the points of one-dimensional adaptive diagonal curve. The order of partition of this curve is different within different subintervals of D. If selection of hyperintervals for partitioning is realized appropriately in an algorithm, the curve condenses in the vicinity of the global minimizers of $f(x)$ (see [21, 31]).

2.2. Lower bounds. Let us suppose that at some iteration $k > 1$ of the global optimization algorithm the admissible region D has been partitioned into hyperintervals $D_i \in \{D^k\}$ defined by their main diagonals $[a_i, b_i]$. At least one of these hyperintervals should be selected for further partitioning. In order to make this selection, the algorithm estimates the goodness (or, in other words, characteristics) of the generated hyperintervals with respect to the global search. The best (in some predefined sense) characteristic obtained over some hyperinterval D_t corresponds to a higher possibility to find the global minimizer within D_t. This hyperinterval is subdivided at the next iteration of the algorithm. Naturally, more than one "promising" hyperinterval can be partitioned at every iteration.

One of the possible characteristics of a hyperinterval can be an estimate of the lower bound of $f(x)$ over this hyperinterval. Once all lower bounds for all hyperintervals of the current partition $\{D^k\}$ have been calculated, the hyperinterval with the smallest lower bound can be selected for the further partitioning.

Different approaches to finding lower bounds of $f(x)$ have been proposed in the literature (see, e.g., [14, 18, 23, 26, 27, 32, 33]) for solving problem (1.1)–(1.3). For example, given the Lipschitz constant L, in [14, 23, 27] a minorant function for $f(x)$ is constructed as the upper envelope of a set of N-dimensional circular cones of the slope L. Trial points of $f(x)$ are coordinates of the vertices of the cones. At each iteration, the global minimizer of the minorant function is determined and chosen as a new trial point. Finding such a point requires analyzing the intersections of all cones and, generally, is a difficult and time-consuming task, especially in high dimensions.

If a partition of D into hyperintervals is used, each cone can be considered over the corresponding hyperinterval, independently from the other cones. This allows one (see, e.g., [14, 18, 20, 26]) to avoid the necessity of establishing the intersections of the cones and to simplify the lower bound estimation. For example, the multidimensional DIRECT algorithm [18] uses one cone with symmetry axis passed through a central point of a hyperinterval for lower bounding $f(x)$ over this hyperinterval. The lower bound is obtained on the boundary of the hyperinterval. This approach is simple, but it gives too rough an estimate of the minimum function value over the hyperinterval.

The more accurate estimate is achieved when two trial points over a hyperinterval are used for constructing a minorant function for $f(x)$. These points can be, for example, the vertices a_i and b_i of the main diagonal of a hyperinterval $D_i \in \{D^k\}$ (see, e.g., [20, 21, 25, 26, 31]). In this case, the objective function (due to the Lipschitz condition (1.2)) must lie above the intersection of the N-dimensional cones $C_1(x, L)$

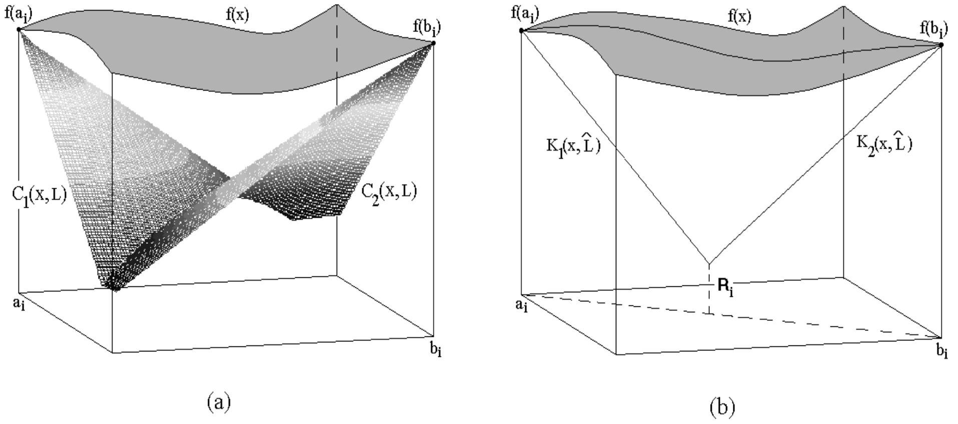


FIG. 2. *Estimation of the lower bound of $f(x)$ over an interval $D_i = [a_i, b_i]$.*

and $C_2(x, L)$ (see the two-dimensional example in Figure 2(a)). These cones have the slope L and are limited by the boundaries of the hyperinterval D_i. The vertices of the cones (in $(N + 1)$-dimensional space) are defined by the coordinates $(a_i, f(a_i))$ and $(b_i, f(b_i))$, respectively. In such a way, the lower bound of $f(x)$ is more precise with respect to the center-sampling strategy. Algorithms using this approach are called diagonal (see, e.g., [20, 21, 25, 26, 31]).

In the new diagonal algorithm proposed in this paper, the objective function is also evaluated at two points of a hyperinterval $D_i = [a_i, b_i]$. Instead of constructing a minorant function for $f(x)$ over the whole hyperinterval D_i, we use a minorant function for $f(x)$ only over the one-dimensional segment $[a_i, b_i]$. This minorant function is the maximum of two linear functions $K_1(x, \hat{L})$ and $K_2(x, \hat{L})$ passing with the slopes $\pm\hat{L}$ through the vertices a_i and b_i (see Figure 2(b)). The lower bound of $f(x)$ over the diagonal $[a_i, b_i]$ of D_i is calculated similarly to [27, 32] at the intersection of the lines $K_1(x, \hat{L})$ and $K_2(x, \hat{L})$ and is given by the following formula (see [20, 25, 26]):

$$(2.9) \qquad R_i = R_i(\hat{L}) = \frac{1}{2}(f(a_i) + f(b_i) - \hat{L}\|b_i - a_i\|), \qquad 0 < L \le \hat{L} < \infty.$$

A valid estimate of the lower bound of $f(x)$ over D_i can be obtained from (2.9) if an overestimate $\hat{L}$ of the Lipschitz constant L is used. As has been proved in [25, 26], inequality

$$(2.10) \qquad \hat{L} \ge 2L$$

guarantees that the value R_i from (2.9) is the lower bound of $f(x)$ over the whole hyperinterval D_i. Thus, the lower bound of the objective function over the whole hyperinterval $D_i \subseteq D$ can be estimated considering $f(x)$ only along the main diagonal $[a_i, b_i]$ of D_i.

A more precise condition than (2.10) ensuring that

$$R_i(\hat{L}) \le f(x), \qquad x \in D_i,$$

is proved in the following theorem.

THEOREM 2.1. *Let L be the known Lipschitz constant for $f(x)$ from* (1.2)*, $D_i = [a_i, b_i]$ be a hyperinterval of a current partition $\{D^k\}$, and f_i^* be the minimum function value over D_i, i.e.,*

$$f_i^* = f(x_i^*), \qquad x_i^* = \arg\min_{x \in D_i} f(x). \tag{2.11}$$

If an overestimate $\hat{L}$ in (2.9) *satisfies inequality*

$$\hat{L} \geq \sqrt{2}L, \tag{2.12}$$

then $R_i(\hat{L})$ from (2.9) *is the lower bound of $f(x)$ over D_i, i.e., $R_i(\hat{L}) \leq f_i^*$.*

Proof. Since x_i^* from (2.11) belongs to D_i and $f(x)$ satisfies the Lipschitz condition (1.2) over D_i, then the following inequalities hold:

$$f(a_i) - f_i^* \leq L\|a_i - x_i^*\|,$$

$$f(b_i) - f_i^* \leq L\|b_i - x_i^*\|.$$

By summarizing these inequalities and using from [24, Lemma 2] the result

$$\max_{x \in D_i}(\|a_i - x\| + \|b_i - x\|) \leq \sqrt{2}\|b_i - a_i\|,$$

we obtain

$$f(a_i) + f(b_i) \leq 2f_i^* + L(\|a_i - x_i^*\| + \|b_i - x_i^*\|)$$

$$\leq 2f_i^* + L\max_{x \in D_i}(\|a_i - x\| + \|b_i - x\|) \leq 2f_i^* + \sqrt{2}L\|b_i - a_i\|.$$

Then, from the last inequality and (2.12) we can deduce that the following estimate holds for the value R_i from (2.9):

$$R_i(\hat{L}) \leq \frac{1}{2}(2f_i^* + \sqrt{2}L\|b_i - a_i\| - \hat{L}\|b_i - a_i\|)$$

$$= f_i^* + \frac{1}{2}\underbrace{(\sqrt{2}L - \hat{L})}_{\leq 0}\|b_i - a_i\| \leq f_i^*. \qquad \square$$

Theorem 2.1 allows us to obtain a more precise lower bound R_i with respect to [25, 26] where estimate (2.10) is considered.

2.3. Finding nondominated hyperintervals. Let us now consider a diagonal partition $\{D^k\}$ of the admissible region D, generated by the new subdivision strategy from section 2.1. Let a positive value $\tilde{L}$ be chosen as an estimate of the Lipschitz constant L from (1.2) and lower bounds $R_i(\tilde{L})$ of the objective function over hyperintervals $D_i \in \{D^k\}$ be calculated by formula (2.9). Using the obtained lower bounds of $f(x)$, the relation of domination can be established between every two hyperintervals of a current partition $\{D^k\}$ of D.

DEFINITION 2.1. *Given an estimate $\tilde{L} > 0$ of the Lipschitz constant L from* (1.2)*, a hyperinterval $D_i \in \{D^k\}$* dominates *a hyperinterval $D_j \in \{D^k\}$ with respect to $\tilde{L}$ if*

$$R_i(\tilde{L}) < R_j(\tilde{L}).$$

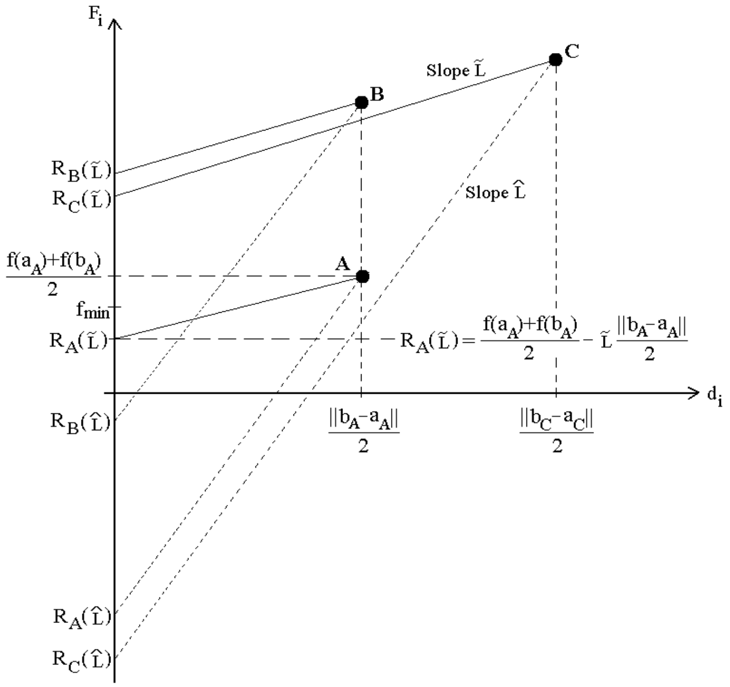


FIG. 3. *Graphical interpretation of lower bounds of $f(x)$ over hyperintervals.*

A hyperinterval $D_t \in \{D^k\}$ is said to be nondominated with respect to $\tilde{L} > 0$ *if for the chosen value $\tilde{L}$ there is no other hyperinterval in $\{D^k\}$ which dominates D_t.*

Each hyperinterval $D_i = [a_i, b_i] \in \{D^k\}$ can be represented by a dot in a two-dimensional diagram (see Figure 3) similar to that used in DIRECT for representing hyperintervals with $f(x)$ evaluated only at one point. The horizontal coordinate d_i and the vertical coordinate F_i of the dot are defined as follows:

$$(2.13) \qquad d_i = \frac{\|b_i - a_i\|}{2}, \quad F_i = \frac{f(a_i) + f(b_i)}{2}, \quad a_i \neq b_i.$$

Note that a point (d_i, F_i) in the diagram can correspond to several hyperintervals with the same length of the main diagonals and the same sum of the function values at their vertices.

For the sake of illustration, let us consider a hyperinterval D_A with the main diagonal $[a_A, b_A]$. This hyperinterval is represented by the dot A in Figure 3. Assuming an estimate of the Lipschitz constant equal to $\tilde{L}$ (such that condition (2.12) is satisfied), a lower bound of $f(x)$ over the hyperinterval D_A is given by the value $R_A(\tilde{L})$ from (2.9). This value is the vertical coordinate of the intersection point of the line passed through the point A with the slope $\tilde{L}$ and the vertical coordinate axis (see Figure 3). In fact, as can be seen from (2.9), intersection of the line with the slope $\tilde{L}$ passed through any dot representing a hyperinterval in the diagram of Figure 3 and the vertical coordinate axis gives us the lower bound (2.9) of $f(x)$ over the corresponding hyperinterval.

Note that the points on the vertical axis ($d_i = 0$) do not represent any hyperinterval. The axis is used to express such values as lower bounds, the current minimum value of the function, etc. It should be highlighted that the current best value f_{min}

is always smaller than or equal to the vertical coordinate of the lowest dot (dot A in Figure 3). Note also that the vertex at which this value has been obtained can belong to a hyperinterval, different from that represented by the lowest dot in the diagram.

By using this graphical representation, it is easy to determine whether a hyperinterval dominates (with respect to a given estimate of the Lipschitz constant) some other hyperinterval from a partition $\{D^k\}$. For example, for the estimate $\tilde{L}$ the following inequalities are satisfied (see Figure 3):

$$R_A(\tilde{L}) < R_C(\tilde{L}) < R_B(\tilde{L}).$$

Therefore, with respect to $\tilde{L}$ the hyperinterval D_A (dot A in Figure 3) dominates both hyperintervals D_B (dot B) and D_C (dot C), while D_C dominates D_B. If our partition $\{D^k\}$ consists only of these three hyperintervals, the hyperinterval D_A is nondominated with respect to $\tilde{L}$.

If a higher estimate $\hat{L} > \tilde{L}$ of the Lipschitz constant is considered (see Figure 3), the hyperinterval D_A still dominates the hyperinterval D_B with respect to $\hat{L}$, since $R_A(\hat{L}) < R_B(\hat{L})$. But D_A in its turn is dominated by the hyperinterval D_C with respect to $\hat{L}$, because $R_A(\hat{L}) > R_C(\hat{L})$ (see Figure 3). Thus, for the chosen estimate $\hat{L}$ the unique nondominated hyperinterval with respect to $\hat{L}$ is D_C, and not D_A as previously.

As we can see from this simple example, some hyperintervals (as the hyperinterval D_B in Figure 3) are always dominated by another hyperintervals, independently of the chosen estimate of the Lipschitz constant L. The following result formalizing this fact takes place.

Lemma 2.1. *Given a partition $\{D^k\}$ of D and the subset $\{D^k\}_d$ of hyperintervals having the main diagonals equal to $d > 0$, for any estimate $\tilde{L} > 0$ of the Lipschitz constant a hyperinterval $D_t \in \{D^k\}_d$ dominates a hyperinterval $D_j \in \{D^k\}_d$ if and only if*

$$F_t = \min\{F_i : \ D_i \in \{D^k\}_d\} < F_j, \tag{2.14}$$

where F_i and F_j are from (2.13).

Proof. The lemma follows immediately from (2.9) since all hyperintervals under consideration have the same length of their main diagonals, i.e., $\|b_i - a_i\| = d$. □

There also exist hyperintervals (for example, the hyperintervals D_A and D_C represented in Figure 3 by the dots A and C, respectively) that are nondominated with respect to one estimate of the Lipschitz constant L and dominated with respect to another estimate of L. Since in practical applications the exact Lipschitz constant (or its valid overestimate) is often unknown, the following idea inspired by DIRECT [18] is adopted.

At each iteration $k > 1$ of the new algorithm, various estimates of the Lipschitz constant L from zero to infinity are chosen for lower bounding $f(x)$ over hyperintervals. The lower bound of $f(x)$ over a particular hyperinterval is calculated by formula (2.9). Note that since all possible values of the Lipschitz constant are considered, condition (2.12) is automatically satisfied and no additional multipliers are required for an estimate of the Lipschitz constant in (2.9). Examination of the set of possible estimates of the Lipschitz constant leads us to the following definition.

Definition 2.2. *A hyperinterval $D_t \in \{D^k\}$ is called* nondominated *if there exists an estimate $0 < \tilde{L} < \infty$ of the Lipschitz constant L such that D_t is nondominated with respect to $\tilde{L}$.*

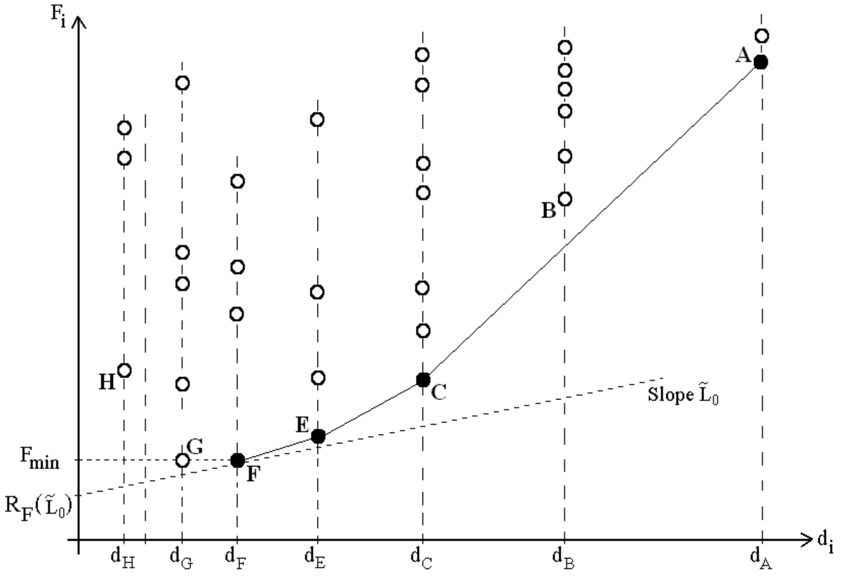


FIG. 4. *Dominated hyperintervals are represented by white dots and nondominated hyperintervals are represented by black dots.*

In other words, nondominated hyperintervals are hyperintervals over which $f(x)$ has the smallest lower bound for some particular estimate of the Lipschitz constant. For example, in Figure 3 the hyperintervals D_A and D_C are nondominated.

Let us now make some observations that allow us to identify the set of nondominated hyperintervals. First of all, only hyperintervals D_t satisfying condition (2.14) can be nondominated. In the two-dimensional diagram (d_i, F_i), where d_i and F_i are from (2.13), such hyperintervals are located at the bottom of each group of points with the same horizontal coordinate, i.e., with the same length of the main diagonals. For example, in Figure 4 these points are designated as A (the largest interval), B, C, E, F, G, and H (the smallest interval).

It is important to notice that not all hyperintervals satisfying (2.14) are nondominated. For example (see Figure 4), the hyperinterval D_H is dominated (with respect to any positive estimate of the Lipschitz constant L) by any of the hyperintervals D_G, D_F, or D_E. The hyperinterval D_G is dominated by D_F. In fact, as follows from (2.9), among several hyperintervals with the same sum of the function values at their vertices, larger hyperintervals dominate smaller ones with respect to any positive estimate of L. Finally, the hyperinterval D_B is dominated either by the hyperinterval D_A (for example, with respect to $\tilde{L}_1 \geq \tilde{L}_{AC}$, where $\tilde{L}_{AC}$ corresponds to the slope of the line passed through the points A and C in Figure 4), or by the hyperinterval D_C (with respect to $\tilde{L}_2 < \tilde{L}_{AC}$).

Note that if an estimate $\tilde{L}$ of the Lipschitz constant is chosen, it is easy to indicate the hyperinterval with the smallest lower bound of $f(x)$, that is, the nondominated hyperinterval with respect to $\tilde{L}$. To do this, it is sufficient to position a line with the slope $\tilde{L}$ below the set of dots in the two-dimensional diagram representing hyperintervals of $\{D^k\}$, and then to shift it upwards. The first dot touched by the line indicates the desirable hyperinterval. For example, in Figure 4 the hyperinterval D_F represented by the point F is a nondominated hyperinterval with respect to $\tilde{L}_0$, since over this hyperinterval $f(x)$ has the smallest lower bound $R_F(\tilde{L}_0)$ for the given estimate $\tilde{L}_0$ of the Lipschitz constant.

Let us now examine various estimates of the Lipschitz constant L from zero to infinity. When a small (close to zero) positive estimate of L is chosen, an almost horizontal line is considered in the two-dimensional diagram representing hyperintervals of a partition $\{D^k\}$. The dot with the smallest vertical coordinate F_{min} (and the biggest horizontal coordinate if there are several such dots) is the first to be touched by this line (the case of the dot F in Figure 4). Therefore, the hyperinterval (or hyperintervals) represented by this dot is nondominated with respect to the chosen estimate of L and, consequently, nondominated in the sense of Definition 2.2. Repeating such a procedure with higher estimates of the Lipschitz constant (that is, considering lines with higher slopes), all nondominated hyperintervals can be identified. In Figure 4 the hyperintervals represented by the dots F, E, C, and A are nondominated hyperintervals.

This procedure can be formalized in terms of the algorithm known as Jarvis march (or gift wrapping; see, e.g., [28]), which is an algorithm for identifying the convex hull of the dots. Thus, the following result identifying the set of nondominated hyperintervals for a given partition $\{D^k\}$ has been proved.

THEOREM 2.2. *Let each hyperinterval* $D_i = [a_i, b_i] \in \{D^k\}$ *be represented by a dot with horizontal coordinate* d_i *and vertical coordinate* F_i *defined in* (2.13). *Then, hyperintervals that are nondominated in the sense of Definition* 2.2 *are located on the lower-right convex hull of the set of dots representing the hyperintervals.*

We conclude this theoretical consideration by the following remark. As has been shown in [31], the lengths of the main diagonals of hyperintervals generated by the new subdivision strategy from section 2.1 are not arbitrary, contrary to traditional diagonal schemes (see, e.g., [20, 24, 25, 26]). They are members of a sequence of values depending both on the size of the initial hypercube $D = [a, b]$ and on the number of executed subdivisions. In such a way, the hyperintervals of a current partition $\{D^k\}$ form several groups. Each group is characterized by the length of the main diagonals of hyperintervals within the group. In the two-dimensional diagram (d_i, F_i), where d_i and F_i are from (2.13), the hyperintervals from a group are represented by dots with the same horizontal coordinate d_i. For example, in Figure 4 there are seven different groups of hyperintervals with the horizontal coordinates equal to d_A, d_B, d_C, d_E, d_F, d_G, and d_H. Note that some groups of a current partition can be empty (see, e.g., the group with the horizontal coordinate between d_H and d_G in Figure 4). These groups correspond to diagonals which are not present in the current partition but can be created (or were created) at the successive (previous) iterations of the algorithm.

It is possible to demonstrate (see [31]) that there exists a correspondence between the length of the main diagonal of a hyperinterval $D_i \in \{D^k\}$ and a nonnegative integer number. This number indicates how many partitions have been performed starting from the initial hypercube D to obtain the hyperinterval D_i. At each iteration $k \geq 1$ it can be considered as an index $l = l(k)$ of the corresponding group of hyperintervals having the same length of their main diagonals, where

$$0 \leq q(k) \leq l(k) \leq Q(k) < +\infty \tag{2.15}$$

and $q(k) = q$ and $Q(k) = Q$ are indices corresponding to the groups of the largest and smallest hyperintervals of $\{D^k\}$, respectively. When the algorithm starts, there exists only one hyperinterval—the admissible region D—which belongs to the group with the index $l = 0$. In this case, both indices q and Q are equal to zero. When a hyperinterval $D_i \in \{D^k\}$ from a group $l' = l'(k)$ is subdivided, all three generated hyperintervals are placed into the group with the index $l' + 1$. Thus, during the work of the algorithm, diagonals of hyperintervals become smaller and smaller, while the corresponding indices of groups of hyperintervals grow consecutively starting from zero.

For example, in Figure 4 there are seven nonempty groups of hyperintervals of a partition $\{D^k\}$ and one empty group. The index $q(k)$ (index $Q(k)$) corresponds to the group of the largest (smallest) hyperintervals represented in Figure 4 by dots with the horizontal coordinate equal to d_A (d_H). For Figure 4 we have $Q(k) = q(k) + 7$. The empty group has the index $l(k) = Q(k) - 1$. Suppose that the hyperintervals D_A, D_H, and D_G (represented in Figure 4 by the dots A, H, and G, respectively) will be subdivided at the kth iteration. In this case, the smallest index will remain the same, i.e., $q(k + 1) = q(k)$, since the group of the largest hyperintervals will not be empty, while the biggest index will increase, i.e., $Q(k + 1) = Q(k) + 1$, since a new group of the smallest hyperintervals will be created. The previously empty group $Q(k) - 1$ will be filled up by the new hyperintervals generated by partitioning the hyperinterval D_G and will have the index $l(k + 1) = Q(k + 1) - 2$.

3. New algorithm. In this section, a new algorithm for solving problem (1.1)–(1.3) is described. First, the new algorithm is presented and briefly commented on. Then its convergence properties are analyzed.

The new algorithm is oriented on solving difficult multidimensional multiextremal problems. To accomplish this task, a two-phase approach consisting of explicitly defined global and local phases is proposed. It is well known that DIRECT also balances global and local information during its work. However, the local phase is too pronounced in this balancing. As has been already mentioned in the introduction, DIRECT executes too many function trials in regions of local optima and, therefore, manifests too slow convergence to the global minimizers when the objective function has many local minimizers.

In the new algorithm, when a sufficient number of subdivisions of hyperintervals near the current best point has been performed, the two-phase approach forces the new algorithm to switch to the exploration of large hyperintervals that could contain better solutions. Since many subdivisions have been executed around the current best point, its neighborhood contains only small hyperintervals and large ones can be located only far from it. Thus, the new algorithm balances global and local search in a more sophisticated way trying to provide a faster convergence to the global minimizers of difficult multiextremal functions.

Thus, the new algorithm consists of the following two phases: local improvement of the current best function value (local phase) and examination of large unexplored hyperintervals in pursuit of new attraction regions of deeper local minimizers (global phase). Each of these phases can consist of several iterations. During the local phase the algorithm tries to better explore the subregion around the current best point. This phase finishes when the following two conditions are verified: (i) an improvement on at least 1% of the minimal function value is not more reached and (ii) a hyperinterval containing the current best point becomes the smallest one. After the end of the local phase the algorithm is switched to the global phase.

The global phase consists of subdividing mainly large hyperintervals, located possibly far from the current best point. It is performed until a function value improving the current minimal value on at least 1% is obtained. When this happens, the algorithm switches to the local phase during which the obtained new solution is improved locally. During its work the algorithm can switch many times from the local phase to the global one. The algorithm stops when the number of generated trial points reaches the maximal allowed number.

We assume without loss of generality that the admissible region $D = [a, b]$ in (1.3) is an N-dimensional hypercube. Suppose that at the iteration $k \geq 1$ of the algorithm a partition $\{D^k\}$ of $D = [a, b]$ has been obtained by the partitioning procedure from (2.1)–(2.8). Suppose also that each hyperinterval $D_i \in \{D^k\}$ is represented by a dot in the two-dimensional diagram (d_i, F_i), where d_i and F_i are from (2.13), and the groups of hyperintervals with the same length of their main diagonals are numerated by indices within a range from $q(k)$ up to $Q(k)$ from (2.15).

To describe the algorithm formally, we need the following additional designations:

$f_{min}(k)$ – the best function value (the term "record" will be also used) found after $k - 1$ iterations.

$x_{min}(k)$ – coordinates of $f_{min}(k)$.

$D_{min}(k)$ – the hyperinterval containing the point $x_{min}(k)$ (if $x_{min}(k)$ is a common vertex of several—up to 2^N—hyperintervals, then the smallest hyperinterval is considered).

f_{min}^{prec} – the old record. It serves to memorize the record $f_{min}(k)$ at the start of the current phase (local or global). The value of f_{min}^{prec} is updated when an improvement of the current record on at least 1% is obtained.

ξ – the parameter of the algorithm, $\xi \geq 0$. It prevents the algorithm from subdividing already well-explored small hyperintervals. If $D_t \in \{D^k\}$ is a nondominated hyperinterval with respect to an estimate $\tilde{L}$ of the Lipschitz constant L, then this hyperinterval can be subdivided at the kth iteration only if the following condition is satisfied:

$$R_t(\tilde{L}) \leq f_{min}(k) - \xi, \tag{3.1}$$

where the lower bound $R_t(\tilde{L})$ is calculated by formula (2.9). The value of ξ can be set in different ways (see section 4).

T_{max} – the maximal allowed number of trial points that the algorithm may generate. The algorithm stops when the number of generated trial points reaches T_{max}. During the course of the algorithm the satisfaction of this termination criterion is verified after every subdivision of a hyperinterval.

Lcounter, *Gcounter* – the counters of iterations executed during the current local and global phases, respectively.

$p(k)$ – the index of the group the hyperinterval $D_{min}(k)$ belongs to. Notice that the inequality $q(k) \leq p(k) \leq Q(k)$ is satisfied for any iteration number k. Since both local and global phases can embrace more than one iteration, the index $p(k)$ (as well as the indices $q(k)$ and $Q(k)$) can change (namely, increase) during these phases. Note also that the group $p(k)$ can be different from the groups containing hyperintervals with the smallest sum of the objective function values at their vertices (see two groups of hyperintervals represented in Figure 4 by the horizontal coordinates equal to d_G and d_F). Moreover, the hyperinterval $D_{min}(k)$ is not represented necessarily by the "lowest" point from the group $p(k)$ in the two-dimensional diagram (d_i, F_i)—even if the current best function value is obtained at a vertex of $D_{min}(k)$, the function value at the other vertex can be too high and the sum of these two values can be greater than the corresponding value of another hyperinterval from the group $p(k)$.

p' – the index of the group containing the hyperinterval $D_{min}(k)$ at the start of the current phase (local or global). Hyperintervals from the groups with indices greater than p' are not considered when nondominated hyperintervals are looked for.

Whereas the index $p(k)$ can assume different values during the current phase, the index p' remains, as a rule, invariable. It is changed only when it violates the left part of condition (2.15). This can happen when groups with the largest hyperintervals disappear and, therefore, the index $q(k)$ increases and becomes equal to p'. In this case, the index p' increases jointly with $q(k)$.

p'' – the index of the group immediately preceding the group p', i.e., $p'' = p' - 1$. This index is used within the local phase and can increase if $q(k)$ increases during this phase.

r' – the index of the middle group of hyperintervals between the groups p' and $q(k)$, i.e., $r' = \lceil (q(k) + p')/2 \rceil$. This index is used within the global phase as a separator between the groups of large and small hyperintervals. It can increase if $q(k)$ increases during this phase.

To clarify the introduced group indices, let us consider an example of a partition $\{D^k\}$ represented by the two-dimensional diagram in Figure 4. Let us suppose that the index $q(k)$ of the group of the largest hyperintervals corresponding to the points with the horizontal coordinate d_A in Figure 4 is equal to 10. The index $Q(k)$ of the group of the smallest hyperintervals with the main diagonals equal to d_H (see Figure 4) is equal to $Q(k) = q(k) + 7 = 17$. Let us also assume that the hyperinterval $D_{min}(k)$ belongs to the group of hyperintervals with the main diagonals equal to d_G (see Figure 4). In this case, the index $p(k)$ is equal to 15 and the index p' is equal to 15 too. The index $p'' = 15 - 1 = 14$ and it corresponds to the group of hyperintervals represented in Figure 4 by the dots with the horizontal coordinate d_F. Finally, the index $r' = \lceil (10 + 15)/2 \rceil = 13$ and it corresponds to hyperintervals with the main diagonals equal to d_E. The indices p', p'', and r' can change only if the index $q(k)$ increases. Otherwise, they remain invariable during the iterations of the current phase (local or global). At the same time, the index $p(k)$ can change at every iteration, as soon as a new best function value belonging to a hyperinterval of a group different from $p(k)$ is obtained.

Now we are ready to present a formal scheme of the new algorithm.

Step 1: Initialization. Set the current iteration number $k := 1$, the current record $f_{min}(k) := \min\{f(a), f(b)\}$, where a and b are from (1.3). Set group indices $q(k) := Q(k) := p(k) := 0$.

Step 2: Local Phase. Memorize the current record $f_{min}^{prec} := f_{min}(k)$ and perform the following steps:

Step 2.1. Set *Lcounter* $:= 1$ and fix the group index $p' := p(k)$.

Step 2.2. Set $p'' := \max\{p' - 1, q(k)\}$.

Step 2.3. Determine nondominated hyperintervals considering only groups of hyperintervals with the indices from $q(k)$ up to p''. Subdivide those nondominated hyperintervals which satisfy inequality (3.1). Set $k := k + 1$.

Step 2.4. Set *Lcounter* $:=$ *Lcounter* $+ 1$ and check whether *Lcounter* $\leq N$. If this is the case, then go to Step 2.2. Otherwise, go to Step 2.5.

Step 2.5. Set $p' = \max\{p', q(k)\}$. Determine nondominated hyperintervals considering only groups of hyperintervals with the indices from $q(k)$ up to p'. Subdivide those nondominated hyperintervals which satisfy inequality (3.1). Set $k := k + 1$.

Step 3: Switch. If condition

$$f_{min}(k) \leq f_{min}^{prec} - 0.01|f_{min}^{prec}| \qquad (3.2)$$

is satisfied, then go to Step 2 and repeat the local phase with the new obtained value of the record $f_{min}(k)$. Otherwise, if the hyperinterval $D_{min}(k)$ is not the smallest one, or the current partition of D consists only of hyperintervals with equal diagonals (i.e., $p(k) < Q(k)$ or $q(k) = Q(k)$), then go to Step 2.1 and repeat the local phase with the old record f_{min}^{prec}.

If the obtained improvement of the best function value is not sufficient to satisfy (3.2) and $D_{min}(k)$ is the smallest hyperinterval of the current partition (i.e., all the following inequalities—(3.2), $p(k) < Q(k)$, and $q(k) = Q(k)$—fail), then go to Step 4 and perform the global phase.

Step 4: Global Phase. Memorize the current record $f_{min}^{prec} := f_{min}(k)$ and perform the following steps:

Step 4.1. Set *Gcounter* := 1 and fix the group index $p' := p(k)$.

Step 4.2. Set $p' = \max\{p', q(k)\}$ and calculate the "middle" group index $r' = \lceil (q(k) + p')/2 \rceil$.

Step 4.3. Determine nondominated hyperintervals considering only groups of hyperintervals with the indices from $q(k)$ up to r'. Subdivide those nondominated hyperintervals which satisfy inequality (3.1). Set $k := k + 1$.

Step 4.4. If condition (3.2) is satisfied, then go to Step 2 and perform the local phase with the new obtained value of the record $f_{min}(k)$. Otherwise, go to Step 4.5.

Step 4.5. Set *Gcounter* := *Gcounter* + 1; check whether *Gcounter* $\leq 2^{N+1}$. If this is the case, then go to Step 4.2. Otherwise, go to Step 4.6.

Step 4.6. Set $p' = \max\{p', q(k)\}$. Determine nondominated hyperintervals considering only groups of hyperintervals with the indices from $q(k)$ up to p'. Subdivide those nondominated hyperintervals which satisfy inequality (3.1). Set $k := k + 1$.

Step 4.7. If condition (3.2) is satisfied, then go to Step 2 and perform the local phase with the new obtained value of the record $f_{min}(k)$. Otherwise, go to Step 4.1: update the value of the group index p' and repeat the global phase with the old record f_{min}^{prec}.

Let us give a few comments on the introduced algorithm. It starts from the local phase. In the course of this phase, it subdivides nondominated hyperintervals with the main diagonals greater than the main diagonal of $D_{min}(k)$ (i.e., from the groups with the indices from $q(k)$ up to p'; see Steps 2.1–2.4). This operation is repeated N times, where N is the problem dimension from (1.3). Recall that during each subdivision of a hyperinterval by the scheme (2.1)–(2.8) only one side of the hyperinterval (namely, the longest side given by formula (2.4)) is partitioned. Thus, performing N iterations of the local phase eventually subdivides all N sides of hyperintervals around the current best point. At the last, $(N+1)$th, iteration of the local phase (see Step 2.5) hyperintervals with the main diagonal equal to $D_{min}(k)$ are considered too. In such a way, the hyperinterval containing the current best point can be partitioned too.

Thus, either the current record is improved or the hyperinterval providing this record becomes smaller. If the conditions of switching to the global phase (see Step 3) are not satisfied, the local phase is repeated. Otherwise, the algorithm switches to the global phase, avoiding unnecessary evaluations of $f(x)$ within already well-explored subregions.

During the global phase the algorithm searches for better new minimizers. It performs a series of loops (see Steps 4.1–4.7) while a nontrivial improvement of the best function value is not obtained, i.e., condition (3.2) is not satisfied. Within a loop of the global phase the algorithm performs a substantial number of subdivisions of large hyperintervals located far from the current best point, namely, hyperintervals from the groups with the indices from $q(k)$ up to r' (see Steps 4.2–4.5). Since each trial point can belong up to 2^N hyperintervals, the number of subdivisions should not be smaller than 2^N. The value of this number equal to 2^{N+1} has been chosen because it provided a good performance of the algorithm in our numerical experiments.

Note that the situation when the current best function value is improved but the amount of this improvement is not sufficient to satisfy (3.2) can be verified at the end of a loop of the global phase (see Step 4.7). In this case, the algorithm is not switched to the local phase. It proceeds with the next loop of the global phase, eventually updating the index p' (see Step 4.1) but not updating the old record f_{min}^{prec}.

Let us now study convergence properties of the new algorithm during minimization of the function $f(x)$ from (1.1)–(1.3) when the maximal allowed number of generated trial points T_{max} is equal to infinity. In this case, the algorithm does not stop (the number of iterations k goes to infinity) and an infinite sequence of trial points $\{x^{j(k)}\}$ is generated. The following theorem establishes the so-called everywhere dense convergence of the new algorithm.

THEOREM 3.1. *For any point $x \in D$ and any $\delta > 0$ there exist an iteration number $k(\delta) \geq 1$ and a point $x' \in \{x^{j(k)}\}$, $k > k(\delta)$, such that $\|x - x'\| < \delta$.*

Proof. Trial points generated by the new algorithm are vertices of the main diagonals of hyperintervals. Due to (2.1)–(2.8), every subdivision of a hyperinterval produces three new hyperintervals with the volume equal to a third of the volume of the subdivided hyperinterval and the proportionally smaller main diagonals. Thus, having fixed a positive value of δ, it is sufficient to prove that after a finite number of iterations $k(\delta)$ the largest hyperinterval of the current partition of D will have the main diagonal smaller than δ. In such a case, in δ-neighborhood of any point of D there will exist at least one trial point generated by the algorithm.

To see this, let us fix an iteration number k' and consider the group $q(k')$ of the largest hyperintervals of a partition $\{D^{k'}\}$. As can be seen from the scheme of the algorithm, for any $k' \geq 1$ this group is taken into account when nondominated hyperintervals are looked for. Moreover, a hyperinterval $D_t \in \{D^{k'}\}$ from this group having the smallest sum of the objective function values at its vertices is partitioned at each iteration $k \geq 1$ of the algorithm. This happens because there always exists a sufficiently large estimate L_∞ of the Lipschitz constant L such that the hyperinterval D_t is a nondominated hyperinterval with respect to L_∞ and condition (3.1) is satisfied for the lower bound $R_t(L_\infty)$ (see Figure 4). Three new hyperintervals generated during the subdivision of D_t by using the strategy (2.1)–(2.8) are inserted into the group with the index $q(k') + 1$. Hyperintervals of the group $q(k') + 1$ have the volume equal to a third of the volume of hyperintervals of the group $q(k')$.

Since each group contains only finite number of hyperintervals, after a sufficiently large number of iterations $k > k'$ all hyperintervals of the group $q(k')$ will be subdivided. The group $q(k')$ will become empty and the index of the group of the largest hyperintervals will increase, i.e., $q(k) = q(k') + 1$. Such a procedure will be repeated with a new group of the largest hyperintervals. So, when the number of iterations grows, the index $q(k)$ increases and due to (2.15) the index $Q(k)$ increases too. This means that there exists a finite number of iterations $k(\delta)$ such that after performing $k(\delta)$ iterations of the algorithm the largest hyperinterval of the current partition $\{D^{k(\delta)}\}$ will have the main diagonal smaller than δ. □

4. Numerical results. In this section, we present results performed to compare the new algorithm with two methods belonging to the same class: the original DIRECT algorithm from [18] and its locally biased modification DIRECTl from [10, 11]. The implementation of these two methods described in [8, 10] and downloadable from [9] has been used in all the experiments.

To execute a numerical comparison, we need to define the parameter ξ of the algorithm from (3.1). This parameter can be set either independently from the current record $f_{min}(k)$ or in a relation with it. Since the objective function $f(x)$ is supposed to be black box, it is not possible to know a priori which of these two ways is better.

In DIRECT [18], where a similar parameter is used, a value ξ related to the current minimal function value $f_{min}(k)$ is fixed as follows:

$$\xi = \varepsilon |f_{min}(k)|, \qquad \varepsilon \geq 0. \tag{4.1}$$

The choice of ε between 10^{-3} and 10^{-7} has demonstrated good results for DIRECT on a set of test functions (see [18]). Later formula (4.1) has been used by many authors (see, e.g., [3, 6, 10, 11, 13]) and also has been realized in the implementation of DIRECT (see [8, 10]) taken for numerical comparison with the new algorithm. Since the value of $\varepsilon = 10^{-4}$ recommended in [18] has produced the most robust results for DIRECT (see, e.g., [10, 11, 13, 18]), exactly this value was used in (4.1) for DIRECT in our numerical experiments. In order to have comparable results, the same formula (4.1) and $\varepsilon = 10^{-4}$ were used in the new algorithm too.

The global minimizer $x^* \in D$ was considered to be found when an algorithm generated a trial point x' inside a hyperinterval with a vertex x^* and the volume smaller than the volume of the initial hyperinterval $D = [a, b]$ multiplied by an accuracy coefficient Δ, $0 < \Delta \leq 1$, i.e.,

$$|x'(i) - x^*(i)| \leq \sqrt[N]{\Delta}(b(i) - a(i)), \qquad 1 \leq i \leq N, \tag{4.2}$$

where N is from (1.3). This condition means that, given Δ, a point x' satisfies (4.2) if the hyperinterval with the main diagonal $[x', x^*]$ and the sides proportional to the sides of the initial hyperinterval $D = [a, b]$ has a volume at least Δ^{-1} times smaller than the volume of D. Note that if in (4.2) the value of Δ is fixed and the problem dimension N increases, the length of the diagonal of the hyperinterval $[x', x^*]$ increases too. In order to avoid this undesirable growth, the value of Δ was progressively decreased when the problem dimension increased.

We stopped the algorithm either when the maximal number of trials T_{max} was reached or when condition (4.2) was satisfied. Note that such a type of stopping criterion is acceptable only when the global minimizer x^* is known, i.e., for the case of test functions. When a real black-box objective function is minimized and global minimization algorithms have an internal stopping criterion, they execute a number of iterations (that can be very high) after a “good” estimate of f^* has been obtained in order to demonstrate a “goodness” of the found solution (see, e.g., [14, 26, 33]).

In the first series of experiments, test functions from [5] and [36] were used because in [10, 11, 18] DIRECT and DIRECTl have been tested on these functions. It can be seen from Table 1 that both methods DIRECT and DIRECTl have executed a very small amount of trials until they generated a point in a neighborhood (4.2) of a global minimizer. For example, condition (4.2) was satisfied for the six-dimensional Hartman’s function after 78 (144) trials performed by DIRECTl (DIRECT). Such a small number of trials is explained by a simple structure of the function. We observe,

Table 1
Number of trial points for test functions used in [18].

Function	N	$D = [a, b]$	Δ	DIRECT	DIRECTl	New
Shekel 5	4	$[0, 10]^4$	10^{-6}	57	53	208
Shekel 7	4	$[0, 10]^4$	10^{-6}	53	45	1465
Shekel 10	4	$[0, 10]^4$	10^{-6}	53	45	1449
Hartman 3	3	$[0, 1]^3$	10^{-6}	113	79	137
Hartman 6	6	$[0, 1]^6$	10^{-7}	144	78	4169
Branin RCOS	2	$[-5, 10] \times [0, 15]$	10^{-4}	41	31	76
Goldstein and Price	2	$[-2, 2]^2$	10^{-4}	37	29	99
Six-Hump Camel	2	$[-3, 3] \times [-2, 2]$	10^{-4}	105	127	128
Shubert	2	$[-8, 10]^2$	10^{-4}	19	15	59

in accordance with [34], that the test functions from [5] used in [18] are not suitable for testing global optimization methods. These functions are characterized by a small chance to miss the region of attraction of the global minimizer (see [34]). Usually, when a real difficult black-box function of high dimension is minimized, the number of trials that it is necessary to execute to place a trial point in the neighborhood of the global minimizer is significantly higher. The algorithm proposed in this paper is oriented on such a type of functions. It tries to perform a good examination of the admissible region in order to reduce the risk of missing the global solution. Therefore, for simple test functions of Table 1 and the stopping rule (4.2) it generated more trial points than DIRECT or DIRECTl.

Hence, more sophisticated test problems are required for carrying out numerical comparison among global optimization algorithms (see also the related discussion in [19]).

Many difficult global optimization tests can be taken from real-life applications (see, e.g., [7] and bibliographic references within it). But the lack of comprehensive information (such as number of local minima, their locations, attraction regions, local and global values, etc.) describing these tests creates an obstacle in verifying efficiency of the algorithms. Very frequently it is also difficult to fix properly many correlated parameters determining some test functions because often the sense of these parameters is not intuitive, especially in high dimensions. Moreover, tests may differ too much one from another and as a result it is not possible to have many test functions with similar properties. Therefore, the use of randomly generated classes of test functions having similar properties can be a reasonable solution for a satisfactory comparison.

Thus, in our numerical experiments we used the GKLS-generator described in [12] (and downloadable for free from http://wwwinfo.deis.unical.it/∼yaro/GKLS.html). It generates classes of multidimensional and multiextremal test functions with known local and global minima. The procedure of generation consists of defining a convex quadratic function (paraboloid) systematically distorted by polynomials. Each test class provided by the generator includes 100 functions and is defined only by the following five parameters:

N – problem dimension;

M – number of local minima;

f^* – value of the global minimum;

ρ^* – radius of the attraction region of the global minimizer;

r^* – distance from the global minimizer to the vertex of the paraboloid.

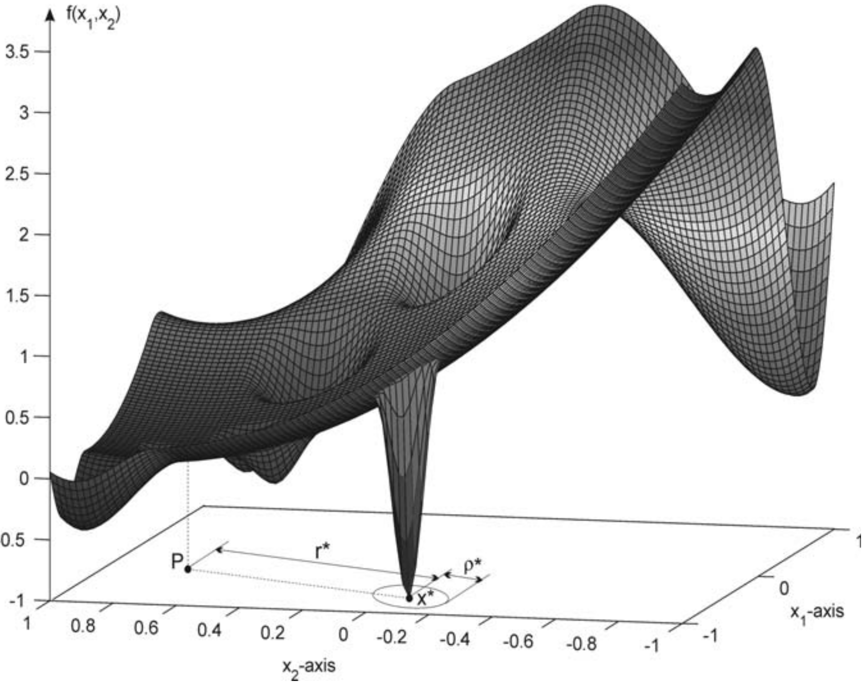


FIG. 5. *An example of the two-dimensional function from the GKLS test class.*

The other necessary parameters are chosen randomly by the generator for each test function of the class. Note that the generator always produces the same test classes for a given set of the user-defined parameters, allowing one to perform repeatable numerical experiments.

By changing the user-defined parameters, classes with different properties can be created. For example, given fixed dimension of the functions and number of local minima, a more difficult class can be created either by shrinking the attraction region of the global minimizer or by moving the global minimizer far away from the paraboloid vertex.

For conducting numerical experiments, we used eight GKLS classes of continuously differentiable test functions of dimensions $N = 2$, 3, 4, and 5. The number of local minima M was equal to 10 and the global minimum value f^* was equal to -1.0 for all classes (these values are default settings of the generator). For each particular problem dimension N we considered two test classes: a simple class and a difficult one. The difficulty of a class was increased either by decreasing the radius ρ^* of the attraction region of the global minimizer (as for two- and five-dimensional classes), or by increasing the distance r^* from the global minimizer x^* to the paraboloid vertex P (three- and four-dimensional classes).

In Figure 5, an example of a test function from the following continuously differentiable GKLS class is given: $N = 2$, $M = 10$, $f^* = -1$, $\rho^* = 0.10$, and $r^* = 0.90$. This function is defined over the region $D = [-1, 1]^2$ and its number is 87 in the given test class. The randomly generated global minimizer of this function is $x^* = (-0.767, -0.076)$ and the coordinates of P are $(-0.489, 0.780)$. Results for the whole class to which the function from Figure 5 belongs are given in Tables 2 and 3 on the second line.

We stopped algorithms either when the maximal number of trials T_{max} equal to 1 000 000 was reached, or when condition (4.2) was satisfied. To describe experiments, we introduce the following designations:

T_s – the number of trials performed by the method under consideration to solve the problem number s, $1 \leq s \leq 100$, of a fixed test class. If the method was not able to solve a problem j in less than T_{max} function evaluations, T_j equal to T_{max} was taken.

m_s – the number of hyperintervals generated to solve the problem s.

The following four criteria were used to compare the methods.

Criterion C1. Number of trials T_{s^*} required for a method to satisfy condition (4.2) for *all* 100 functions of a particular test class, i.e.,

$$T_{s^*} = \max_{1 \leq s \leq 100} T_s, \qquad s^* = \arg \max_{1 \leq s \leq 100} T_s. \tag{4.3}$$

Criterion C2. The corresponding number of hyperintervals, m_{s^*}, generated by the method, where s^* is from (4.3).

Criterion C3. Average number of trials T_{avg} performed by the method during minimization of *all* 100 functions from a particular test class, i.e.,

$$T_{avg} = \frac{1}{100} \sum_{s=1}^{100} T_s. \tag{4.4}$$

Criterion C4. Number p (number q) of functions from a class for which DIRECT or DIRECTl executed less (more) function evaluations than the new algorithm. If T_s is the number of trials performed by the new algorithm and T'_s is the corresponding number of trials performed by a competing method, p and q are evaluated as follows:

$$p = \sum_{s=1}^{100} \delta'_s, \qquad \delta'_s = \begin{cases} 1, & T'_s < T_s, \\ 0 & \text{otherwise}; \end{cases} \tag{4.5}$$

$$q = \sum_{s=1}^{100} \delta_s, \qquad \delta_s = \begin{cases} 1, & T_s < T'_s, \\ 0 & \text{otherwise}. \end{cases} \tag{4.6}$$

If $p + q < 100$, then both the methods under consideration solve the remaining $(100 - p - q)$ problems with the same number of function evaluations.

Note that results based on Criteria C1 and C2 are mainly influenced by minimization of the most difficult functions of a class. Criteria C3 and C4 deal with average data of a class.

Criterion C1 is of fundamental importance for the methods comparison on the whole test class because it shows how many trials it is necessary to execute to solve *all* the problems of a class. Thus, it represents the worst case results of the given method on the fixed class.

At the same time, the number of generated hyperintervals (Criterion C2) provides an important characteristic of any partition algorithm for solving (1.1)–(1.3). It reflects indirectly the degree of qualitative examination of D during the search for a global minimum. The greater the number, the more information about the admissible domain is available and, therefore, the smaller the risk should be of missing the global

Table 2
Number of trial points for GKLS test functions (Criterion C1*).*

N	Δ	Class		50%			100%		
		r^*	ρ^*	DIRECT	DIRECTl	New	DIRECT	DIRECTl	New
2	10^{-4}	.90	.20	111	152	166	1159	2318	403
2	10^{-4}	.90	.10	1062	1328	613	3201	3414	1809
3	10^{-6}	.66	.20	386	591	615	12507	13309	2506
3	10^{-6}	.90	.20	1749	1967	1743	>1000000 (4)	29233	6006
4	10^{-6}	.66	.20	4805	7194	4098	>1000000 (4)	118744	14520
4	10^{-6}	.90	.20	16114	33147	15064	>1000000 (7)	287857	42649
5	10^{-7}	.66	.30	1660	9246	3854	>1000000 (1)	178217	33533
5	10^{-7}	.66	.20	55092	126304	24616	>1000000 (16)	>1000000 (4)	93745

Table 3
Number of hyperintervals for GKLS test functions (Criterion C2*).*

N	Δ	Class		50%			100%		
		r^*	ρ^*	DIRECT	DIRECTl	New	DIRECT	DIRECTl	New
2	10^{-4}	.90	.20	111	152	269	1159	2318	685
2	10^{-4}	.90	.10	1062	1328	1075	3201	3414	3307
3	10^{-6}	.66	.20	386	591	1545	12507	13309	6815
3	10^{-6}	.90	.20	1749	1967	5005	>1000000	29233	17555
4	10^{-6}	.66	.20	4805	7194	15145	>1000000	118744	73037
4	10^{-6}	.90	.20	16114	33147	68111	>1000000	287857	211973
5	10^{-7}	.66	.30	1660	9246	21377	>1000000	178217	206323
5	10^{-7}	.66	.20	55092	126304	177927	>1000000	>1000000	735945

minimizer. However, algorithms should not generate many redundant hyperintervals since this slows down the search and is therefore a disadvantage of the method.

Let us first compare the three methods on Criteria C1 and C2. Results of numerical experiments with eight GKLS tests classes are shown in Tables 2 and 3. The accuracy coefficient Δ from (4.2) is given in the second column of the tables. Table 2 reports the maximal number of trials required for satisfying condition (4.2) for half of the functions of a particular class (columns "50%") and for all 100 function of the class (columns "100%"). The notation "$> 1\,000\,000$ (j)" means that after $1\,000\,000$ function evaluations the method under consideration was not able to solve j problems. The corresponding numbers of generated hyperintervals are indicated in Table 3. Since DIRECT and DIRECTl use during their work the center-sampling partition strategy, the number of generated trial points and the number of generated hyperintervals coincide for these methods.

Note that on half of the test functions from each class (which were the most simple for each method with respect to the other functions of the class) the new algorithm manifested a good performance with respect to DIRECT and DIRECTl in terms of the number of generated trial points (see columns "50%" in Table 2). When all the functions were taken in consideration (and, consequently, difficult functions of the class were considered too), the number of trials produced by the new algorithm was much fewer in comparison with two other methods (see columns "100%" in Table 2), ensuring at the same time a substantial examination of the admissible domain (see Table 3).

In our opinion, the impossibility of DIRECT to determine global minimizers of several test functions is related to the following fact. DIRECT found quickly the vertex of the paraboloid (at which the function value is set by default equal to 0) used for determining GKLS test functions. Hence, the parameter ξ was very close to zero (due to (4.1)) and condition similar to (3.1) was satisfied for almost all small

Table 4
Number of trial points for shifted GKLS test functions (Criterion C1*).*

N	Δ	Class		50%			100%		
		r^*	ρ^*	DIRECT	DIRECTl	New	DIRECT	DIRECTl	New
2	10^{-4}	.90	.20	111	146	165	1087	1567	403
2	10^{-4}	.90	.10	911	1140	508	2973	2547	1767
3	10^{-6}	.66	.20	364	458	606	6292	10202	1912
3	10^{-6}	.90	.20	1485	1268	1515	14807	28759	4190
4	10^{-6}	.66	.20	4193	4197	3462	37036	95887	14514
4	10^{-6}	.90	.20	14042	24948	11357	251801	281013	32822
5	10^{-7}	.66	.30	1568	3818	3011	102869	170709	15343
5	10^{-7}	.66	.20	32926	116025	15071	454925	> 1000000(1)	77981

Table 5
Number of hyperintervals for shifted GKLS test functions (Criterion C2*).*

N	Δ	Class		50%			100%		
		r^*	ρ^*	DIRECT	DIRECTl	New	DIRECT	DIRECTl	New
2	10^{-4}	.90	.20	111	146	281	1087	1567	685
2	10^{-4}	.90	.10	911	1140	905	2973	2547	3227
3	10^{-6}	.66	.20	364	458	1585	6292	10202	5337
3	10^{-6}	.90	.20	1485	1268	4431	14807	28759	12949
4	10^{-6}	.66	.20	4193	4197	14961	37036	95887	73049
4	10^{-6}	.90	.20	14042	24948	57111	251801	281013	181631
5	10^{-7}	.66	.30	1568	3818	17541	102869	170709	106359
5	10^{-7}	.66	.20	32926	116025	108939	454925	> 1000000	685173

hyperintervals. Moreover, many small hyperintervals around the paraboloid vertex with function values close to one another and to the current minimal value were created. In such a situation, DIRECT subdivided many of these hyperintervals. Thus, at each iteration DIRECT partitioned a large number of small hyperintervals and, therefore, was not able to go out from the attraction region of the paraboloid vertex.

Since DIRECTl at each iteration subdivides only one hyperinterval among all hyperintervals with the same function value, it was able to determine some other local minimizers (and the global minimizer too) in the given maximal number of trials T_{max}. Thus, DIRECTl overcame the stagnation of the search around the paraboloid vertex. But due to the locally biased character of DIRECTl, it spent too many trials exploring various local minimizers which were not global. For this reason, DIRECTl was unable to find the global minimizers of four difficult five-dimensional functions.

In order to avoid stagnation of DIRECT near the paraboloid vertex and to put DIRECT and DIRECTl in a more advantageous situation, we shifted all generated functions, adding to their values the constant 2. In such a way the value of each function at the paraboloid vertex became equal to 2 (and the global minimum value f^* was increased by 2, i.e., became equal to 1). Results of numerical experiments with shifted GKLS classes (defined in the rest by the same parameters) are reported in Tables 4 and 5. Note that in this case DIRECT has found the global solutions of all problems. DIRECTl has not found the global minimizer of one five-dimensional function. It can be seen from Tables 4 and 5 that also on these test classes the new algorithm has manifested its superiority with respect to DIRECT and DIRECTl in terms of the number of generated trial points (Criterion C1).

Table 6 summarizes (based on the data from Tables 2–5) results (in terms of Criterion C1) of numerical experiments performed on 1600 test functions from GKLS and shifted GKLS continuously differentiable classes. It represents the ratio between

Table 6
Improvement obtained by the new algorithm in terms of Criterion C1.

N	Δ	Class		GKLS		Shifted GKLS	
		r^*	ρ^*	DIRECT/New	DIRECTl/New	DIRECT/New	DIRECTl/New
2	10^{-4}	.90	.20	2.88	5.75	2.70	3.89
2	10^{-4}	.90	.10	1.77	1.89	1.68	1.44
3	10^{-6}	.66	.20	4.99	5.31	3.29	5.34
3	10^{-6}	.90	.20	>166.50	4.87	3.53	6.86
4	10^{-6}	.66	.20	>68.87	8.18	2.55	6.61
4	10^{-6}	.90	.20	>23.45	6.75	7.67	8.56
5	10^{-7}	.66	.30	>29.82	5.31	6.70	11.13
5	10^{-7}	.66	.20	>10.67	>10.67	5.83	>12.82

the maximal number of trials performed by DIRECT and DIRECTl with respect to the corresponding number of trials performed by the new algorithm. It can be seen from Table 6 that the new method outperforms both competitors significantly on the given test classes when Criteria C1 and C2 are considered.

Let us now compare the three methods using Criteria C3 and C4. Tables 7 and 8 report the average number of trials performed during minimization of all 100 functions from the same GKLS and shifted GKLS classes, respectively (Criterion C3). The "Improvement" columns in these tables represent the ratios between the average numbers of trials performed by DIRECT and DIRECTl with respect to the corresponding numbers of trials performed by the new algorithm. The symbol ">" reflects the situation when not all functions of a class were successfully minimized by the method under consideration in the sense of condition (4.2). This means that the method stopped when T_{max} trials had been executed during minimization of several functions of this particular test class. In these cases, the value of T_{max} equal to 1 000 000 was used in calculations of the average value in (4.4), providing in such a way a lower estimate of the average. As can be seen from Tables 7 and 8, the new method outperforms DIRECT and DIRECTl also on Criterion C3.

Finally, results of comparison between the new algorithm and its two competitors in terms of Criterion C4 are reported in Table 9. This table shows how often the new algorithm was able to minimize each of 100 functions of a class with a smaller number of trials with respect to DIRECT or DIRECTl. The notation "$p : q$" means that among 100 functions of a particular test class there are p functions for which DIRECT (or DIRECTl) spent fewer function trials than the new algorithm and q functions for which the new algorithm generated fewer trial points with respect to DIRECT (or DIRECTl) (p and q are from (4.5) and (4.6), respectively). For example, let us compare the new method with DIRECTl on the GKLS two-dimensional class with parameters $r^* = 0.90$, $\rho^* = 0.20$ (see Table 9, the cell "52 : 47" in the first line). We can see that DIRECTl was better (was worse) than the new method on $p = 52$ ($q = 47$) functions of this class, and one problem was solved by the two methods with the same number of trials.

It can be seen from Table 9 that DIRECT and DIRECTl behave better than the new algorithm with respect to Criterion C4 when simple functions are minimized (we recall that for each problem dimension the first class is simpler than the second one). For example, for the difficult GKLS two-dimensional class and DIRECTl we have 23 : 77 instead of 52 : 47 for the simple class. If a more difficult test class is taken, the new method outperforms its two competitors (see the second—difficult—classes of the dimensions $N = 2$, 4, and 5 in Table 9). For the three-dimensional

Table 7
Average number of trial points for GKLS test functions (Criterion C3*).*

N	Δ	Class		DIRECT	DIRECTl	New	Improvement	
		r^*	ρ^*				DIRECT/New	DIRECTl/New
2	10^{-4}	.90	.20	198.89	292.79	176.25	1.13	1.66
2	10^{-4}	.90	.10	1063.78	1267.07	675.74	1.57	1.88
3	10^{-6}	.66	.20	1117.70	1785.73	735.76	1.52	2.43
3	10^{-6}	.90	.20	>42322.65	4858.93	2006.82	>21.09	2.42
4	10^{-6}	.66	.20	>47282.89	18983.55	5014.13	>9.43	3.79
4	10^{-6}	.90	.20	>95708.25	68754.02	16473.02	>5.81	4.17
5	10^{-7}	.66	.30	>16057.46	16758.44	5129.85	>3.13	3.27
5	10^{-7}	.66	.20	>217215.58	>269064.35	30471.83	>7.13	>8.83

Table 8
Average number of trial points for shifted GKLS test functions (Criterion C3*).*

N	Δ	Class		DIRECT	DIRECTl	New	Improvement	
		r^*	ρ^*				DIRECT/New	DIRECTl/New
2	10^{-4}	.90	.20	185.83	249.25	173.43	1.07	1.44
2	10^{-4}	.90	.10	953.34	1088.13	609.36	1.56	1.79
3	10^{-6}	.66	.20	951.04	1434.33	683.73	1.39	2.10
3	10^{-6}	.90	.20	2226.36	3707.85	1729.55	1.29	2.14
4	10^{-6}	.66	.20	7110.72	14523.45	4388.22	1.62	3.31
4	10^{-6}	.90	.20	24443.60	56689.06	12336.56	1.98	4.60
5	10^{-7}	.66	.30	5876.99	10487.80	4048.31	1.45	2.59
5	10^{-7}	.66	.20	59834.38	>182385.59	19109.20	3.13	>9.54

Table 9
Comparison between the new algorithm and DIRECT and DIRECTl in terms of Criterion C4*.*

N	Δ	Class		GKLS		Shifted GKLS	
		r^*	ρ^*	DIRECT : New	DIRECTl : New	DIRECT : New	DIRECTl : New
2	10^{-4}	.90	.20	61 : 39	52 : 47	61 : 38	54 : 46
2	10^{-4}	.90	.10	36 : 64	23 : 77	37 : 63	23 : 77
3	10^{-6}	.66	.20	66 : 34	54 : 46	65 : 35	62 : 38
3	10^{-6}	.90	.20	58 : 42	51 : 49	56 : 44	54 : 46
4	10^{-6}	.66	.20	51 : 49	37 : 63	50 : 50	44 : 56
4	10^{-6}	.90	.20	47 : 53	42 : 58	46 : 54	43 : 57
5	10^{-7}	.66	.30	66 : 34	26 : 74	67 : 33	42 : 58
5	10^{-7}	.66	.20	34 : 66	27 : 73	32 : 68	32 : 68

classes DIRECT and DIRECTl were better than the new method (see Table 9). This happens because the second three-dimensional class (even being more difficult than the first one because the number q has increased in all the cases) continues to be too simple. Thus, since the new method is oriented on solving difficult multidimensional multiextremal problems, the more hard objective functions are presented in a test class, the more pronounced is the advantage of the new algorithm.

5. A brief conclusion. The problem of global minimization of a multidimensional "black-box" function satisfying the Lipschitz condition over a hyperinterval with an unknown Lipschitz constant has been considered in this paper. A new algorithm developed in the framework of diagonal approach for solving the Lipschitz global optimization problems has been presented. In the algorithm, the partition of the admissible region into a set of smaller hyperintervals is performed by a new efficient diagonal partition strategy. This strategy allows one to accelerate significantly the search procedure in terms of function evaluations with respect to the traditional diagonal partition strategies. A new technique balancing usage of the local and global information has also been incorporated in the new method.

In order to calculate the lower bounds of $f(x)$ over hyperintervals, possible estimates of the Lipschitz constant varying from zero to infinity are considered at each iteration of the algorithm. The procedure of estimating the Lipschitz constant evolves the ideas of the popular method DIRECT from [18] to the case of diagonal algorithms. The everywhere dense convergence of the new algorithm has been established. Extensive numerical experiments executed on more than 1600 test functions have demonstrated a quite satisfactory performance of the new algorithm with respect to DIRECT [18] and DIRECTl [10, 11] when hard multidimensional functions are minimized.

REFERENCES

[1] C. A. Baker, L. T. Watson, B. Grossman, W. H. Mason, and R. T. Haftka, *Parallel global aircraft configuration design space exploration*, in Practical Parallel Computing, M. Paprzycki, L. Tarricone, and L. T. Yang, eds., Nova Science Publishers, Hauppauge, NY, 2001, pp. 79–96.

[2] M. C. Bartholomew-Biggs, S. C. Parkhurst, and S. P. Wilson, *Using DIRECT to solve an aircraft routing problem*, Comput. Optim. Appl., 21 (2002), pp. 311–323.

[3] R. G. Carter, J. M. Gablonsky, A. Patrick, C. T. Kelley, and O. J. Eslinger, *Algorithms for noisy problems in gas transmission pipeline optimization*, Optim. Eng., 2 (2001), pp. 139–157.

[4] S. E. Cox, R. T. Haftka, C. A. Baker, B. Grossman, W. H. Mason, and L. T. Watson, *A comparison of global optimization methods for the design of a high-speed civil transport*, J. Global Optim., 21 (2001), pp. 415–433.

[5] L. C. W. Dixon and G. P. Szegö, eds., *Towards Global Optimization*, Vol. 2, North–Holland, Amsterdam, 1978.

[6] D. E. Finkel and C. T. Kelley, *An Adaptive Restart Implementation of DIRECT*, Technical report CRSC-TR04-30, Center for Research in Scientific Computation, North Carolina State University, Raleigh, NC, 2004.

[7] C. A. Floudas, P. M. Pardalos, C. Adjiman, W. Esposito, Z. Gümüs, S. Harding, J. Klepeis, C. Meyer, and C. Schweiger, *Handbook of Test Problems in Local and Global Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999.

[8] J. M. Gablonsky, *An Implemention of the DIRECT Algorithm*, Technical report CRSC-TR98-29, Center for Research in Scientific Computation, North Carolina State University, Raleigh, NC, 1998.

[9] J. M. Gablonsky, *DIRECT v2.04 FORTRAN code with documentation*, 2001, http://www4.ncsu.edu/~ctk/SOFTWARE/DIRECTv204.tar.gz.

[10] J. M. Gablonsky, *Modifications of the DIRECT Algorithm*, Ph.D. thesis, North Carolina State University, Raleigh, NC, 2001.

[11] J. M. Gablonsky and C. T. Kelley, *A locally-biased form of the DIRECT algorithm*, J. Global Optim., 21 (2001), pp. 27–37.

[12] M. Gaviano, D. Lera, D. E. Kvasov, and Y. D. Sergeyev, *Algorithm 829: Software for generation of classes of test functions with known local and global minima for global optimization*, ACM Trans. Math. Software, 29 (2003), pp. 469–480.

[13] J. He, L. T. Watson, N. Ramakrishnan, C. A. Shaffer, A. Verstak, J. Jiang, K. Bae, and W. H. Tranter, *Dynamic data structures for a direct search algorithm*, Comput. Optim. Appl., 23 (2002), pp. 5–25.

[14] R. Horst and P. M. Pardalos, eds., *Handbook of Global Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1995.

[15] R. Horst and H. Tuy, *Global Optimization – Deterministic Approaches*, Springer-Verlag, Berlin, 1993.

[16] W. Huyer and A. Neumaier, *Global optimization by multilevel coordinate search*, J. Global Optim., 14 (1999), pp. 331–355.

[17] D. R. Jones, *The DIRECT global optimization algorithm*, in Encyclopedia of Optimization, C. A. Floudas and P. M. Pardalos, eds., Vol. 1, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001, pp. 431–440.

[18] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, *Lipschitzian optimization without the Lipschitz constant*, J. Optim. Theory Appl., 79 (1993), pp. 157–181.

[19] C. Khompatraporn, J. D. Pintér, and Z. B. Zabinsky, *Comparative assessment of algorithms and software for global optimization*, J. Global Optim., 31 (2005), pp. 613–633.

[20] D. E. Kvasov, C. Pizzuti, and Y. D. Sergeyev, *Local tuning and partition strategies for diagonal GO methods*, Numer. Math., 94 (2003), pp. 93–106.

[21] D. E. Kvasov and Ya. D. Sergeyev, *Multidimensional global optimization algorithm based on adaptive diagonal curves*, Comput. Math. Math. Phys., 43 (2003), pp. 42–59.

[22] K. Ljungberg, S. Holmgren, and Ö. Carlborg, *Simultaneous search for multiple QTL using the global optimization algorithm DIRECT*, Bioinformatics, 20 (2004), pp. 1887–1895.

[23] R. H. Mladineo, *An algorithm for finding the global maximum of a multimodal multivariate function*, Math. Program., 34 (1986), pp. 188–200.

[24] A. Molinaro, C. Pizzuti, and Ya. D. Sergeyev, *Acceleration tools for diagonal information global optimization algorithms*, Comput. Optim. Appl., 18 (2001), pp. 5–26.

[25] J. D. Pintér, *Extended univariate algorithms for N-dimensional global optimization*, Computing, 36 (1986), pp. 91–103.

[26] J. D. Pintér, *Global Optimization in Action (Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications)*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[27] S. A. Piyavskij, *An algorithm for finding the absolute extremum of a function*, Comput. Math. Math. Phys., 12 (1972), pp. 57–67 (in English); Zh. Výchisl. Mat. Mat. Fiz., 12 (1972), pp. 888–896 (in Russian).

[28] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, Springer-Verlag, New York, 1993.

[29] Ya. D. Sergeyev, *An information global optimization algorithm with local tuning*, SIAM J. Optim., 5 (1995), pp. 858–870.

[30] Ya. D. Sergeyev, *A one-dimensional deterministic global minimization algorithm*, Comput. Math. Math. Phys., 35 (1995), pp. 705–717.

[31] Ya. D. Sergeyev, *An efficient strategy for adaptive partition of N-dimensional intervals in the framework of diagonal algorithms*, J. Optim. Theory Appl., 107 (2000), pp. 145–168.

[32] B. O. Shubert, *A sequential method seeking the global maximum of a function*, SIAM J. Numer. Anal., 9 (1972), pp. 379–388.

[33] R. G. Strongin and Y. D. Sergeyev, *Global Optimization with Non-convex Constraints: Sequential and Parallel Algorithms*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

[34] A. Törn, M. M. Ali, and S. Viitanen, *Stochastic global optimization: Problem classes and solution techniques*, J. Global Optim., 14 (1999), pp. 437–447.

[35] L. T. Watson and C. Baker, *A fully-distributed parallel global search algorithm*, Engrg. Computations, 18 (2001), pp. 155–169.

[36] Y. Yao, *Dynamic tunneling algorithm for global optimization*, IEEE Trans. Syst., Man, Cybern., 19 (1989), pp. 1222–1230.

SIAM J. OPTIM.
Vol. 16, No. 4, pp. 939–964

DISCRETE TRANSFORMS, SEMIDEFINITE PROGRAMMING, AND SUM-OF-SQUARES REPRESENTATIONS OF NONNEGATIVE POLYNOMIALS*

TAE ROH† AND LIEVEN VANDENBERGHE†

Abstract. We present a new semidefinite programming formulation of sum-of-squares representations of nonnegative polynomials, cosine polynomials, and trigonometric polynomials of one variable. The parametrization is based on discrete transforms (specifically, the discrete Fourier, cosine, and polynomial transforms) and has a simple structure that can be exploited by straightforward modifications of standard interior-point algorithms.

Key words. semidefinite programming, interior-point methods, nonnegative polynomials

AMS subject classifications. 90C22, 90C25

DOI. 10.1137/040612646

1. Introduction.

We discuss fast algorithms for semidefinite programs (SDPs) derived from weighted sum-of-squares representations of polynomials, cosine polynomials, and trigonometric polynomials of one variable.

Several well-known theorems state that a (generalized) polynomial $f : \mathbf{R} \rightarrow \mathbf{R}$ is nonnegative on an interval or a union of intervals I,

$$f(t) \geq 0, \qquad t \in I, \tag{1}$$

if and only if it can be expressed as a *weighted sum of squares*

$$f(t) = \sum_{k=1}^{r} w_k(t)(y_k^{\mathrm{T}} q(t))^2, \tag{2}$$

where $w_k(t) \geq 0$ on I. (For trigonometric polynomials, q and y_k are complex-valued, and we replace $(y_k^{\mathrm{T}} q)^2$ with $|y_k^{\mathrm{H}} q|^2$, where y_k^{H} denotes the complex conjugate transpose of y_k.) The weight functions w_k, the required number of terms r, and the vector of basis functions q depend on I and the class of functions f under consideration. Specific examples of sum-of-squares theorems are given in sections 3.1, 4.1, and 5.1.

It is also well known that the weighted sum-of-squares property (2) can be expressed as a set of linear equations and linear matrix inequalities (LMIs) in the coefficients of f and a number of auxiliary matrix variables. In other words, (2) is equivalent to a convex constraint of the form

$$x = \sum_{i=1}^{s} \mathcal{H}_i(X_i), \quad X_i \succeq 0, \quad i = 1, \ldots, s, \tag{3}$$

where x is the vector of coefficients of f with respect to some basis, $\mathcal{H}_i$ is a linear mapping, and $s \leq r$ [24, 25, 21]. Combining these results, we can cast the constraint (1),

*Received by the editors July 30, 2004; accepted for publication (in revised form) April 4, 2005; published electronically January 26, 2006. This work was supported by NSF grant ECS-0200320.
http://www.siam.org/journals/siopt/16-4/61264.html

†Department of Electrical Engineering, University of California, Los Angeles, CA 90095-1594 (roh@ee.ucla.edu, vandenbe@ee.ucla.edu).

which is an infinite number of linear inequalities in the coefficients x, as a finite number of linear equations and linear matrix inequalities. Thus, we can solve a wide variety of optimization problems over polynomials, subject to piecewise-polynomial upper and lower bounds, as SDPs. Numerous applications of this idea can be found in signal processing and control [26, 23, 27, 11, 34, 4, 8, 9, 18].

In this paper we propose a specific choice for the mappings $\mathcal{H}_i$ in (3). We show that the weighted sum-of-squares property can be expressed in the following common form or its complex-valued counterpart:

$$x = \sum_{i=1}^{s} A_i \,\mathbf{diag}\left(C_i X_i C_i^{\mathrm{T}}\right), \quad X_i \succeq 0, \quad i = 1, \ldots, s, \tag{4}$$

where $\mathbf{diag}(C_i X_i C_i^{\mathrm{T}})$ denotes the vector of diagonal elements of $C_i X_i C_i^{\mathrm{T}}$, and the matrices A_i and C_i are defined in terms of discrete orthogonal transforms and their inverses. This unified parametrization offers several advantages. First, we will see that SDPs with constraints of the form (4), in which x and the matrices X_i are variables, can be solved very efficiently by taking advantage of some simple properties of the **diag** operator. This allows one to develop a single solver that solves SDPs derived from weighted sum-of-squares representations much more quickly than general-purpose codes. Second, in many cases additional savings are possible by using fast discrete transform algorithms for the multiplications with A_i and C_i. Third, the matrices C_i can be chosen to be orthogonal, while A_i is generally a product of an orthogonal and a diagonal matrix. These orthogonality properties are attractive from a numerical stability viewpoint.

Our interest in numerical methods for SDPs derived from sum-of-squares representations is motivated by several recent papers. Nesterov in [24] pointed out the connections between sum-of-squares representations, semidefinite programming, and classical results in moment theory. He also described a straightforward method for converting weighted sum-of-squares representations (2) into constraints of the form (3). We explain the method for the case with $w_i(t) = 1$. Let $q : \mathbf{R} \to \mathbf{R}^{m+1}$. Suppose $p_i(t)$, $i = 0, \ldots, n$, are basis functions whose span contains all products $q_k(t)q_l(t)$, so there exist matrices $F_i \in \mathbf{S}^{m+1}$ such that

$$q(t)q(t)^{\mathrm{T}} = \sum_{i=0}^{n} p_i(t) F_i.$$

A function f can be expressed as a sum of squares $f(t) = \sum_{k=1}^{r} (y_k^{\mathrm{T}} q(t))^2$ for some r and y_k if and only if

$$f(t) = \sum_{k=1}^{r} (y_k^{\mathrm{T}} q(t))^2 = \mathbf{tr}(q(t)q(t)^{\mathrm{T}} X) = \sum_{i=0}^{n} \mathbf{tr}(F_i X) p_i(t),$$

where $X = \sum_{k=1}^{r} y_k y_k^{\mathrm{T}}$. We see that f is a sum of squares if and only if $f(t) = x_0 p_0(t) + \cdots + x_n p_n(t)$, where

$$x_i = \mathbf{tr}(F_i X), \quad i = 0, \ldots, n, \quad X \succeq 0, \tag{5}$$

for some $X \in \mathbf{S}^{m+1}$. Therefore, (3) holds with $\mathcal{H}_1(X) = (\mathbf{tr}(F_0 X), \ldots, \mathbf{tr}(F_n X))$, and $s = 1$.

As an example, it is well known that a nonnegative polynomial of even degree

$$f(t) = x_0 + x_1 t + \cdots + x_{2m} t^{2m}$$

can be expressed as a sum of squares of two polynomials of degree m or less. To derive equivalent LMI conditions, we take $q(t) = (1, t, \ldots, t^m)$, and note that

$$q(t)q(t)^{\mathrm{T}} = \sum_{i=0}^{2m} t^i F_i, \qquad F_{i,kl} = \begin{cases} 1, & k+l=i, \\ 0, & \text{otherwise.} \end{cases}$$

For this choice of F_i, (5) reduces to

$$x_i = \sum_{k+l=i} X_{kl}, \quad i = 0, \ldots, 2m, \quad X \succeq 0. \tag{6}$$

We can conclude that $f(t)$ is nonnegative if and only if there exists an $X \in \mathbf{S}^{m+1}$ such that (6) holds. We refer to Nesterov [24] and Faybusovich [12, 13] for more examples and extensions of Nesterov's approach.

SDPs derived from sum-of-squares representations involve auxiliary matrix variables and are often large scale and difficult to solve using general-purpose solvers. This has spurred research into specialized implementations of interior-point methods. The most successful approaches have been based on dual barrier methods [14, 16, 4] and exploit properties of the logarithmic barrier function for the dual constraints associated with (3). Genin et al. [14] consider problems involving matrix-valued polynomials that are nonnegative on the unit circle, the real axis, or the imaginary axis. They note that the dual variables have low displacement rank (for example, due to Toeplitz or Hankel structure) and use this property to reduce the cost of computing the gradient and Hessian of the dual barrier function. This results in a substantial reduction of the complexity per iteration, as compared to a general-purpose solver. In [4] similar gains are achieved for a more specific class of problems, involving nonnegative scalar trigonometric polynomials. As in the method of [14], the basic idea is to evaluate the gradient and Hessian of the dual barrier function fast. In [4] this is accomplished by using the discrete Fourier transform (DFT) of triangular factors of the inverses of the dual variables. The techniques discussed in this paper can be interpreted as an extension of the DFT method of [4] to a much wider class of problems and to general interior-point methods (primal, dual, or primal-dual). Several of the key ideas in this paper also extend to SDPs derived from sum-of-squares characterizations of multivariate polynomials. In this context, our techniques are related to recent work by Löfberg and Parrilo on improving the efficiency of SDP solvers for sum-of-squares programming (see [22], which appeared after the first submission of this paper).

Notation. The set of real symmetric $n \times n$ matrices is denoted $\mathbf{S}^n$; the set of Hermitian $n \times n$ matrices is denoted $\mathbf{H}^n$. $A \succeq 0$ means A is positive semidefinite; $A \succ 0$ means A is positive definite. $\mathbf{tr}(A)$ is the trace of A. For a square matrix A, $\mathbf{diag}(A)$ is the vector of diagonal elements of A. For an n-vector a, $\mathbf{diag}(a)$ is the diagonal matrix with the elements of a on its diagonal. A^{T} is the transpose of the matrix A, $\bar{A}$ is the complex conjugate, and $A^{\mathrm{H}} = (\bar{A})^{\mathrm{T}}$ is the complex conjugate transpose. $A \circ B$ denotes the Hadamard product of two matrices A and B of the same dimensions, i.e., the matrix with elements $(A \circ B)_{ik} = A_{ik} B_{ik}$. The same notation is used for vectors: $(x \circ y)_i = x_i y_i$. For real matrices, $\mathbf{sqr}(A) = A \circ A$; for complex matrices, $\mathbf{sqr}(A) = A \circ \bar{A}$. We use the notation $(x_0, x_1, \ldots, x_n)$ for the (column) vector $[x_0 \ x_1 \ \cdots \ x_n]^{\mathrm{T}}$. $\mathbf{1}$ is the vector with all components one with dimension determined from the context. Throughout the paper the symbol j is reserved for the number $\sqrt{-1}$. We use $\deg(f)$ to denote the degree of a polynomial, cosine polynomial, or trigonometric polynomial f. For a trigonometric polynomial $f(\omega) = x_0 + 2\Re(x_1 e^{-j\omega} + \cdots + x_n e^{-jn\omega})$, we define $\deg(f) = n$ if $x_n \neq 0$.

2. A class of structured SDPs.

Suppose the matrices F_i in the standard form SDP

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(DX) \\ \text{subject to} & \mathbf{tr}(F_i X) = b_i, \quad i = 1, \ldots, m, \\ & X \succeq 0 \end{array} \tag{7}$$

can be factored as

$$F_i = C^{\mathrm{T}} \mathbf{diag}(a_i) C, \quad i = 1, \ldots, m, \tag{8}$$

where $C \in \mathbf{R}^{q \times n}$ and $a_i \in \mathbf{R}^q$. In other words, the matrices F_i can be written as different linear combinations of q rank-one matrices $c_i c_i^{\mathrm{T}}$, where c_i^{T} is the ith row of C. Substituting (8) in (7) we obtain

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(DX) \\ \text{subject to} & A\, \mathbf{diag}(CXC^{\mathrm{T}}) = b, \\ & X \succeq 0, \end{array} \tag{9}$$

where $A \in \mathbf{R}^{m \times q}$ has rows a_i^{T}. In this section we will see that if $q \ll mn$, the SDP (9) can be solved very efficiently by taking advantage of the structure in the constraints. In sections 3–5 we will then show that this type of structure arises in SDPs derived from sum-of-squares representations of nonnegative polynomials.

Note that a factorization of the form (8) always exists. For example, one can use the eigenvalue decomposition to factor F_i as $F_i = V_i\, \mathbf{diag}(\lambda_i) V_i^{\mathrm{T}}$ with $V_i \in \mathbf{R}^{n \times r_i}$, $\lambda_i \in \mathbf{R}^{r_i}$, where $r_i = \mathbf{rank}(F_i)$, and then take $q = \sum_i r_i$,

$$C = \begin{bmatrix} V_1^{\mathrm{T}} \\ V_2^{\mathrm{T}} \\ \vdots \\ V_m^{\mathrm{T}} \end{bmatrix}, \quad a_1 = \begin{bmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad a_2 = \begin{bmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{bmatrix}, \quad \ldots, \quad a_m = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \lambda_m \end{bmatrix}. \tag{10}$$

For general dense matrices, with $r_i = n$ and $q = mn$, there is no advantage in expressing the SDP as (9). If the matrices F_i are all low rank ($r_i \ll n$), then (10) provides a factorization (8) with $q \ll mn$. In this case our techniques are similar to known methods for exploiting low-rank structure [6]. Our focus in this paper, however, is on more general types of structure in which the matrices F_i are not low-rank.

2.1. Solution via interior-point methods.

It will be convenient in later sections to use the problem format

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(DX) + c^{\mathrm{T}} y \\ \text{subject to} & A\, \mathbf{diag}(CXC^{\mathrm{T}}) + By = b, \\ & X \succeq 0, \end{array} \tag{11}$$

which includes a vector variable $y \in \mathbf{R}^p$. The problem parameters are $c \in \mathbf{R}^p$, $D \in \mathbf{S}^n$, $b \in \mathbf{R}^m$, $A \in \mathbf{R}^{m \times q}$, $B \in \mathbf{R}^{m \times p}$, and $C \in \mathbf{R}^{q \times n}$. The corresponding dual SDP is

$$\begin{array}{ll} \text{maximize} & b^{\mathrm{T}} z \\ \text{subject to} & C^{\mathrm{T}} \mathbf{diag}(A^{\mathrm{T}} z) C \preceq D, \\ & B^{\mathrm{T}} z = c \end{array} \tag{12}$$

with variable $z \in \mathbf{R}^m$.

Interior-point methods for solving the pair of SDPs (11) and (12) typically require the solution of one or two sets of linear equations of the form

$$-T^{-1}\Delta X T^{-1} + C^{\mathrm{T}} \mathbf{diag}(A^{\mathrm{T}}\Delta z)C = R, \tag{13}$$

$$A\,\mathbf{diag}(C\Delta X C^{\mathrm{T}}) + B\Delta y = r_1, \tag{14}$$

$$B^{\mathrm{T}}\Delta z = r_2 \tag{15}$$

at each iteration. The variables are Δy, Δz, ΔX; the matrix $T \succ 0$ and the right-hand sides $R \in \mathbf{S}^n$, $r_1 \in \mathbf{R}^m$, and $r_2 \in \mathbf{R}^p$ are given. We refer to these equations as Newton equations, because they can be obtained by linearizing nonlinear equations that characterize the central path. The matrices T and the right-hand sides R, r_1, r_2 change at each iteration and depend on the particular method used. In some methods (for example, dual barrier methods) the matrix T may have additional structure that can be exploited [5, 14, 4]. In this paper, however, we will make no assumption about T, other than positive definiteness. The technique outlined below, therefore, applies to a wide variety of interior-point methods, including primal methods, dual methods, and primal-dual methods based on the Nesterov–Todd scaling [30]. Other primal-dual methods (in particular, the methods in [3, 17, 19]) involve Newton equations with a closely related structure.

It is well known that the number of iterations in an interior-point method is typically in the range 10–50, almost independent of the problem dimensions, and that the overall cost is dominated by the cost of solving the Newton equations. An efficient method that takes advantage of the structure in the Newton equations (13)–(15) is as follows. We first eliminate ΔX from the first equation to get

$$A\,\mathbf{diag}(CTC^{\mathrm{T}}\,\mathbf{diag}(A^{\mathrm{T}}\Delta z)CTC^{\mathrm{T}}) + B\Delta y = r_3, \tag{16}$$

$$B^{\mathrm{T}}\Delta z = r_2, \tag{17}$$

where $r_3 = r_1 + A\,\mathbf{diag}(CTRTC^{\mathrm{T}})$. The 1,1-block can be written in matrix-vector form by using the identity $\mathbf{diag}(P\,\mathbf{diag}(u)Q^{\mathrm{T}}) = (P \circ Q)u$:

$$\begin{aligned} A\,\mathbf{diag}\left(CTC^{\mathrm{T}}\,\mathbf{diag}(A^{\mathrm{T}}\Delta z)CTC^{\mathrm{T}}\right) &= A\left((CTC^{\mathrm{T}}) \circ (CTC^{\mathrm{T}})\right)A^{\mathrm{T}}\Delta z \\ &= A\,\mathbf{sqr}(CTC^{\mathrm{T}})A^{\mathrm{T}}\Delta z. \end{aligned}$$

Equations (16) and (17), therefore, reduce to $m + p$ equations in $m + p$ variables:

$$\begin{bmatrix} A\,\mathbf{sqr}(CTC^{\mathrm{T}})A^{\mathrm{T}} & B \\ B^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta y \end{bmatrix} = \begin{bmatrix} r_3 \\ r_2 \end{bmatrix}. \tag{18}$$

From the solution Δz, Δy, we find ΔX by solving (13).

To justify this approach, we can contrast it with the calculations used in common general-purpose implementations (such as Sedumi [28] or SDPT3 [31]). In a general-purpose code the Newton equations are also solved by eliminating ΔX and solving the reduced Newton equations (18). The difference lies in the way the $1,1$-block $H = A\,\mathbf{sqr}(CTC^{\mathrm{T}})A^{\mathrm{T}}$ is assembled. In a general-purpose algorithm the linear mapping $C^{\mathrm{T}}\,\mathbf{diag}(A^{\mathrm{T}}z)C$ is represented in the canonical form

$$C^{\mathrm{T}}\,\mathbf{diag}(A^{\mathrm{T}}z)C = \sum_{i=1}^{m} z_i F_i,$$

where $F_i = C^{\mathrm{T}} \,\mathbf{diag}(a_i) C$ and a_i^{T} is the ith row of A. The matrix H is computed as

$$H_{ik} = \mathbf{tr}(TF_iTF_k), \qquad i,k = 1,\ldots,m.$$

These computations can be arranged in different ways, for example, by first computing the m matrices TF_i and then forming the $m(m+1)/2$ inner products $\mathbf{tr}(TF_iTF_k)$. If we assume that the matrices F_i are dense and full-rank and that the problem dimensions m, n, p are of the same order, this yields an $O(n^4)$ method for constructing the coefficient matrix in (18), which can then be solved in $O(n^3)$ operations. The direct formula $H = A\,\mathbf{sqr}(CTC^{\mathrm{T}})A^{\mathrm{T}}$ is faster, because it requires $O(n^3)$ operations (again assuming that all problem dimensions are of the same order). Moreover, in the applications that we describe below, the matrices A and C represent discrete transforms or inverse discrete transforms, so fast methods often exist for multiplications with A and C.

2.2. Extension to complex data and variables.

In applications involving trigonometric polynomials we will encounter SDPs in which some of the data and variables are complex numbers. It is, therefore, of interest to consider the complex counterpart of (11) and (12),

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(DX) + c^{\mathrm{T}}y \\ \text{subject to} & A\,\mathbf{diag}(CXC^{\mathrm{H}}) + By = b, \\ & X \succeq 0, \end{array} \tag{19}$$

$$\begin{array}{ll} \text{maximize} & \Re(b^{\mathrm{H}}z) \\ \text{subject to} & C^{\mathrm{H}}\,\mathbf{diag}\left(\Re(A^{\mathrm{H}}z)\right)C \preceq D, \\ & \Re(B^{\mathrm{H}}z) = c. \end{array}$$

The primal variables are $X \in \mathbf{H}^n$ and $y \in \mathbf{R}^p$. The dual variable is $z \in \mathbf{C}^m$. The problem parameters are $D \in \mathbf{H}^n$, $c \in \mathbf{R}^p$, $A \in \mathbf{C}^{m\times q}$, $C \in \mathbf{C}^{q\times n}$, $B \in \mathbf{C}^{m\times p}$, and $b \in \mathbf{C}^m$.

The Newton equations for (19) can be written as

$$\begin{aligned} -T^{-1}\Delta X T^{-1} + C^{\mathrm{H}}\,\mathbf{diag}\left(\Re(A^{\mathrm{H}}\Delta z)\right)C &= R, \\ A\,\mathbf{diag}(C\Delta X C^{\mathrm{H}}) + B\Delta y &= r_1, \\ \Re(B^{\mathrm{H}}\Delta z) &= r_2. \end{aligned}$$

Eliminating ΔX from the first equation gives

$$A\,\mathbf{diag}\left(CTC^{\mathrm{H}}\,\mathbf{diag}(\Re(A^{\mathrm{H}}\Delta z))CTC^{\mathrm{H}}\right) + B\Delta y = r_3, \tag{20}$$

$$\Re(B^{\mathrm{H}}\Delta z) = r_2, \tag{21}$$

where $r_3 = r_1 + A\,\mathbf{diag}(CTRTC^{\mathrm{H}})$. Again using the identity $\mathbf{diag}(P\,\mathbf{diag}(u)Q^{\mathrm{T}}) = (P \circ Q)u$, we can write the 1,1-block as

$$\begin{aligned} A\,\mathbf{diag}\left(CTC^{\mathrm{H}}\,\mathbf{diag}(\Re(A^{\mathrm{H}}\Delta z))CTC^{\mathrm{H}}\right) &= A\left((CTC^{\mathrm{H}}) \circ (CTC^{\mathrm{H}})^{\mathrm{T}}\right)\Re(A^{\mathrm{H}}\Delta z) \\ &= A\,\mathbf{sqr}(CTC^{\mathrm{H}})\Re(A^{\mathrm{H}}\Delta z). \end{aligned}$$

Plugging this in (20) and (21) and expanding complex data and variables in their real and imaginary parts ($A = A_{\mathrm{r}} + jA_{\mathrm{i}}$, etc.), we obtain

$$\begin{bmatrix} A_{\mathrm{r}}\,\mathbf{sqr}(CTC^{\mathrm{H}})A_{\mathrm{r}}^{\mathrm{T}} & A_{\mathrm{r}}\,\mathbf{sqr}(CTC^{\mathrm{H}})A_{\mathrm{i}}^{\mathrm{T}} & B_{\mathrm{r}} \\ A_{\mathrm{i}}\,\mathbf{sqr}(CTC^{\mathrm{H}})A_{\mathrm{r}}^{\mathrm{T}} & A_{\mathrm{i}}\,\mathbf{sqr}(CTC^{\mathrm{H}})A_{\mathrm{i}}^{\mathrm{T}} & B_{\mathrm{i}} \\ B_{\mathrm{r}}^{\mathrm{T}} & B_{\mathrm{i}}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \Delta z_{\mathrm{r}} \\ \Delta z_{\mathrm{i}} \\ \Delta y \end{bmatrix} = \begin{bmatrix} r_{3,\mathrm{r}} \\ r_{3,\mathrm{i}} \\ r_2 \end{bmatrix}. \tag{22}$$

The extension to the case where only some of the rows of A and B (and the corresponding elements of Δz) in (20) and (21) are complex is straightforward: in (22) we simply delete the equations and variables corresponding to the zero rows in A_{i} and Δz_{i}.

3. Trigonometric polynomials.

Let f be a trigonometric polynomial of degree n or less, i.e., a function of the form

$$\begin{aligned} f(\omega) &= \bar{x}_n e^{jn\omega} + \cdots + \bar{x}_1 e^{j\omega} + x_0 + x_1 e^{-j\omega} + \cdots + x_n e^{-jn\omega} \\ &= x_0 + 2\Re(x_1 e^{-j\omega} + \cdots + x_n e^{-jn\omega}), \end{aligned} \tag{23}$$

where $x = (x_0, \ldots, x_n) \in \mathbf{R} \times \mathbf{C}^n$. In this section we show that f is nonnegative on a subinterval of $[0, 2\pi]$ if and only if it satisfies an SDP constraint of the form

$$x = \sum_{k=1}^{r} A_k \,\mathbf{diag}\left(C_k X_k C_k^{\mathrm{H}}\right), \quad X_k \succeq 0, \quad k = 1, \ldots, r,$$

with $r = 1$ or $r = 2$. This result follows by reformulating classical sum-of-squares characterizations of nonnegative trigonometric polynomials via the discrete Fourier transform.

3.1. Sum-of-squares characterizations.

If the trigonometric polynomial (23) is nonnegative and of degree n (i.e., $x_n \neq 0$), then it can be expressed as

$$f(\omega) = |g(e^{-j\omega})|^2,$$

where $g(s) = u_0 + u_1 s + \cdots + u_n s^n$ is a polynomial of degree n with (in general) complex coefficients u_k. This is known as the *Riesz–Fejér theorem* or the *spectral factorization theorem* [29, p. 3], [20, p. 60]. Several efficient methods exist for computing g from x; see, for example, [32, Appendix D].

The following generalization of the Riesz–Fejér theorem can be found in [2, p. 133], [20, p. 294], [8, Theorem 2], [16, p. 44], [12, 13]. If f is nonnegative on $[\alpha - \beta, \alpha + \beta]$, where $0 < \beta < \pi$, then it can be expressed as

$$f(\omega) = |g(e^{-j\omega})|^2 + (\cos(\omega - \alpha) - \cos\beta)\, |h(e^{-j\omega})|^2,$$

where g and h are polynomials with $\deg(g) \leq n$ and $\deg(h) \leq n - 1$. In other words, f is the sum of two nonnegative trigonometric polynomials. The first trigonometric polynomial $|g(e^{-j\omega})|^2$ is nonnegative everywhere; the second term is the product of a nonnegative trigonometric polynomial $|h(e^{-j\omega})|^2$ with the trigonometric polynomial $\cos(\omega - \alpha) - \cos\beta$, which is nonnegative on $[\alpha - \beta, \alpha + \beta]$.

3.2. Discrete Fourier transform.

The discrete Fourier transform (DFT) offers a convenient way to map the coefficients of a pseudopolynomial

$$F(s) = x_{-n} s^{-n} + \cdots + x_{-1} s^{-1} + x_0 + x_1 s + \cdots + x_n s^n \tag{24}$$

to its values at equidistant points on the unit circle, and vice versa. Let $W_{\mathrm{DFT}} \in \mathbf{C}^{N \times N}$ be the length-N DFT matrix with $N \geq 2n + 1$:

$$W_{\mathrm{DFT}} = \begin{bmatrix} w_0 & w_1 & \cdots & w_{N-1} \end{bmatrix},$$

where

$$w_k = (1, e^{-jk\omega_N}, e^{-j2k\omega_N}, \ldots, e^{-j(N-1)k\omega_N}), \qquad \omega_N = 2\pi/N.$$

For the pseudopolynomial F given by (24), define

$$\tilde{x} = (x_0, x_1, \ldots, x_n, 0, \ldots, 0, x_{-n}, \ldots, x_{-1}) \in \mathbf{C}^N,$$
$$y = \big(F(1), F(e^{-j\omega_N}), \ldots, F\big(e^{-j(N-1)\omega_N}\big)\big) \in \mathbf{C}^N.$$

Then it is easily verified that

$$y = W_{\mathrm{DFT}}\tilde{x}, \qquad \tilde{x} = \frac{1}{N} W_{\mathrm{DFT}}^{\mathrm{H}} y.$$

In other words, the DFT maps the coefficients of F to the values of F at N equidistant points on the unit circle; the inverse DFT maps these sample values back to the coefficients.

If $x_{-k} = \bar{x}_k$, then $F(e^{-j\omega})$ is the trigonometric polynomial

$$F(e^{-j\omega}) = f(\omega) = x_0 + 2\Re(x_1 e^{-j\omega} + \cdots + x_n e^{-jn\omega})$$

and the relation between $x = (x_0, x_1, \ldots, x_n)$ and $y = (f(0), f(\omega_N), \ldots, f((N-1)\omega_N))$ simplifies to

$$x = \frac{1}{N} W^{\mathrm{H}} y,$$

where the columns of W are the first $n+1$ columns of W_{DFT}:

$$W = \begin{bmatrix} w_0 & w_1 & \cdots & w_n \end{bmatrix} \in \mathbf{C}^{N \times (n+1)}. \tag{25}$$

3.3. Semidefinite representations. We now combine the observations in the previous two paragraphs to obtain SDP characterizations of nonnegative trigonometric polynomials. Let f be the trigonometric polynomial (23). Suppose $N \geq 2n+1$, W is defined as in (25), and $W_1 \in \mathbf{C}^{N \times n}$ is the matrix formed by the first n columns of W_{DFT}.

THEOREM 1. *f is nonnegative everywhere if and only if there exists an $X \in \mathbf{H}^{n+1}$ such that*

$$x = W^{\mathrm{H}}\, \mathbf{diag}(WXW^{\mathrm{H}}), \qquad X \succeq 0. \tag{26}$$

The result follows directly from the following fact: two vectors $x \in \mathbf{R} \times \mathbf{C}^n$ and $u \in \mathbf{C}^{n+1}$ satisfy

$$x_0 + 2\Re(x_1 e^{-j\omega} + \cdots + x_n e^{-jn\omega}) = |u_0 + u_1 e^{-j\omega} + \cdots + u_n e^{-jn\omega}|^2 \tag{27}$$

for all ω if and only if

$$x = \frac{1}{N} W^{\mathrm{H}}\, \mathbf{diag}(Wuu^{\mathrm{H}}W^{\mathrm{H}}). \tag{28}$$

To see this, we simply note that the elements of $\mathbf{diag}(Wuu^{\mathrm{H}}W^{\mathrm{H}})$ are the right-hand side of (27) evaluated at $\omega = 2\pi k/N$ for $k = 0, 1, \ldots, N-1$. As we observed in section 3.2, the inverse DFT of this vector gives the (unique) coefficients of the trigonometric polynomial that assumes those specified values. Therefore, the coefficients x defined in (27) are given by (28). Since every nonnegative trigonometric polynomial can be expressed as (27), (26) holds with $X = (1/N)uu^{\mathrm{H}}$.

Conversely, if (26) holds, then by factoring X as $X = (1/N)\sum_{k=0}^{n} u_k u_k^{\mathrm{H}}$, with $u_k = (u_{k0}, u_{k1}, \dots, u_{kn})$, we express f in the form

$$f(\omega) = \sum_{k=0}^{n} |u_{k0} + u_{k1}e^{-j\omega} + \cdots + u_{kn}e^{-jn\omega}|^2,$$

which shows $f(\omega) \geq 0$. This completes the proof of Theorem 1.

THEOREM 2. *f is nonnegative on $[\alpha - \beta, \alpha + \beta]$, where $0 < \beta < \pi$ if and only if there exist $X_1 \in \mathbf{H}^{n+1}$, $X_2 \in \mathbf{H}^n$ such that*

$$x = W^{\mathrm{H}}\big(\mathbf{diag}\left(WX_1W^{\mathrm{H}}\right) + d \circ \mathbf{diag}\left(W_1X_2W_1^{\mathrm{H}}\right)\big), \quad X_1 \succeq 0, \quad X_2 \succeq 0, \tag{29}$$

where $d \in \mathbf{R}^N$ has elements $d_k = \cos(2\pi k/N - \alpha) - \cos\beta$ for $k = 0, \dots, N-1$.

The proof of this theorem is similar to the proof of Theorem 1. We have

$$\begin{aligned} & x_0 + 2\Re(x_1e^{-j\omega} + \cdots + x_ne^{-jn\omega}) \\ & = \left|\sum_{k=0}^{n} u_k e^{-jk\omega}\right|^2 + (\cos(\omega - \alpha) - \cos\beta)\left|\sum_{k=0}^{n-1} v_k e^{-jk\omega}\right|^2 \end{aligned} \tag{30}$$

for all ω if and only if

$$x = \frac{1}{N}W^{\mathrm{H}}\big(\mathbf{diag}(Wuu^{\mathrm{H}}W^{\mathrm{H}}) + d \circ \mathbf{diag}\left(W_1vv^{\mathrm{H}}W_1^{\mathrm{H}}\right)\big).$$

According to the extension of the Riesz–Fejér theorem mentioned in section 3.1, if f is nonnegative on $[\alpha - \beta, \alpha + \beta]$, then it can be represented as (30), so (29) holds with $X_1 = (1/N)uu^{\mathrm{H}}$, $X_2 = (1/N)vv^{\mathrm{H}}$. Conversely, if (29) holds, then f can be expressed as a sum of functions of the form (30), so it is clearly nonnegative on $[\alpha - \beta, \alpha + \beta]$. This proves Theorem 2.

The constraint (26) is better known in a different form [14, 4, 11]. Let E_i be the ith "shift" matrix, i.e., $E_i \in \mathbf{R}^{(n+1)\times(n+1)}$ with elements

$$E_{i,kl} = \begin{cases} 1, & k - l = i, \\ 0, & \text{otherwise.} \end{cases}$$

It is easily seen that $E_i = (1/N)W^{\mathrm{H}}\,\mathbf{diag}(w_i)W$, where W and w_i are defined in (25) with $N \geq 2n + 1$. Therefore, (26) holds if and only if

$$x_i = w_i^{\mathrm{H}}\,\mathbf{diag}(WXW^{\mathrm{H}}) = \mathbf{tr}\left(\mathbf{diag}(w_i)^{\mathrm{H}}WXW^{\mathrm{H}}\right) = N\,\mathbf{tr}\left(E_i^{\mathrm{T}}X\right) = N\sum_{k-l=i} X_{kl}.$$

Hence the linear mapping $\mathcal{H} : \mathbf{H}^{n+1} \to \mathbf{R} \times \mathbf{C}^n$ defined by

$$\mathcal{H}(X) = \frac{1}{N}W^{\mathrm{H}}\,\mathbf{diag}(WXW^{\mathrm{H}}) \tag{31}$$

can also be expressed as

$$\mathcal{H}(X) = \big(\mathbf{tr}\left(E_0^{\mathrm{T}}X\right), \mathbf{tr}\left(E_1^{\mathrm{T}}X\right), \dots, \mathbf{tr}\left(E_n^{\mathrm{T}}X\right)\big). \tag{32}$$

We obtain the well-known result that $f(\omega) \geq 0$ if and only if there exists an $X \succeq 0$ such that $x_i = \sum_{k-l=i} X_{kl}$ for $i = 0, \dots, n$.

The adjoint of $\mathcal{H}$ (with respect to the inner products $\Re(x^{\mathrm{H}}z)$ on $\mathbf{R} \times \mathbf{C}^n$ and $\mathbf{tr}(XZ)$ on $\mathbf{H}^{n+1}$) can be derived using either one of the two expressions for $\mathcal{H}$. From (32),

$$\mathcal{H}^{\mathrm{adj}}(z) = \frac{1}{2}\begin{bmatrix} 2z_0 & \bar{z}_1 & \cdots & \bar{z}_n \\ z_1 & 2z_0 & \cdots & \bar{z}_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ z_n & z_{n-1} & \cdots & 2z_0 \end{bmatrix}, \tag{33}$$

the Hermitian Toeplitz matrix with first column $(z_0, z_1/2, \ldots, z_n/2)$. From (31),

$$\begin{aligned} \Re(z^{\mathrm{H}}\mathcal{H}(X)) &= \frac{1}{N}\Re(z^{\mathrm{H}}W^{\mathrm{H}}\,\mathbf{diag}(WXW^{\mathrm{H}})) \\ &= \frac{1}{N}\Re\big(\,\mathbf{tr}\,\big(\mathbf{diag}(Wz)^{\mathrm{H}}WXW^{\mathrm{H}}\big)\big) \\ &= \frac{1}{N}\,\mathbf{tr}((W^{\mathrm{H}}\,\mathbf{diag}(\Re(Wz))W)X), \end{aligned}$$

so

$$\mathcal{H}^{\mathrm{adj}}(z) = \frac{1}{N}W^{\mathrm{H}}\,\mathbf{diag}(\Re(Wz))W.$$

Although it is not immediately clear that this is equal to the Toeplitz matrix (33), it is sufficient to note that the convolution of z with an arbitrary $y \in \mathbf{C}^{n+1}$ is given by

$$\frac{1}{N}W^{\mathrm{H}}((Wz)\circ(Wy)) = \frac{1}{N}W^{\mathrm{H}}\,\mathbf{diag}(Wz)Wy.$$

The matrix $(1/N)W^{\mathrm{H}}\,\mathbf{diag}(Wz)W$ is, therefore, the lower triangular Toeplitz matrix with $(z_0, z_1, \ldots, z_n)$ as its first column. Adding the complex conjugate transpose and dividing by 2 gives

$$\frac{1}{2N}W^{\mathrm{H}}\big(\,\mathbf{diag}(Wz) + \mathbf{diag}(Wz)^{\mathrm{H}}\big)W = \frac{1}{N}W^{\mathrm{H}}\,\mathbf{diag}(\Re(Wz))W,$$

so this is indeed the Hermitian Toeplitz matrix with first column $(z_0, z_1/2, \ldots, z_n/2)$.

4. Cosine polynomials. In this section we consider semidefinite formulations of the constraint

$$f(\omega) = x_0 + x_1\cos\omega + \cdots + x_n\cos n\omega \geq 0, \qquad \omega \in [\alpha, \beta],$$

where $x \in \mathbf{R}^{n+1}$ and $0 \leq \alpha < \beta \leq \pi$. This is in fact a special case of the constraints considered in the previous section, since f is a trigonometric polynomial with real coefficients. For example, using Theorem 1, we can say that $f(\omega) \geq 0$ for all ω if and only if

$$(x_0, x_1/2, \ldots, x_n/2) = W^{\mathrm{H}}\,\mathbf{diag}(WXW^{\mathrm{H}})$$

for some $X \succeq 0$, where $N \geq 2n+1$ and W is formed by the first $n+1$ columns of the length-N DFT matrix. The purpose of this section is to show that simpler semidefinite parametrizations, using smaller matrices, can be obtained for cosine polynomials.

4.1. Sum-of-squares characterizations. Let f be a cosine polynomial of degree n, i.e.,

$$f(\omega) = x_0 + x_1 \cos\omega + \cdots + x_n \cos n\omega, \tag{34}$$

with $x \in \mathbf{R}^{n+1}$ and $x_n \neq 0$. If f is nonnegative on $[\alpha, \beta]$, where $0 \leq \alpha < \beta \leq \pi$, then it can be expressed as

$$f(\omega) = \begin{cases} g(\omega)^2 + (\cos\omega - \cos\beta)(\cos\alpha - \cos\omega)h(\omega)^2, & n \text{ even}, \\ (\cos\omega - \cos\beta)g(\omega)^2 + (\cos\alpha - \cos\omega)h(\omega)^2, & n \text{ odd}, \end{cases}$$

where g and h are cosine polynomials with $\deg(g) \leq \lfloor n/2 \rfloor$, $\deg(h) \leq \lfloor (n-1)/2 \rfloor$. This result can be derived from the characterization of nonnegative polynomials on $[-1, 1]$ (see section 5.1) by making a change of variables $t = \cos\omega$.

If $\alpha = 0$, $\beta = \pi$, i.e., f is nonnegative everywhere, these expressions can be simplified. If $n = 2m$, we have

$$\begin{aligned} f(\omega) &= g(\omega)^2 + (1 - \cos^2\omega)h(\omega)^2 \\ &= g(\omega)^2 + (\sin\omega)^2 h(\omega)^2 \\ &= g(\omega)^2 + \tilde{h}(\omega)^2, \end{aligned} \tag{35}$$

where $\tilde{h}$ is of the form $\tilde{h}(\omega) = v_1 \sin\omega + v_2 \sin 2\omega + \cdots + v_m \sin m\omega$. This follows from the fact that the function $\sin k\omega / \sin\omega$ is a cosine polynomial of degree $k - 1$.

If $n = 2m + 1$, we have

$$\begin{aligned} f(\omega) &= (\cos\omega + 1)g(\omega)^2 + (1 - \cos\omega)h(\omega)^2 \\ &= 2(\cos(\omega/2))^2 g(\omega)^2 + 2(\sin(\omega/2))^2 h(\omega)^2 \\ &= \tilde{g}(\omega)^2 + \tilde{h}(\omega)^2, \end{aligned} \tag{36}$$

where $\tilde{g}$ and $\tilde{h}$ have the form

$$\tilde{g}(\omega) = \sum_{k=0}^{m} u_k \cos((k + 1/2)\omega), \qquad \tilde{h}(\omega) = \sum_{k=0}^{m} v_k \sin((k + 1/2)\omega).$$

This follows from the fact that $\cos((k+1/2)\omega)/\cos(\omega/2)$ and $\sin((k+1/2)\omega)/\sin(\omega/2)$ are cosine polynomials of degree k.

4.2. Discrete cosine transform. The matrices

$$V_{\mathrm{DCT}} = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ 1 & \cos(\pi/N) & \cdots & \cos((N-1)\pi/N) & \cos(\pi) \\ 1 & \cos(2\pi/N) & \cdots & \cos(2(N-1)\pi/N) & \cos(2\pi) \\ \vdots & \vdots & & \vdots & \vdots \\ 1 & \cos(\pi) & \cdots & \cos((N-1)\pi) & \cos(N\pi) \end{bmatrix} \in \mathbf{S}^{N+1}$$

and

$$W_{\mathrm{DCT}} = \frac{2}{N} D V_{\mathrm{DCT}} D,$$

where $D = \mathbf{diag}(1/2, 1, 1, \ldots, 1, 1, 1/2)$, are inverses:

$$W_{\mathrm{DCT}} V_{\mathrm{DCT}} = I \tag{37}$$

(see, for example, [7, p. 124]). The mapping $V_{\mathrm{DCT}}Du$ is sometimes referred as the discrete cosine transform (DCT) of u.

Suppose $N \geq n$, and let W and V be the matrices formed by taking the first $n+1$ columns of W_{DCT} and V_{DCT}, respectively. These matrices satisfy $W^{\mathrm{T}}V = I$ as a consequence of (37) and the symmetry of W_{DCT}. The matrix V maps the coefficients $x_0, \ldots, x_n$ of the cosine polynomial (34) to its values at $\omega = k\pi/N$, $k = 0, \ldots, N$. Multiplying with W^{T} maps these sample values to the coefficients. In other words, if $y = (f(0), f(\pi/N), \ldots, f((N-1)\pi/N), f(\pi))$, then

$$y = Vx, \qquad x = W^{\mathrm{T}}y.$$

4.3. Semidefinite representations. We now use the DCT and the sum-of-squares theorems in section 4.1 to express constraints on a cosine polynomial

$$f(\omega) = x_0 + x_1 \cos\omega + \cdots + x_n \cos n\omega$$

in semidefinite form. Assume $N \geq n$ and define $\omega_N = \pi/N$. As in section 4.2, $W \in \mathbf{R}^{(N+1)\times(n+1)}$ denotes the matrix formed with the first $n+1$ columns of W_{DCT}.

Theorem 3. *$f(\omega) \geq 0$ on $[\alpha, \beta]$ if and only if there exist $X_1 \in \mathbf{S}^{m_1+1}$ and $X_2 \in \mathbf{S}^{m_2+1}$ such that*

$$x = W^{\mathrm{T}}\left(d_1 \circ \mathbf{diag}\left(V_1 X_1 V_1^{\mathrm{T}}\right) + d_2 \circ \mathbf{diag}\left(V_2 X_2 V_2^{\mathrm{T}}\right)\right), \quad X_1 \succeq 0, \quad X_2 \succeq 0, \tag{38}$$

where $m_1 = \lfloor n/2 \rfloor$, $m_2 = \lfloor (n-1)/2 \rfloor$, and $d_1, d_2 \in \mathbf{R}^{N+1}$ are defined as

$$d_{1,k} = \begin{cases} 1, & n \text{ even}, \\ \cos k\omega_N - \cos\beta, & n \text{ odd}, \end{cases}$$

$$d_{2,k} = \begin{cases} (\cos k\omega_N - \cos\beta)(\cos\alpha - \cos k\omega_N), & n \text{ even}, \\ \cos\alpha - \cos k\omega_N, & n \text{ odd} \end{cases}$$

for $k = 0, \ldots, N$. The columns of $V_1 \in \mathbf{R}^{(N+1)\times(m_1+1)}$ and $V_2 \in \mathbf{R}^{(N+1)\times(m_2+1)}$ are the first m_1+1, respectively, m_2+1, columns of V_{DCT}.

We prove the theorem for n even ($n = 2m$). By the sum-of-squares characterization in section 4.1, if f is nonnegative on $[\alpha, \beta]$, then it can be expressed as

$$f(\omega) = g(\omega)^2 + (\cos\omega - \cos\beta)(\cos\alpha - \cos\omega)h(\omega)^2 \tag{39}$$

for some cosine polynomials

$$g(\omega) = \sum_{k=0}^{m} u_k \cos k\omega, \qquad h(\omega) = \sum_{k=0}^{m-1} v_k \cos k\omega.$$

From section 4.2, we can express the right-hand side of (39) as a cosine polynomial by computing the values at $\omega = k\pi/N$, $k = 0, \ldots, N$, which gives the vectors

$$d_1 \circ \mathbf{diag}\left(V_1 u u^{\mathrm{T}} V_1^{\mathrm{T}}\right) + d_2 \circ \mathbf{diag}\left(V_2 v v^{\mathrm{T}} V_2^{\mathrm{T}}\right),$$

and then multiplying on the left with W^{T}. In other words, (39) is equivalent to

$$x = W^{\mathrm{T}}\left(d_1 \circ \mathbf{diag}\left(V_1 u u^{\mathrm{T}} V_1^{\mathrm{T}}\right) + d_2 \circ \mathbf{diag}\left(V_2 v v^{\mathrm{T}} V_2^{\mathrm{T}}\right)\right).$$

Therefore, (38) holds with $X_1 = uu^{\mathrm{T}}$ and $X_2 = vv^{\mathrm{T}}$. Conversely, if (38) holds, with X_1 and X_2 of rank greater than 2, then f is a sum of cosine polynomials that are nonnegative on $[\alpha, \beta]$, so it is also nonnegative.

If $\alpha = 0$ and $\beta = \pi$, we can start from (35) and (36) and express the semidefinite constraints in a slightly simpler form.

THEOREM 4. *$f(\omega) \geq 0$ everywhere if and only if there exist $X_1 \in \mathbf{S}^{m_1+1}$, $X_2 \in \mathbf{S}^{m_2+1}$ such that*

$$x = W^{\mathrm{T}}\big(\mathbf{diag}\left(V_1 X_1 V_1^{\mathrm{T}}\right) + \mathbf{diag}\left(V_2 X_2 V_2^{\mathrm{T}}\right)\big), \quad X_1 \succeq 0, \quad X_2 \succeq 0, \tag{40}$$

where $m_1 = \lfloor n/2 \rfloor$, $m_2 = \lfloor (n-1)/2 \rfloor$. If n is even, we define $V_1 \in \mathbf{R}^{(N+1)\times(m_1+1)}$ as the matrix formed by the first $m_1 + 1$ columns of V_{DCT} and

$$V_2 = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \sin(\omega_N) & \sin(2\omega_N) & \cdots & \sin(m\omega_N) \\ \sin(2\omega_N) & \sin(4\omega_N) & \cdots & \sin(2m\omega_N) \\ \vdots & \vdots & & \vdots \\ \sin(N\omega_N) & \sin(2N\omega_N) & \cdots & \sin(mN\omega_N) \end{bmatrix} \in \mathbf{R}^{(N+1)\times(m_2+1)}.$$

If n is odd, we define V_1 and V_2 as

$$V_1 = \begin{bmatrix} 1 & \cdots & 1 \\ \cos(\omega_N/2) & \cdots & \cos((m+1/2)\omega_N) \\ \cos(\omega_N) & \cdots & \cos(2(m+1/2)\omega_N) \\ \vdots & & \vdots \\ \cos(N\omega_N/2) & \cdots & \cos(N(m+1/2)\omega_N) \end{bmatrix} \in \mathbf{R}^{(N+1)\times(m_1+1)},$$

$$V_2 = \begin{bmatrix} 0 & \cdots & 0 \\ \sin(\omega_N/2) & \cdots & \sin((m+1/2)\omega_N) \\ \sin(\omega_N) & \cdots & \sin(2(m+1/2)\omega_N) \\ \vdots & & \vdots \\ \sin(N\omega_N/2) & \cdots & \sin(N(m+1/2)\omega_N) \end{bmatrix} \in \mathbf{R}^{(N+1)\times(m_2+1)}.$$

Note that the matrices X_1 and X_2 in the constraints (38) and (40) have dimension roughly $n/2$, as opposed to the constraints for general trigonometric polynomials of degree n given in section 3, which involve matrix variables of size n. It is also interesting to note that the matrices V_1, V_2, and W are orthogonal or nearly orthogonal (i.e., have a condition number close to 1).

4.4. Example: Linear-phase Nyquist filter.

We consider the lowpass filter design problem

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & -t \leq H(\omega) \leq t, \quad \omega_{\mathrm{s}} \leq \omega \leq \pi, \end{array} \tag{41}$$

in which H is the frequency response of a linear-phase *Nyquist-M* filter [32, section 4.6]:

$$H(\omega) = h_0 + h_1 \cos\omega + \cdots + h_n \cos n\omega$$

with

$$h_0 = 1/M, \quad h_{kM} = 0, \quad k = 1, 2, \ldots, \lfloor n/M \rfloor. \tag{42}$$

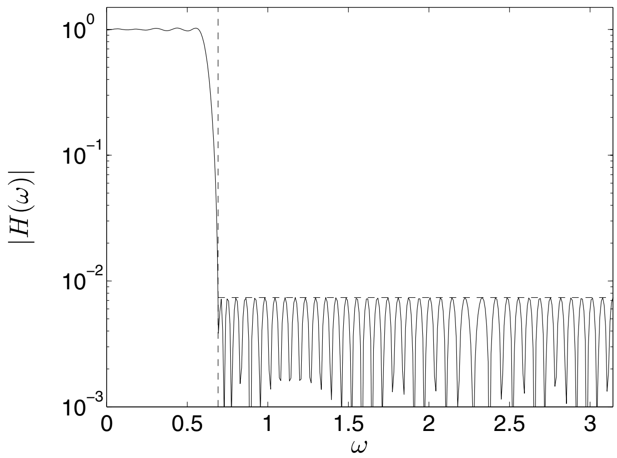


FIG. 1. *Frequency response of a linear-phase Nyquist-*5 *filter of length* 51 *and stopband edge* $\omega_{\rm s} = 1.1\pi/5 = 0.69$.

The variables in (41) are t and the $n - \lfloor n/M \rfloor$ coefficients h_i that are not determined by (42). Since H is a cosine polynomial, we can apply Theorem 3 to formulate this problem as an SDP,

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & h + te_0 = W^{\rm T}\big(d_1 \circ \mathbf{diag}\,\big(V_1 X_1 V_1^{\rm T}\big) + d_2 \circ \mathbf{diag}\,\big(V_2 X_2 V_2^{\rm T}\big)\big), \\ & -h + te_0 = W^{\rm T}\big(d_1 \circ \mathbf{diag}\,1\big(V_1 X_3 V_1^{\rm T}\big) + d_2 \circ \mathbf{diag}\,\big(V_2 X_4 V_2^{\rm T}\big)\big), \\ & X_1 \succeq 0, \quad X_2 \succeq 0, \quad X_3 \succeq 0, \quad X_4 \succeq 0, \end{array} \tag{43}$$

where $e_0 = (1, 0, \ldots, 0) \in \mathbf{R}^{n+1}$ and W, d_1, d_2, V_1, V_2 are defined as in Theorem 3 with $\alpha = \omega_{\rm s}$, $\beta = \pi$. The variables are t, the $n - \lfloor n/M \rfloor$ unknown entries of $h = (h_0, h_1, \ldots, h_n)$, and four symmetric matrices X_i, which have dimension roughly $n/2$. Figure 1 shows an example with $n = 50$, $M = 5$, $\omega_{\rm s} = 1.1\pi/M$.

5. Real polynomials.

5.1. Sum-of-squares characterizations.

Let f be a polynomial of degree n with real coefficients. If f is nonnegative on $\mathbf{R}$, then n is even and f can be expressed as

$$f(t) = g(t)^2 + h(t)^2, \tag{44}$$

where $\deg(g) \leq n/2$ and $\deg(h) \leq n/2$. If f is nonnegative on $[a, \infty)$, then f can be expressed as

$$f(t) = g(t)^2 + (t - a)h(t)^2,$$

where $\deg(g) \leq \lfloor n/2 \rfloor$ and $\deg(h) \leq \lfloor (n-1)/2 \rfloor$. Finally, if f is nonnegative on $[a, b]$, where $a < b$, then it can be expressed as

$$f(t) = \begin{cases} g(t)^2 + (t-a)(b-t)h(t)^2, & n \text{ even}, \\ (t-a)g(t)^2 + (b-t)h(t)^2, & n \text{ odd}, \end{cases} \tag{45}$$

where g and h are polynomials with $\deg(g) \leq \lfloor n/2 \rfloor$ and $\deg(h) \leq \lfloor (n-1)/2 \rfloor$. This last result is known as the *Markov–Lukács theorem* [29, section 1.21], [20, section 3.2].

5.2. Discrete polynomial transforms. Let $p_k(t)$, $k = 0, 1, \ldots$, be a system of orthogonal and normalized polynomials on a bounded or unbounded interval $I \subseteq \mathbf{R}$ with respect to a nonnegative weight function $w(t)$:

$$\int_I p_k(t)p_l(t)w(t)\,dt = \begin{cases} 0, & k \neq l, \\ 1, & k = l. \end{cases}$$

The kth polynomial p_k has degree k with a positive leading coefficient a_k. It is well known that orthogonal polynomials satisfy a three-term recursion

$$p_{k+1}(t) = (\alpha_k t - \beta_k)p_k(t) - \gamma_k p_{k-1}(t), \tag{46}$$

where we define $p_{-1}(t) = 0$. The coefficients α_k, γ_k are positive and satisfy

$$\alpha_k = \frac{a_{k+1}}{a_k} > 0, \qquad \frac{\alpha_k \gamma_{k+1}}{\alpha_{k+1}} = 1. \tag{47}$$

The recursion (46) for $k = 0, \ldots, N$ can be written in matrix-vector form as

$$tp(t) = Jp(t) + (1/\alpha_N)p_{N+1}(t)e_N, \tag{48}$$

where $p(t) = (p_0(t), p_1(t), \ldots, p_N(t))$, $e_N = (0, 0, \ldots, 0, 1) \in \mathbf{R}^{N+1}$, and

$$J = \begin{bmatrix} \beta_0/\alpha_0 & 1/\alpha_0 & 0 & \cdots & 0 & 0 \\ \gamma_1/\alpha_1 & \beta_1/\alpha_1 & 1/\alpha_1 & \cdots & 0 & 0 \\ 0 & \gamma_2/\alpha_2 & \beta_2/\alpha_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \beta_{N-1}/\alpha_{N-1} & 1/\alpha_{N-1} \\ 0 & 0 & 0 & \cdots & \gamma_N/\alpha_N & \beta_N/\alpha_N \end{bmatrix}.$$

It follows from (47) that J is symmetric. Another well-known property of orthogonal polynomials is that p_k has exactly k distinct roots in the interior of I [10, p. 236]. From (48) we see that this implies

$$\lambda_i p(\lambda_i) = Jp(\lambda_i), \qquad i = 0, \ldots, N,$$

where λ_0, λ_1, $\ldots$, λ_N are the roots of p_{N+1}. In other words $p(\lambda_i)$ is an eigenvector of J with eigenvalue λ_i [15].

These properties provide an efficient method for computing the matrix

$$V_{\mathrm{DPT}} = \begin{bmatrix} p_0(\lambda_0) & p_1(\lambda_0) & \cdots & p_N(\lambda_0) \\ p_0(\lambda_1) & p_1(\lambda_1) & \cdots & p_N(\lambda_1) \\ \vdots & \vdots & & \vdots \\ p_0(\lambda_N) & p_1(\lambda_N) & \cdots & p_N(\lambda_N) \end{bmatrix} \in \mathbf{R}^{(N+1)\times(N+1)}$$

directly from the coefficients α_k, β_k, γ_k in the recursion (46). Let

$$J = Q\,\mathbf{diag}(\lambda)Q^{\mathrm{T}}$$

be the eigenvalue decomposition of J with normalized eigenvectors ($QQ^{\mathrm{T}} = Q^{\mathrm{T}}Q = I$) and the signs in the first row of Q chosen to be positive. The ith column of Q is then a positive multiple of $p(\lambda_i)$, and therefore

$$V_{\mathrm{DPT}} = DQ^{\mathrm{T}}$$

with D positive diagonal. The matrix D is easily determined by dividing the first column of V_{DPT}, which is a constant $p_0(t) = (\int w(t)\,dt)^{-1/2}$, by the elements in the first row q_1^{T} of Q: $D = p_0(t)\,\mathbf{diag}(q_1)^{-1}$. It follows that

$$V_{\mathrm{DPT}}^{\mathrm{T}} D^{-2} V_{\mathrm{DPT}} = I,$$

so the matrix

$$W_{\mathrm{DPT}} = D^{-1}Q^{\mathrm{T}} = D^{-2}V_{\mathrm{DPT}} \tag{49}$$

satisfies $W_{\mathrm{DPT}}^{\mathrm{T}} V_{\mathrm{DPT}} = I$. The matrices V_{DPT} and W_{DPT} thus define a pair of forward and inverse "discrete polynomial transforms" [7, section 8.5].

Now suppose $N \geq n$, and let W and V be the matrices formed by the first $n+1$ columns of W_{DPT} and V_{DPT}. Since V_{DPT} and $W_{\mathrm{DPT}}^{\mathrm{T}}$ are inverses, we have $W^{\mathrm{T}}V = I$. The linear transformations Vx and $W^{\mathrm{T}}y$ map the coefficients of the polynomial

$$f(t) = x_0p_0(t) + x_1p_1(t) + \cdots + x_np_n(t)$$

to $N+1$ values at $\lambda_0, \ldots, \lambda_N$ and vice versa: If

$$y = (f(\lambda_0), f(\lambda_1), \ldots, f(\lambda_N)),$$

then $y = Vx$ and $x = W^{\mathrm{T}}y$.

5.3. Semidefinite representations. We can apply the discrete transform associated with the orthogonal polynomials p_k, combined with the sum-of-squares results in section 5.1, to derive LMI conditions for nonnegativity of the polynomial

$$f(t) = x_0p_0(t) + x_1p_1(t) + \cdots + x_np_n(t).$$

Assume $N \geq n$. Let $W \in \mathbf{R}^{(N+1)\times(n+1)}$ be the matrix formed by the first $n+1$ columns of W_{DPT} in (49), and let $\lambda = (\lambda_0, \lambda_1, \ldots, \lambda_N)$ be the vector of zeros of p_{N+1}.

THEOREM 5. *$f(t) \geq 0$ for $t \in \mathbf{R}$ if and only if n is even and there exists an $X \in \mathbf{S}^{n/2+1}$ such that*

$$x = W^{\mathrm{T}}\,\mathbf{diag}\left(V_1XV_1^{\mathrm{T}}\right), \qquad X \succeq 0.$$

Here V_1 is the matrix formed by the first $n/2+1$ columns of V_{DPT}.

THEOREM 6. *$f(t) \geq 0$ on $[a,\infty)$ if and only if there exist $X_1 \in \mathbf{S}^{m_1+1}$ and $X_2 \in \mathbf{S}^{m_2+1}$ such that*

$$x = W^{\mathrm{T}}\left(\mathbf{diag}\left(V_1X_1V_1^{\mathrm{T}}\right) + (\lambda - a)\circ\mathbf{diag}\left(V_2X_2V_2^{\mathrm{T}}\right)\right), \quad X_1 \succeq 0, \quad X_2 \succeq 0.$$

Here $m_1 = \lfloor n/2 \rfloor$, $m_2 = \lfloor (n-1)/2 \rfloor$, and V_1 and V_2 are the matrices formed by the first m_1+1, respectively, m_2+1, columns of V_{DPT}.

THEOREM 7. *$f(t) \geq 0$ on $[a,b]$ if and only if there exist $X_1 \in \mathbf{S}^{m_1+1}$, $X_2 \in \mathbf{S}^{m_2+1}$ such that*

$$x = W^{\mathrm{T}}\left(d_1\circ\mathbf{diag}(V_1X_1V_1^{\mathrm{T}}) + d_2\circ\mathbf{diag}(V_2X_2V_2^{\mathrm{T}})\right), \quad X_1 \succeq 0, \quad X_2 \succeq 0.$$

Here $m_1 = \lfloor n/2 \rfloor$, $m_2 = \lfloor (n-1)/2 \rfloor$, and V_1 and V_2 are the matrices formed by the first m_1+1, respectively, m_2+1, columns of V_{DPT}. The vectors $d_1, d_2 \in \mathbf{R}^{N+1}$ are defined as

$$d_1 = \begin{cases} \mathbf{1}, & n \textit{ even}, \\ \lambda - a\mathbf{1}, & n \textit{ odd}, \end{cases} \qquad d_2 = \begin{cases} (\lambda - a\mathbf{1})\circ(b\mathbf{1} - \lambda), & n \textit{ even}, \\ b\mathbf{1} - \lambda, & n \textit{ odd}. \end{cases}$$

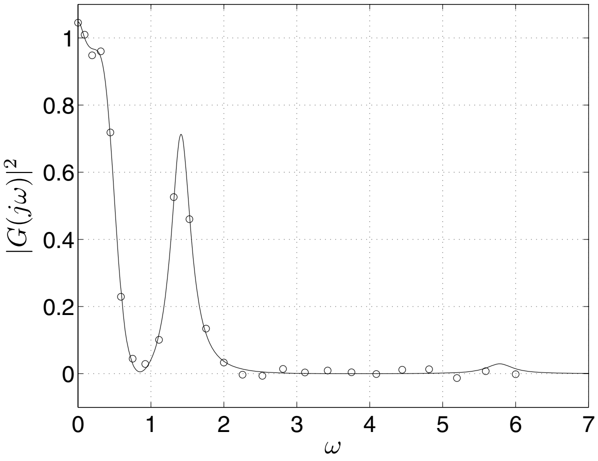


FIG. 2. *Minimax magnitude fit of a rational transfer function to* 25 *data points.*

The proofs follow exactly the same pattern as in sections 3.3 and 4.3, and are omitted.

There exist several other interesting choices for the matrices V_1, V_2, and W. First, we can define V_1 and V_2 as the first columns of the matrix Q^{T} (instead of the first columns of $V_{\mathrm{DPT}} = DQ^{\mathrm{T}}$) if we change the definition of W accordingly and construct W from the first columns of DQ^{T}. With this choice, V_1 and V_2 are orthogonal. Alternatively, we can define W to be the first columns of the matrix Q^{T}, and redefine V_1 and V_2 as the first columns of $D^{1/2}Q^{\mathrm{T}}$. With this choice W is orthogonal.

Second, we can note that the basis polynomials used in the definitions of V_1 and V_2 need not be the same as in the definition of W. This follows from the fact that in (44)–(45), we can use a different basis to represent the polynomials f, g, and h. We could, therefore, define V_1 and V_2 as generalized Vandermonde matrices with k, l elements $q_l(t_k)$, where t_k are the zeros of p_N, and q_0, q_1, ... is any polynomial basis. This is equivalent to replacing the matrices V_k by V_kT_k, where T_k is nonsingular. In particular, we can replace V_1 and V_2 with orthogonal matrices that have the same column spaces.

5.4. Example: Minimax magnitude fit of rational transfer function.

We consider the problem of fitting the magnitude of a rational transfer function

$$G(s) = \frac{a_0 + a_1 s + \cdots + a_n s^n}{b_0 + b_1 s + \cdots + b_m s^m}$$

to data points, i.e., choosing the (real) coefficients a_i, b_i so that $|G(j\omega_k)|^2 \approx \gamma_k$ for $k = 1, \ldots, K$, where ω_k and γ_k are given. Using a minimax criterion and introducing an auxiliary variable δ we can formulate this problem as

$$\begin{array}{ll} \text{minimize} & \delta \\ \text{subject to} & -\delta \le |G(j\omega_k)|^2 - \gamma_k \le \delta, \quad k = 1, \ldots, K. \end{array}$$

Figure 2 shows an example with $n = 6$, $m = 8$, and $K = 25$.

This problem can be posed as a quasi-convex optimization problem. We first express the magnitude squared of the transfer function as

$$|G(j\omega)|^2 = \frac{f_1(\omega^2)}{f_2(\omega^2)},$$

where f_1 and f_2 are the real polynomials,

$$f_1(t) = a_{\rm e}(t)^2 + ta_{\rm o}(t)^2, \qquad f_2(t) = b_{\rm e}(t)^2 + tb_{\rm o}(t)^2 \tag{50}$$

with

$$a_{\rm e}(t) = \sum_{k=0}^{\lfloor n/2 \rfloor} a_{2k}(-t)^k, \qquad a_{\rm o}(t) = \sum_{k=0}^{\lfloor (n-1)/2 \rfloor} a_{2k+1}(-t)^k,$$

$$b_{\rm e}(t) = \sum_{k=0}^{\lfloor m/2 \rfloor} b_{2k}(-t)^k, \qquad b_{\rm o}(t) = \sum_{k=0}^{\lfloor (m-1)/2 \rfloor} b_{2k+1}(-t)^k.$$

Clearly $f_1(t) \geq 0$ and $f_2(t) \geq 0$ for $t \geq 0$. Conversely, if f_1 and f_2 are nonnegative on the nonnegative real axis, then by the result mentioned in section 5.1, they can be expressed as (50). The fitting problem is therefore equivalent to

$$\begin{array}{ll} \text{minimize} & \delta \\ \text{subject to} & (\gamma_k - \delta) f_2(\omega_k^2) \leq f_1(\omega_k^2) \leq (\gamma_k + \delta) f_2(\omega_k^2), \quad k = 1, \ldots, K, \\ & f_1(t) \geq 0, \quad f_2(t) \geq 0 \quad \text{for } t \geq 0. \end{array} \tag{51}$$

The variables are δ and the coefficients of the polynomials

$$f_1(t) = x_0 p_0(t) + x_1 p_1(t) + \cdots + x_n p_n(t), \qquad f_2(t) = p_0(t) + y_1 p_1(t) + \cdots + y_m p_m(t)$$

for some choice of orthogonal basis polynomials $p_k(t)$. We normalize the first coefficient of f_2 to rule out the trivial solution $f_1(t) = f_2(t) = 0$. (Alternatively, one might prefer to replace $f_2(t) \geq 0$ with $f_2(t) \geq \epsilon$ for some small positive ϵ, which would also ensure that there are no poles on the imaginary axis.)

Problem (51) can be solved via bisection on δ. In each bisection step we fix δ and determine whether the constraints are feasible or not. This feasibility problem can be cast as an SDP feasibility problem,

$$\begin{array}{c} (\gamma_k - \delta) f_2(\omega_k^2) \leq f_1(\omega_k^2) \leq (\gamma_k + \delta) f_2(\omega_k)^2, \quad k = 1, \ldots, K, \\ x = W^{\rm T} \left(\mathbf{diag}\left(V_1 X_1 V_1^{\rm T} \right) + \lambda \circ \mathbf{diag}\left(V_2 X_2 V_2^{\rm T} \right) \right), \\ y = \tilde{W}^{\rm T} \big(\mathbf{diag}\left(\tilde{V}_1 \tilde{X}_1 \tilde{V}_1^{\rm T} \right) + \tilde{\lambda} \circ \mathbf{diag}\left(\tilde{V}_2 \tilde{X}_2 \tilde{V}_2^{\rm T} \right) \big), \\ X_1 \succeq 0, \quad X_2 \succeq 0, \quad \tilde{X}_1 \succeq 0, \quad \tilde{X}_2 \succeq 0, \end{array} \tag{52}$$

where $x = (x_0, x_1, \ldots, x_n)$ and $y = (1, y_1, \ldots, y_m)$. The variables are x_k, y_k and the matrices X_i and $\tilde{X}_i$. The matrices W, V_1, V_2 and the vector λ are defined as in Theorem 6 with $a = 0$. The matrices $\tilde{W}$, $\tilde{V}_1$, $\tilde{V}_2$ and $\tilde{\lambda}$ are defined similarly but with n replaced by m.

6. Numerical examples.

The SDP characterizations of nonnegative polynomials derived in the previous sections can be expressed in the following common form. A (trigonometric, cosine, real) polynomial with coefficients x is nonnegative on a given interval if and only if there exist Hermitian matrices X_k such that

$$x = \sum_{k=1}^{s} A_k \,\mathbf{diag}(C_k X_k C_k^{\mathrm{H}}), \quad X_k \succeq 0, \quad k = 1, \dots, s.$$

In the case of cosine polynomials or real polynomials, the matrices A_k, C_k and the variables x and X_k are real. This representation allows us to formulate a wide variety of optimization problems involving polynomials as SDPs of the form

$$\begin{array}{ll} \text{minimize} & c^{\mathrm{T}} y \\ \text{subject to} & \displaystyle\sum_{k=1}^{s_i} A_{ik} \,\mathbf{diag}(C_{ik} X_{ik} C_{ik}^{\mathrm{H}}) + B_i y = b_i, \quad i = 1, \dots, L, \\ & X_{ik} \succeq 0, \quad k = 1, \dots, s_i, \quad i = 1, \dots, L. \end{array} \tag{53}$$

The variables are $y \in \mathbf{R}^p$ and the Hermitian matrices X_{ik}. Each of the L constraints expresses a nonnegativity condition on a polynomial with coefficients $b_i - B_i y$.

The SDP (53) is a special case of (11) or (19) if we interpret X as a block-diagonal matrix with diagonal blocks X_{ik}, and define A, C, and B as block matrices constructed from A_{ik}, C_{ik}, and B_i. In this section we present numerical results for a primal-dual interior-point method that uses the fast method for solving the Newton equations described in section 2.1. We first provide some details of the implementation.

6.1. Implementation.

All examples are instances of the SDP (53) with real data and variables. Applying the method of section 2.1 to an SDP with block-diagonal structure (53) leads to a reduced Newton system (18),

$$\begin{bmatrix} H_1 & 0 & \cdots & 0 & B_1 \\ 0 & H_2 & \cdots & 0 & B_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & H_L & B_L \\ B_1^{\mathrm{T}} & B_2^{\mathrm{T}} & \cdots & B_L^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \Delta z_1 \\ \Delta z_2 \\ \vdots \\ \Delta z_L \\ \Delta y \end{bmatrix} = \begin{bmatrix} r_{3,1} \\ r_{3,2} \\ \vdots \\ r_{3,L} \\ r_2 \end{bmatrix}, \tag{54}$$

where

$$H_i = \sum_{k=1}^{s_i} A_{ik} \,\mathbf{sqr}\left(C_{ik} T_{ik} C_{ik}^{\mathrm{T}}\right) A_{ik}^{\mathrm{T}}.$$

When solving (54), we can exploit the "block-arrow" structure by first eliminating the variables Δz_i and then solving a positive definite set of linear equations in the variables Δy:

$$\left(\sum_{i=1}^{L} B_i^{\mathrm{T}} H_i^{-1} B_i \right) \Delta y = \sum_{i=1}^{L} B_i^{\mathrm{T}} H_i^{-1} r_{3,i} - r_2. \tag{55}$$

From the solution Δy we obtain Δz_i by solving $H_i \Delta z_i = r_{3,i} - B_i \Delta y$.

In the numerical experiments described below we implemented this idea as follows. We compute the Hadamard products $\mathbf{sqr}(C_{ik} T_{ik} C_{ik}^{\mathrm{T}})$ and factor them as

$$\mathbf{sqr}\left(C_{ik} T_{ik} C_{ik}^{\mathrm{T}}\right) = V_{ik} V_{ik}^{\mathrm{T}}.$$

The eigenvalue decomposition is used for this purpose, since the matrix $\mathbf{sqr}(C_{ik}T_{ik}C_{ik}^{\mathrm{T}})$ is often rank-deficient. We then factor the matrices

$$H_i = \sum_{k=1}^{s_i} A_{ik}V_{ik}V_{ik}^{\mathrm{T}}A_{ik}^{\mathrm{T}}$$

as $H_i = R_i^{\mathrm{T}}R_i$ via QR factorizations of the matrices

$$\begin{bmatrix} A_{i1}V_{i1} & A_{i2}V_{i2} & \cdots & A_{ir_i}V_{ir_i} \end{bmatrix}^{\mathrm{T}} = Q_iR_i.$$

This is more stable than using a Cholesky factorization of H_i, since it allows us to compute the triangular factors R_i without explicitly forming H_i. Equation (55) now reduces to

$$\left(\sum_{i=1}^{L} B_i^{\mathrm{T}}R_i^{-1}R_i^{-T}B_i\right)\Delta y = \sum_{i=1}^{L} B_i^{\mathrm{T}}R_i^{-1}R_i^{-T}r_{3,i} - r_2.$$

To improve the numerical stability, we again avoid forming the coefficient matrix and use a QR factorization

$$\begin{bmatrix} B_1^{\mathrm{T}}R_1^{-1} & B_2^{\mathrm{T}}R_2^{-1} & \cdots & B_L^{\mathrm{T}}R_L^{-1} \end{bmatrix}^{\mathrm{T}} = QR$$

instead. Given Q and R we can find Δy by solving

$$R\Delta y = Q^{\mathrm{T}}\tilde{r}_3 - R^{-T}r_2,$$

where

$$\tilde{r}_3 = \begin{bmatrix} \left(R_1^{-T}r_{3,1}\right)^{\mathrm{T}} & \left(R_2^{-T}r_{3,2}\right)^{\mathrm{T}} & \cdots & \left(R_L^{-T}r_{3,L}\right)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}.$$

Except for the algorithm used for solving the Newton equations, the code is a rudimentary implementation of an SDPT3-style path-following method [30, 31], following the outline given in the appendix of [33]. Infeasible starting points are used: we take $y = 0$, $X_{ik} = I$ in the primal problem; in the dual problem

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{i=1}^{L} b_i^{\mathrm{T}}z_i \\ \text{subject to} & C_{ik}^{\mathrm{T}}\,\mathbf{diag}\left(A_{ik}^{\mathrm{T}}z_i\right)C_{ik} + Z_{ik} = 0, \quad Z_i \succeq 0, \quad k = 1,\ldots,s_i, \quad i = 1,\ldots,L, \\ & \displaystyle\sum_{i=1}^{L} B_i^{\mathrm{T}}z_i = c \end{array}$$

we take $Z_{ik} = I$ and for z_i the least-norm solution of the last equality constraint. The stopping criterion is based on the following quantities.

- The duality gap

$$\eta_{\mathrm{abs}} = \sum_{i=1}^{L}\sum_{k=1}^{s_i} \mathbf{tr}(X_{ik}Z_{ik}).$$

(This is only truly the duality gap when the primal and dual iterates are feasible.)

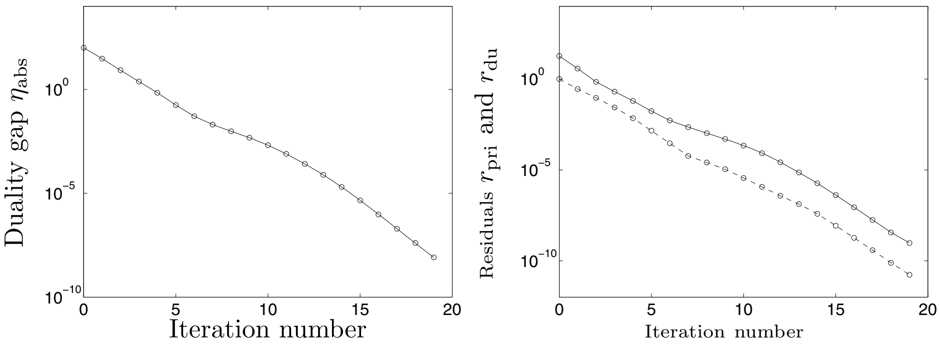


Fig. 3. *Progress of the primal-dual method for the design of a lowpass Nyquist-*5 *filter. The left plot shows the duality gap versus iteration number. The right plot shows the primal residual* (*solid line*) *and the dual residual* (*dashed line*).

- The relative duality gap

$$\eta_{\text{rel}} = \begin{cases} -\eta_{\text{abs}}/c^{\mathrm{T}}y, & c^{\mathrm{T}}y < 0, \\ \eta_{\text{abs}}/\sum_i b_i^{\mathrm{T}}z_i, & \sum_i b_i^{\mathrm{T}}z_i > 0, \\ \infty, & \text{otherwise.} \end{cases}$$

- The primal residual

$$r_{\text{pri}} = \max_{i=1,\ldots,L} \frac{\|b_i - B_i y - \sum_{k=1}^{s_i} A_{ik}\,\mathbf{diag}\left(C_{ik}X_{ik}C_{ik}^{\mathrm{T}}\right)\|_2}{\max\{1, \|b_i\|_2\}}.$$

- The dual residual

$$r_{\text{du}} = \max\left\{\frac{\|c - \sum_{i=1}^{L} B_i^{\mathrm{T}}z_i\|_2}{\max\{1, \|c\|_2\}}, \max_{i,k} \|S_{ik} + C_{ik}^{\mathrm{T}}\,\mathbf{diag}\left(A_{ik}^{\mathrm{T}}z_i\right)C_{ik}\|_2\right\}.$$

In these expressions, $\|\cdot\|_2$ denotes the Euclidean norm for vectors and the matrix norm (maximum singular value norm) for matrices. The algorithm terminates if

$$r_{\text{pri}} \le \epsilon_{\text{feas}} \quad \text{and} \quad r_{\text{du}} \le \epsilon_{\text{feas}} \quad \text{and} \quad (\eta_{\text{abs}} \le \epsilon_{\text{gap}} \quad \text{or} \quad \eta_{\text{rel}} \le \epsilon_{\text{gap}}),$$

where $\epsilon_{\text{feas}} = 10^{-7}$ and $\epsilon_{\text{gap}} = 10^{-8}$. The code was implemented in MATLAB version 6.5.1 on a 2.4 GHz Pentium IV PC with 1 GB of memory.

6.2. Linear-phase FIR filter design. We first illustrate the behavior of the algorithm with the example problem of section 4.4. Figure 3 shows the progress of the algorithm applied to the SDP (43) with the same parameters as used for Figure 1. The algorithm terminates after 19 iterations with a CPU time of 0.05 s per iteration.

6.3. Minimax magnitude fit of transfer function. The example in section 5.4 was solved by bisection on δ. The optimal value of δ was computed with an

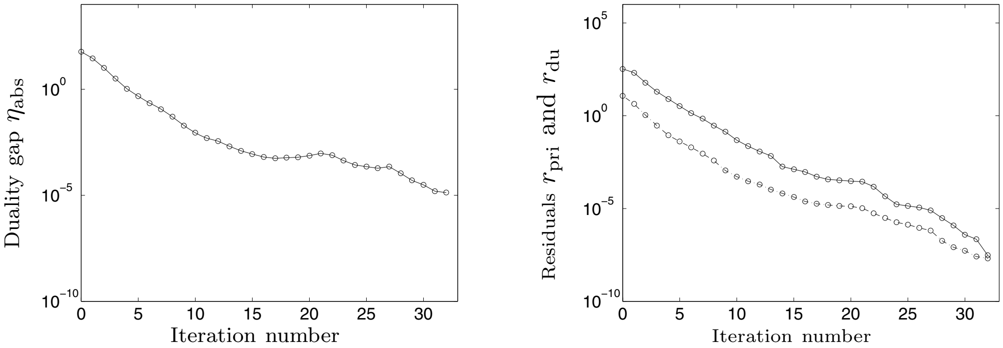


FIG. 4. *Progress of the primal-dual method applied to the phase-*I *problem in the last bisection step for computing the function in Figure* 2. *The left plot shows the duality gap versus iteration number. The right plot shows the primal residual* (*solid line*) *and the dual residual* (*dashed line*).

absolute accuracy of 10^{-5}. We used the basis of Laguerre polynomials to construct the SDP constraints (52). The feasibility problems (for fixed δ) were solved by applying the interior-point method to the "phase-I" problem

$$\begin{array}{ll} \text{minimize} & u \\ \text{subject to} & (\gamma_k - \delta) f_2\big(\omega_k^2\big) - u \le f_1(\omega_k^2) \le (\gamma_k + \delta) f_2(\omega_k)^2 + u, \quad k = 1, \dots, K, \\ & x = W^{\rm T}\big(\,\mathbf{diag}(V_1 X_1 V_1^{\rm T}) + \lambda \circ \mathbf{diag}\left(V_2 X_2 V_2^{\rm T}\right)\big), \\ & y = \tilde{W}^{\rm T}\big(\,\mathbf{diag}\left(\tilde{V}_1 \tilde{X}_1 \tilde{V}_1^{\rm T}\right) + \tilde{\lambda} \circ \mathbf{diag}\left(\tilde{V}_2 \tilde{X}_2 \tilde{V}_2^{\rm T}\right)\big), \\ & X_1 \succeq 0, \quad X_2 \succeq 0, \quad \tilde{X}_1 \succeq 0, \quad \tilde{X}_2 \succeq 0 \end{array} \tag{56}$$

with variables u, x, y, X_i, and $\tilde{X}_i$.

Figure 4 shows the convergence of the primal-dual path-following method applied to the SDP (56) in the final bisection step. Although a primal feasible point for problem (56) is known, the algorithm was started at the default infeasible starting points. Instead of using the stopping criterion based on the duality gap described in section 6.1, we terminated the interior-point algorithm as soon as the sign of the optimal value of (56) was known.

We observed that the convergence of the algorithm for this example problem was much more sensitive to the choice of problem parameters than for the other numerical examples. Although the stability of our interior-point implementation certainly leaves room for improvement, optimization problems over real polynomials on unbounded intervals appear to be much more difficult to solve than problems with cosine polynomials.

6.4. Magnitude FIR filter design. The next example is a family of a lowpass filter design problem similar to examples described in [1] and [8]. The design variables are the (real) filter coefficients h_i of an FIR filter of length $n+1$ with transfer function

$$H(\omega) = h_0 + \sum_{k=0}^{n} h_k e^{-jk\omega}.$$

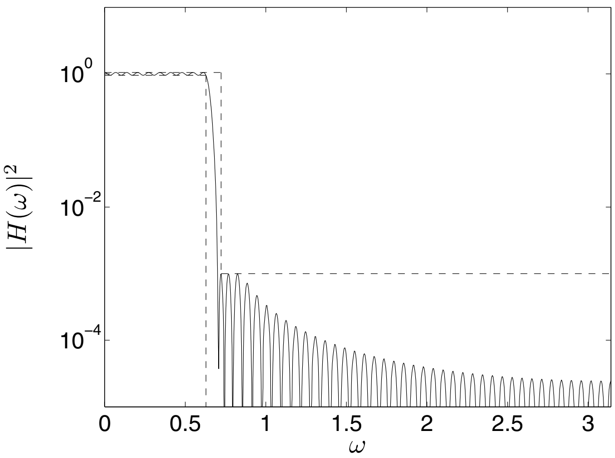


FIG. 5. *Frequency response of lowpass filter with length* 102. *The filter minimizes the stopband energy subject to the upper and lower bounds shown in dashed lines.*

The objective is to minimize the stopband energy

$$\int_{\omega_{\rm s}}^{\pi} |H(\omega)|^2 \, d\omega.$$

The constraints include upper and lower bounds on the filter magnitude $|H(\omega)|$ in the passband, and an upper bound on the magnitude in the stopband:

$$1/\delta_{\rm p} \le |H(\omega)|^2 \le \delta_{\rm p}, \quad 0 \le \omega \le \omega_{\rm p}, \quad |H(\omega)|^2 \le \delta_{\rm s}, \quad \omega_{\rm s} \le \omega \le \pi.$$

This problem can be formulated as a convex problem by expressing the constraints in terms of $Y(\omega) = |H(\omega)|^2$, which is a cosine polynomial

$$Y(\omega) = y_0 + y_1 \cos\omega + \cdots + y_n \cos n\omega.$$

The resulting problem is

$$\begin{array}{ll} \text{minimize} & \displaystyle\int_{\omega_{\rm s}}^{\pi} Y(\omega)\, d\omega \\ \text{subject to} & 1/\delta_{\rm p} \le Y(\omega) \le \delta_{\rm p}, \quad 0 \le \omega \le \omega_{\rm p}, \\ & Y(\omega) \le \delta_{\rm s}, \quad \omega_{\rm s} \le \omega \le \pi, \\ & Y(\omega) \ge 0, \quad 0 \le \omega \le \pi, \end{array} \tag{57}$$

with variables $y \in \mathbf{R}^{n+1}$. From the optimal y, the filter coefficients h_k can be computed via spectral factorization [34].

Since Y is a cosine polynomial, problem (57) can be cast as an SDP of the form (53) as explained in section 4. The problem dimensions are $L = 4$ and $s_i = 2$ for $i = 1, \ldots, L$. The primal variables are the $n+1$-vector y, and eight symmetric matrices X_{ik} of size $\lfloor n/2 \rfloor$ or $\lfloor (n-1)/2 \rfloor$.

We first consider an instance with parameters

$$n = 101, \quad \delta_{\rm p} = 1.05, \quad \delta_{\rm s} = 0.001, \quad \omega_{\rm p} = 0.2\pi, \quad \omega_{\rm s} = 0.23\pi.$$

Figure 5 shows the specifications and the optimal filter magnitude. Figure 6 shows the duality gap and the relative primal and dual residuals versus the iteration number. The code terminates after 20 iterations and requires 0.41 s per iteration.

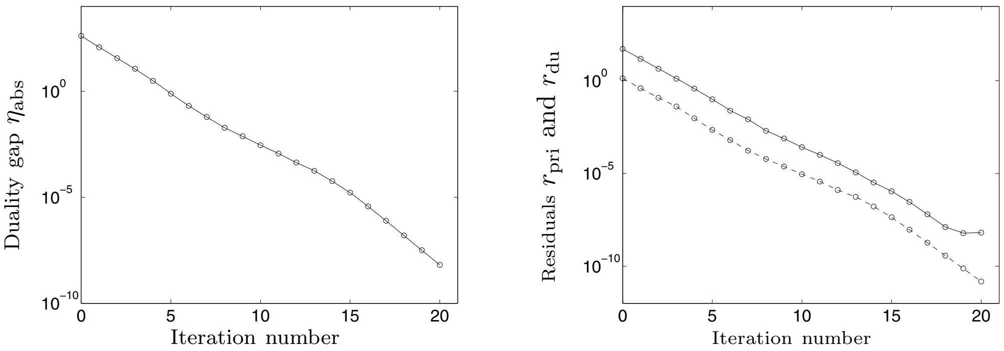


Fig. 6. *Progress of a primal-dual method for the lowpass filter design problem. The left plot shows the duality gap versus iteration number. The right plot shows the primal residual* (*solid line*) *and the dual residual* (*dashed line*).

Table 1

*Numerical results for a family of magnitude filter design problems. The first three columns specify the design parameters. The last two columns show the CPU time per iteration in seconds for a special-purpose interior-point implementation that exploits problem structure and for the general-purpose solver SDPT*3.

Design parameters			Time per iteration (s)	
n	ω_{s}	δ_{s}	Fast impl.	SDPT3
25	0.300π	5.62×10^{-3}	0.04	0.17
50	0.280π	3.16×10^{-3}	0.10	1.81
75	0.270π	1.00×10^{-3}	0.21	5.78
100	0.260π	1.00×10^{-3}	0.41	14.2
125	0.255π	1.00×10^{-3}	0.71	29.0
150	0.250π	1.00×10^{-3}	1.15	55.7
175	0.248π	1.00×10^{-3}	1.77	86.5
200	0.248π	3.16×10^{-4}	2.46	137
225	0.244π	2.24×10^{-4}	3.50	203
250	0.244π	1.78×10^{-4}	4.79	302
275	0.244π	1.78×10^{-4}	6.57	
300	0.244π	1.78×10^{-4}	8.56	

Table 1 show the solution times for 12 filter design problems from the same family with $\omega_{\mathrm{p}} = 0.23\pi$ and $\delta_{\mathrm{p}} = 1.1$ and n ranging from 25 to 300. The stopband parameters ω_{s} and δ_{s} are modified to tighten the specifications as n increases. The last two columns show the CPU time per iteration for the specialized interior-point implementation and for the general-purpose solver SDPT3, applied to the primal problem (53). (To express this problem as a standard form SDP, we split the y variable as a difference of two nonnegative vectors before passing it to SDPT3.) Figure 7 shows a graph of the CPU time versus n. The results clearly illustrate the benefits of exploiting problem structure when solving the Newton equations.

7. Conclusion. We have described a new SDP formulation of sum-of-squares theorems of nonnegative polynomials, cosine polynomials, and trigonometric polynomials. The formulation results in structured SDPs that can be solved very efficiently by taking advantage of simple properties of the **diag** operator.

The SDP parametrizations involve discrete transform matrices that are often orthogonal, or products of orthogonal and diagonal matrices. This should benefit the numerical stability of interior-point algorithms based on the parametrization.

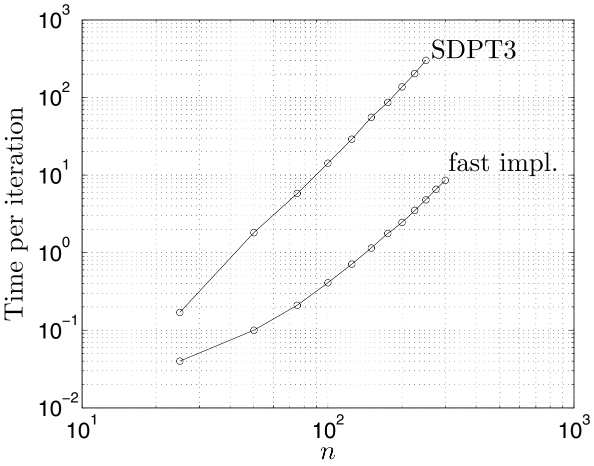


FIG. 7. *CPU time per iteration versus problem dimension for the results in Table* 1.

Although we have not analyzed the numerical properties, the numerical experiments are encouraging. In particular, the FIR filter examples that we solved successfully are much larger than those reported with other fast implementations of interior-point methods [16, 4].

REFERENCES

[1] J. W. ADAMS, *FIR digital filters with least-squares stopbands subject to peak-gain constraints,* IEEE Trans. Circuits Syst., 39 (1991), pp. 376–388.

[2] N. I. AHIEZER AND M. KREIN, *Some Questions in the Theory of Moments,* Transl. Math. Monogr. 2, AMS, Providence, RI, 1962 (translated from the 1938 Russian manuscript by W. Fleming and D. Prill).

[3] F. ALIZADEH, J.-P. A. HAEBERLY, AND M. L. OVERTON, *Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results,* SIAM J. Optim., 8 (1998), pp. 746–768.

[4] B. ALKIRE AND L. VANDENBERGHE, *Convex optimization problems involving finite autocorrelation sequences,* Math. Program. Ser. A, 93 (2002), pp. 331–359.

[5] S. J. BENSON AND Y. YE, *DSDP5: A Software Package Implementing the Dual-Scaling Algorithm for Semidefinite Programming,* Technical report ANL/MCS-TM-255, Mathematics and Computer Science Division, Argonne National Laboratory, 2004.

[6] S. J. BENSON, Y. YE, AND X. ZHANG, *Solving large-scale sparse semidefinite programs for combinatorial optimization,* SIAM J. Optim., 10 (2000), pp. 443–461.

[7] W. L. BRIGGS AND V. E. HENSON, *The DFT: An Owner's Manual for the Discrete Fourier Transform,* SIAM, Philadelphia, PA, 1995.

[8] T. N. DAVIDSON, Z.-Q. LUO, AND J. F. STURM, *Linear matrix inequality formulation of spectral mask constraints*, IEEE Trans. Signal Process., 50 (2002), pp. 2702–2715.

[9] T. N. DAVIDSON, Z.-Q. LUO, AND K. M. WONG, *Design of orthogonal pulse shapes for communications via semidefinite programming,* IEEE Trans. Signal Process., 48 (2000), pp. 1433–1445.

[10] P. J. DAVIS, *Interpolation and Approximation,* Dover, New York, 1975 (first published by Blaisdell Publishing Company in 1963).

[11] B. DUMITRESCU, I. TABUS, AND P. STOICA, *On the parametrization of positive real sequences and MA parameter estimation,* IEEE Trans. Signal Process., 49 (2001), pp. 2630–2639.

[12] L. FAYBUSOVICH, *On Nesterov's approach to semi-infinite programming,* Acta Appl. Math., 74 (2002), pp. 195–215.

[13] L. FAYBUSOVICH, *Semidefinite descriptions of cones defining spectral mask constraints,* European J. Oper. Res., 169 (2006), pp. 1207–1221.

[14] Y. GENIN, Y. HACHEZ, YU. NESTEROV, AND P. VAN DOOREN, *Optimization problems over*

positive pseudopolynomial matrices, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 57–79.

[15] G. H. Golub and J. H. Welsch, *Calculation of Gauss quadrature rules,* Math. Comp., 23 (1969), pp. 221–230.

[16] Y. Hachez, *Convex Optimization over Non-Negative Polynomials: Structured Algorithms and Applications,* Ph.D. thesis, Université Catholique de Louvain, Belgium, 2003.

[17] C. Helmberg, F. Rendl, R. Vanderbei, and H. Wolkowicz, *An interior-point method for semidefinite programming,* SIAM J. Optim., 6 (1996), pp. 342–361.

[18] D. Henrion, D. Arzelier, and D. Peaucelle, *Positive polynomial matrices and improved LMI robustness conditions,* Automatica, 39 (2003), pp. 1479–1485.

[19] M. Kojima, S. Shindoh, and S. Hara, *Interior-point methods for the monotone linear complementarity problem in symmetric matrices,* SIAM J. Optim., 7 (1997), pp. 86–125.

[20] M. G. Krein and A. A. Nudelman, *The Markov Moment Problem and Extremal Problems,* Transl. Math. Monogr. 50, AMS, Providence, RI, 1977.

[21] J. B. Lasserre, *Global optimization with polynomials and the problem of moments,* SIAM J. Optim., 11 (2001), pp. 796–817.

[22] J. Löfberg and P. A. Parrilo, *From coefficients to samples: A new approach to SOS optimization,* in Proceedings of the 43rd IEEE Conference on Decision and Control, 2004, pp. 3154–3159.

[23] P. Moulin, M. Anitescu, K. O. Kortanek, and F. A. Potra, *The role of linear semi-infinite programming in signal-adapted QMF bank design,* IEEE Trans. Signal Process., 45 (1995), pp. 2160–2174.

[24] Y. Nesterov, *Squared functional systems and optimization problems,* in High Performance Optimization Techniques, J. Frenk, C. Roos, T. Terlaky, and S. Zhang, eds., Kluwer Academic, Norwell, MA, 2000, pp. 405–440.

[25] P. A. Parrilo, *Semidefinite programming relaxations for semialgebraic problems,* Math. Program. Ser. B, 96 (2003), pp. 293–320.

[26] A. Stoica, R. Moses, and P. Stoica, *Enforcing positiveness on estimated spectral densities,* Electron. Lett., 29 (1993), pp. 2009–2011.

[27] P. Stoica, T. McKelvey, and J. Mari, *MA estimation in polynomial time,* IEEE Trans. Signal Process., 48 (2000), pp. 1999–2012.

[28] J. F. Sturm, *Using SEDUMI* 1.02*: A Matlab toolbox for optimization over symmetric cones,* Optim. Methods Softw., 11–12 (1999), pp. 625–653.

[29] G. Szegö, *Orthogonal Polynomials,* AMS, Providence, RI, 2003 (first published in 1939).

[30] M. J. Todd, K. C. Toh, and R. H. Tütüncü, *On the Nesterov–Todd direction in semidefinite programming,* SIAM J. Optim., 8 (1998), pp. 769–796.

[31] R. H. Tütüncü and M. J. Todd, *Solving semidefinite-quadratic-linear programs using SDPT*3*,* Math. Program. Ser. B, 95 (2003), pp. 189–217.

[32] P. P. Vaidyanathan, *Multirate Systems and Filter Banks,* Prentice-Hall, Englewood Cliffs, NJ, 1993.

[33] L. Vandenberghe, V. R. Balakrishnan, R. Wallin, A. H. Hanson, and T. Roh, *Interior point algorithms for semidefinite programming problems derived from the KYP lemma,* in Positive Polynomials in Control, D. Henrion and A. Garulli, eds., Lecture Notes in Control and Inform. Sci. 312, Springer, New York, 2005, pp. 195–238.

[34] S.-P. Wu, S. Boyd, and L. Vandenberghe, *FIR filter design via spectral factorization and convex optimization,* in Applied and Computational Control, Signals, and Circuits, Vol. 1, B. Datta, ed., Birkhäuser, Boston, 1998, pp. 215–245.

SIAM J. Optim.
Vol. 16, No. 4, pp. 965–985

INCORPORATING CONDITION MEASURES IN THE CONTEXT OF COMBINATORIAL OPTIMIZATION*

JORGE R. VERA† AND IVÁN DERPICH‡

Abstract. Integer programming algorithms have some kind of exponential complexity in the worst case. However, it is also observed that data instances of similar sizes might have very different practical complexity when solved by computer algorithms. This paper considers an alternative complexity analysis of some integer programming algorithms, which is based on measures of "intrinsic difficulty." The work extends to the setup of integer programming some notions of condition measures which have been developed for convex optimization. We present bounds on the so-called lattice width of polyhedra and address the impact on the complexity of integer programming algorithms like Lenstra's algorithm as well as branch and bound algorithms. The condition measures introduced here reflect shape and spatial orientation factors which are not fully captured by the traditional combinatorial analysis.

Key words. complexity of integer programming, conditioning, complexity theory, error analysis

AMS subject classification. 90C10, 90C57, 68Q25

DOI. 10.1137/040609264

1. Introduction.

This paper is concerned with the consistency or inconsistency of a linear system of the form

$$\text{find } x \in Z^n \text{ such that } Ax \leq b, \tag{1}$$

where A is an m by n real matrix, b is an m-dimensional real vector, and Z denotes the set of integer numbers. The problem is to know whether a real polyhedron contains or does not contain a lattice point. This problem is the basis for the entire area of integer programming. Because of that, we also consider the corresponding optimization variant:

$$\max\{c^T x : Ax \leq b, x \in Z^n\}. \tag{2}$$

A more general problem is usually studied in which x is required to belong to a general lattice. We are restricting the analysis to the special case of the integer lattice, but any other case could be analyzed by reduction to the integer lattice through a linear transformation (see, for instance, Lovász [15]). Through several decades, many connected problems have been analyzed by a large number of researchers, inspired by the fact that the lattice problem is much harder than the usual linear problem, which is solvable in polynomial time in the context of the bit complexity theory; see Schrijver [25]. In that context, the above problems are NP-complete, and although several algorithms exist for them, all have exponential worst case complexity. The practice of integer programming and combinatorial optimization, however, has shown

*Received by the editors June 2, 2004; accepted for publication (in revised form) May 12, 2005; published electronically January 26, 2006. This research was supported by FONDECYT project 1000590.

http://www.siam.org/journals/siopt/16-4/60926.html

†Department of Industrial and System Engineering, School of Engineering, Catholic University of Chile, Campus San Joaquin, Vicuña Mackenna 4860, Santiago, Chile (jvera@ing.puc.cl).

‡Department of Industrial Engineering, School of Engineering, University of Santiago de Chile, Ecuador 3769, Santiago, Chile, and Graduate Program, Catholic University of Chile, Santiago, Chile (iderpich@ usach.cl).

that there is a great variability on the actual difficulty of a real problem. Practitioners are used to applying known techniques that can significantly speed up the operation of algorithms like branch and bound algorithms; see Wolsey [32]. For instance, it is common knowledge that the availability of good upper and lower bounds is crucial for speed, as well as a "clever" selection of the branching variables. A practical observation is that problem instances of the same size (and roughly the same "bit length") might result in very different actual difficulty. There are effects which are not fully captured by the current complexity analysis based on the bit complexity model.

Those characteristics have also been observed in the continuous optimization setting, for instance for linear programming. In that context, alternative complexity analysis has been developed based on the notion of "condition measures" for a problem instances. These measures attempt to recover some "intrinsic" difficulty of the problem instance. The developments in this line, initiated by Renegar [21], have led to complexity analysis in which the running time of algorithms depends strongly on the condition measures.

In this paper we present some potential extensions of this condition based complexity to the area of integer programming. The purpose of going into this line of research is to search for factors affecting the complexity of combinatorial algorithms, which might be related to intrinsic properties of the problem instances. As mentioned already, a condition based complexity theory has been studied in convex optimization, producing interesting results. Our hypothesis is that some conditioning effects are also present in the discrete setting and that they can be adequately studied. Specifically, we present here bounds on the lattice width of a polyhedron, a measure which affects complexity of integer programming algorithms. We revise the complexity analysis of Lenstra's algorithm for integer programming, in which the combined effect of geometric factors as well as conditioning properties of the associated polyhedron appear. We also analyze implications for branch-and-bound-type algorithms. Our results might be the initial steps toward the construction of a condition based complexity theory in the area of discrete optimization, in which intrinsic effects associated to the data of the problem can be taken into consideration.

The structure of this paper is as follows. Section 2 gives a summary of the notion of conditioning as developed for continuous optimization. This is necessary not only for the relevant background but also because the continuous notion will prove useful in the discrete context. Section 3 considers the basis of Lenstra's algorithm for integer programming, specifically the flatness theorem, and the revision of the complexity analysis. In section 4 we develop the complexity analysis for Lenstra's algorithm as well as for branch and bound algorithms, and in section 5 we suggest a comparison of the presented bounds with the ones existing in the literature of combinatorial origin. This analysis is based on recent probabilistic analysis of condition measures.

2. The concept of ill-posedness in optimization and condition measures.

In this section we present a brief review of the notion of conditioning and how it has been applied in optimization. This will serve the purpose of motivating our attempt of further extending those notions to the context of lattice problems.

The concept of conditioning is used in numerical analysis to characterize some properties of problem instances with regard to numerical stability, sensitivity of solutions, etc. The notion has been well studied in numerical linear algebra (there are many references on this, but a good one is Golub and Van Loan [8], which also contains many references), where the concept of condition number of a matrix A is defined as

$\kappa(A) = \|A\|\|A^{-1}\|$ for a specific norm. This number is known to affect sensitivity of solutions to linear systems with respect to changes on the data as well as numerical precision requirements. Another interesting property of this concept is that if A is regular, then $\kappa(A)$ is inversely proportional to the distance from A to the set of singular matrices (in the 2-norm), a measure we call the "distance to ill-posedness" for A. A vast research has been developed to extend this kind of notion to the context of mathematical programming, but the research of interest to us is that developed by J. Renegar, who defines a notion of condition measure for convex optimization problems with analogous properties as the matrix conditioning. The measure defined is also connected to a notion of continuous complexity of algorithms, as has been motivated by Smale [26] (see also Blum et al. [2]).

The specific notion of condition measure for mathematical programming which is of interest to us is briefly reviewed next. Consider the problem

$$z(d) = \min\{\langle c, x\rangle \ : \ b - Ax \in C_Y, x \in C_X\},$$

where X and Y are finite dimensional linear spaces, $C_X \subset X, C_Y \subset Y$ are convex cones, A is a linear operator, $b \in Y$, and c is a linear functional. $\langle \cdot, \cdot \rangle$ denotes an inner product. This is a very general setting which includes several important problems, like linear programming itself and semidefinite programming, which has found much attention in recent times (see, for example, Wolkowicz, Saigal, and Vandenberghe [31]). Suppose the instance $d = (A, b, c)$ corresponds to a feasible problem. Let $\mathcal{F} = \{d \ : \ P(d)$ has a finite solution$\}$. Let $\rho(d) = dist(d, \mathcal{F}^C)$ be the distance (in an appropriate norm) to the set of instances corresponding to problems which are either infeasible or unbounded. This is the distance to ill-posedness for the purpose of computing a solution to the problem. We define

$$C(d) = \frac{\|d\|}{\rho(d)}$$

as the condition measure for instance d. The relevance of the concept of distance to ill-posedness was considered for linear programming by Renegar [21, 22] and by Vera [28, 29]. It is shown in those references that $C(d)$ is connected to measures of sensitivity of solutions with respect to changes on the data. The effect of condition measures in the performance of interior point methods for linear programming has been considered by Renegar [23, 24], where it is concluded that the number of iterations needed to approximate a solution is proportional, among other factors, to $\log C(d)$. The analysis uses a very general characterization of the convergence of Newton's method, which is an extension of the analysis of Nesterov and Nemirovskii [18] based on the property of self-concordance of the barrier function. Vera [30] considers the effect of conditioning on the numerical precision requirements of interior point algorithms. Further characterizations of the distance to ill-posedness for a conic linear program are given in Freund and Vera [5, 7]. Freund and Vera [6] have also considered complexity questions regarding the ellipsoidal algorithm. Further references to current work on the subject are Nuñez and Freund [19], Epelman and Freund [4], and Peña [20].

3. Flatness of polyhedra and condition measures.

In this section we review some results regarding the geometry of convex bodies and their connection with integer programming questions. The starting point is the so-called the "flatness theorem."

3.1. Preliminaries. Let K be a convex body in $\mathbf{R}^n$. We define the lattice width of K as

$$w_L(K) = \min_{v \in Z^n, v \neq 0} w(v, K),$$

where

$$w(v, K) = \max\{v^T x : x \in K\} - \min\{v^T x : x \in K\}.$$

If we restrict the vectors v to the set of unitary vectors in the euclidean space, we obtain the usual "coordinate" width of the body, which is finite if K is bounded. In our specific setup we will assume that K (later a polyhedron) is a bounded set. The lattice width is an interesting measure if we ask the question of its value when K *does not* contain an integral point. Intuition tells us that in this case K cannot be very large in all directions and, hence, has to be either small or large but "flat" in some direction. Khintchine [13] showed that there exists a universal function $f(n)$ such that $w_L(K) \leq f(n)$ if K does not contain an integer point. The specific form of $f(n)$ has been a subject of study for several years and is very important in the formulation of Lenstra's algorithm to decide consistency of (1). The algorithm developed by Lenstra [14] is based on the following general kind of result, which is known as the flatness theorem.

THEOREM 3.1. *Given a convex body K, there exists a function $f(n)$ such that we can achieve one of the following:*

1. *find an integral vector in K, or*
2. *find an integral vector u such that*

$$w(u, K) \leq f(n).$$

Lenstra's estimate for $f(n)$ is in the order of $c_0^{n^2}$, where c_0 is a constant, and the construction can be done in polynomial time. Groetschel, Lovász, and Schrijver [9] improve the bound, still with a polynomial time construction. If the result of the theorem is applied recursively, we can construct an algorithm which slices the polyhedron into lower dimensional ones until it finds an integral point or concludes that none exists.

THEOREM 3.2. *For every fixed n, given a bounded convex body K, there exists a polynomial time algorithm which can find an integral point in K or can assert that none exists.*

A detailed analysis of these two results, together with the proofs, can be found in Lovász [15] and in Groetschel, Lovász, and Schrijver [9]. A significant improvement in the estimate of $f(n)$ can be achieved if we relax our requirement of polynomial time constructibility. For instance, Kannan and Lovász [11] have improved the value to $c_0 n^2$, where c_0 is a numerical constant.

In the existing results, the estimates for $f(n)$ are all of a certain combinatorial nature and do not directly take into consideration information about the "shape" and orientation of the convex body, which might be favorable or unfavorable to the existence of lattice points. Some research on characterizing the shape of polyhedra in connection with the flatness theorem has been done by Kannan, Lovász, and Scarf [12] and by Bárány, Scarf, and Shallcross [1].

3.2. Condition measures and the flatness theorem. In the following subsections we state the results we have obtained which introduce condition measures into the estimates of the flatness theorem. For the purpose of the presentation we work with a bounded polyhedron given by a system of linear inequalities $Ax \leq b$. More specifically, we denote by $d = (A, b) \in \mathbf{R}^{m\times n+m}$ the data corresponding to this specific instance of the problem. We then denote by $P(d)$ the corresponding polyhedron $\{x : Ax \leq b\}$. We will denote by $\alpha_1, \ldots, \alpha_m$ the row vectors of A. We need to measure norms in the data space. There are several alternatives for this, depending on the norms used for vectors and matrices. For this work, unless specified otherwise, we use the 2-norm in the $\mathbf{R}^n$ and $\mathbf{R}^m$ spaces, to measure size of vectors, and the 2-norm for matrices, that is,

$$\|A\|_2 = \max_{\|u\|_2=1} \|Au\|_2.$$

Hence, we define the norm of the data as $\|d\| = \max\{\|b\|_2, \|A\|_2\}$ for an instance $d = (A, b)$. The selection of this norm is functional to the analysis, as it simplifies some of the expressions. As different matrix norms are equivalent, others can be used and corresponding transformation factors will be introduced in the result.

Our analysis is a revision of the flatness theorem, obtaining a bound which depends on other factors besides dimensional information. As in some of the traditional analysis of the flatness theorem, the basis for the estimate is a “rounding” of the polyhedron using inscribed and circumscribed ellipses. This construction is, somehow, arbitrary. The only relevant parameters will be the ratio of the diameters of the ellipses and some indicator of their shape. In any event, we expect the ellipses to interpret the shape of the polyhedron accurately.

So, suppose we can construct a pair of ellipses, with common center x^0, of the form

$$E = \{x \in \mathbf{R}^n : (x - x^0)^T Q(x - x^0) \leq 1\}$$

and

$$E' = \{x \in \mathbf{R}^n : (x - x^0)^T Q(x - x^0) \leq \gamma^2\},$$

such that

$$E \subset P(d) \subset E', \tag{3}$$

where Q is an appropriate positive definite matrix.

There are different possibilities for the definition of such a pair of ellipses, giving different values for the parameter γ. A well-known result of John [10] states the existence of optimal ellipses in the sense that E' has minimum volume. In this case, $\gamma = n$, but the computation is a hard problem. A polynomial time approximation to the so-called Lowner–John pair can be constructed using the shallow cuts ellipsoidal algorithm (see, for instance, Groetschel, Lovász, and Schrijver [9]).

The approach we follow here is based on a construction common in the analysis of interior point methods in convex optimization. Suppose that we know a self-concordant barrier function, Φ, for the convex body, with parameter ϑ (see, for instance, Nesterov and Nemirovskii [18] for an extended discussion). Then let

$$x^0 = \text{argmin}\{\Phi(x) : x \in \text{int}P(d)\}.$$

Let $Q = \nabla^2\Phi(x^0)$ and E be the corresponding ellipse of unit radius. This is called the Dikin ellipse. Then we can take $\gamma = \vartheta + 1$. For instance, if we use the traditional logarithmic barrier function, $\Phi(x) = -\sum_{i=1}^{m} \log(b_i - \alpha_i^T x)$, then $\vartheta = m$. Nesterov and Nemirovskii [18] have shown that there exists a universal barrier with the best possible parameter, $\vartheta = n$. The problem is that the evaluation of this universal barrier is complicated.

In this setup, the point x^0 is called the analytic center of $P(d)$, and the matrix Q is given by

$$Q = A^T D(x^0)^{-2} A \quad \text{with} \quad D(x) = \text{diag}(b_1 - \alpha_1^T x, \ldots, b_m - \alpha_m^T x),$$

where diag() denotes a diagonal matrix constructed with the corresponding elements. The selection of this ellipse construction is justified by the fact that the matrix Q will connect naturally, as we will see, with condition properties of the corresponding polyhedron. The following result follows easily from the geometry of the ellipses.

LEMMA 3.3. *Let Q be a symmetric positive definite real matrix defining a pair of ellipses as in* (3). *Let $u \in \mathbf{R}^n$, $u \neq 0$. Then*

$$w(u, P(d)) \leq 2(m+1)\sqrt{u^T Q^{-1} u}.$$

Proof. It follows by upper bounding $w(u, P(d))$ in terms of $w(u, E')$ and solving the corresponding optimization problems. □

COROLLARY 3.4. *Let Q be a symmetric positive definite real matrix defining a pair of ellipses as in* (3). *Then*

$$w_L(P(d)) \leq 2(m+1)\sqrt{\min\{u^T Q^{-1} u : u \in Z^n, u \neq 0\}}.$$

To obtain the best possible bound in this form we need to solve the problem

$$\min\{u^T Q^{-1} u : u \in Z^n, u \neq 0\}. \tag{4}$$

This is a version of the well-known "shortest vector" problem, which is known to be a hard problem; see Micciancio [17]. We show an upper bound on the optimal value of (4), which will capture some aspects of the original problem which contribute to the relative difficulty of a particular instance.

PROPOSITION 3.5. *Let $v_1,\ldots,v_n$ be the orthonormal eigenvectors of the positive definite matrix Q, and let $\lambda_{\min}$ be the smallest eigenvalue of Q. Then, for any $u \in \mathbf{R}^n$,*

$$u^T Q^{-1} u \leq \left(\frac{1}{\lambda_{\min}}\right) \sum_{i=1}^{n} (v_i^T u)^2.$$

Proof. As Q is symmetric positive definite, the result follows from the fact that

$$Q^{-1} = \sum_{i=1}^{n} \left(\frac{1}{\lambda_i}\right) v_i v_i^T,$$

where λ_i is the ith eigenvalue of Q, corresponding to the eigenvector v_i. □

The next result relates eigenvalues of the matrix Q with other elements of the data. It will be needed in the analysis.

LEMMA 3.6. *Let $Q = A^T D^{-2} A$, where $D = \text{diag}(d_1, \ldots, d_n)$, $d_i > 0, i = 1, \ldots, n$. Let $\lambda_{\min}$ and $\lambda_{\max}$ be the smallest and largest eigenvalue of Q, respectively, and let $\mu_{\min}$ and $\mu_{\max}$ be the smallest and largest eigenvalue of $A^T A$, respectively. Then*

$$\lambda_{\min} \geq \mu_{\min} d_{\max}^{-2}, \quad \lambda_{\max} \leq \mu_{\max} d_{\min}^{-2},$$

where $d_{\max} = \max\{d_1, \ldots, d_n\}$ *and* $d_{\min} = \min\{d_1, \ldots, d_n\}$.

Proof. This is Proposition 4.1 of [30]. □

Let us define the following set in the space of instances for the problem:

$$\mathcal{F} = \{(A, b) : \text{ the set } P(d) \text{ is bounded}\}.$$

This set is important for the purpose of determining whether $P(d)$ contains a lattice point or not. In fact, if $d = (A, b)$ is in the interior of $\mathcal{F}$, then it is certainly possible for $P(d)$ to have no lattice points. However, if $P(d)$ is an unbounded set such that the recession cone is nondegenerate (with nonempty interior), then we can be sure that $P(d)$ does have lattice points and, furthermore, $w_L(P(d)) = \infty$. It is in this sense that we can think of the set $\mathcal{F}^C$ as the set of ill-posed instances for the problem of estimating $w_L(P(d))$ (and, furthermore, bounding the complexity of Lenstra's algorithm). For $d \in \mathcal{F}$ let us define

$$\rho_U(d) = \inf\{\|\Delta d\| : P(d + \Delta d) \in \mathcal{F}^C\}.$$

This can be thought of as the distance to ill-posedness for this specific problem setting.

LEMMA 3.7. *Let* $d = (A, b) \in \mathcal{F}$. *Let* $\mu_{\min}$ *be the smallest eigenvalue of* $A^T A$. *Then*

$$\rho_U^2(d) \leq \mu_{\min}.$$

Proof. The set defined by $Ax \leq b$ is bounded if and only if the homogenous system $Au \leq 0, u \neq 0$, is inconsistent. Let E be a perturbation of the matrix A such that the matrix $A + E$ is rank deficient. Then there exists $v \neq 0$ such that $(A + E)v = 0$. This implies that

$$\rho_U(d) \leq \|E\|_2$$

for any matrix E with this property. Let us choose E to be the matrix of minimum distance to rank deficiency. We know that in this case $\|E\|_2$ equals the smallest singular value of A, which is equal to $\sqrt{\mu_{\min}}$. From this the result follows. □

With this, we are ready to prove our first main result.

THEOREM 3.8. *Let* λ_i *and* v_i *denote the ith eigenvalue and eigenvector, respectively, of the matrix* Q, $i = 1, \ldots, n$. *Let* u *be a feasible solution to* (4). *Define*

$$\bar{C}_I(u, P(d)) = \left(\sum_{i=1}^{n} (v_i^T u)^2\right)^{1/2}.$$

Then

$$w(u, P(d)) \leq 2(m+1)\bar{C}_I(u, P(d)) \left(\frac{\|b - Ax^0\|_\infty}{\rho_U(d)}\right).$$

Proof. We first observe from Lemma 3.3 that

$$w(u, P(d)) \leq 2(m+1)\sqrt{u^T Q^{-1} u}.$$

Using Proposition 3.5 now we have

$$\begin{aligned} u^T Q^{-1} u &\leq \frac{1}{\lambda_{\min}} \sum_{i=1}^{n} (v_i^T u)^2 \\ &\leq \frac{1}{\lambda_{\min}} \bar{C}_I(u, P(d))^2. \end{aligned}$$

Now, from Lemma 3.7

$$\frac{1}{\mu_{\min}} \leq \frac{1}{\rho_U(d)^2}.$$

Using this bound together with Lemma 3.6 we have

$$\frac{1}{\lambda_{\min}} \leq \left(\frac{d_{\max}}{\rho_U(d)}\right)^2.$$

We then have

$$\begin{aligned} w(u, P(d)) &\leq 2(m+1)\sqrt{u^T Q^{-1} u} \\ &\leq 2(m+1)\frac{1}{\sqrt{\lambda_{\min}}}\bar{C}_I(u, P(d)) \\ &\leq 2(m+1)\bar{C}_I(u, P(d))\left(\frac{d_{\max}}{\rho_U(d)}\right), \end{aligned}$$

which proves the result, as $d_{\max} = \|b - Ax^0\|_\infty$. □

We observe that the theorem gives a bound which is valid for any direction u. The actual size of the bound will depend on the combined effect of $\bar{C}_I(u, P(d))$ and $\rho_U(d)$. To interpret the result of the theorem, consider a polyhedron which is small in volume. Then $\|b - Ax^0\|_\infty$ will be small and the bound is small. In fact, in this case the width of that polyhedron in any direction will be small. On the other hand, consider a polyhedron which is very thin. In this case, a small perturbation of the constraints can make the polyhedron unbounded. Hence, $\rho_U(d)$ is small. Now, the vector of minimum lattice width does not need to coincide with the vector defining the thinnest direction of the polyhedron (recall that it has to have integral components). Due to the "elongated" shape of the polyhedron, the lattice width could be achieved at points far apart, giving a large value for the number. The bound incorporates both effects simultaneously. The term in parentheses has to do with the shape of the polyhedron, and the term $\bar{C}_I(u, P(d))$ summarizes the relative orientation of the polyhedron with respect to the vector u, in which direction flatness is being measured. Now, a small value for $\bar{C}_I(u, P(d))$ is obtained when u is one of the unit coordinate vectors and one of the vectors v_i is parallel to it. This situation corresponds to the case when the polyhedron is parallel to the coordinate axis.

4. Consequences for integer programming algorithms.

The flatness theorem is the central stone on Lenstra's construction of a polynomial time algorithm for integer programming in fixed dimension. We elaborate in this section on the implications on the complexity bound of the algorithm if we use the above presented version of the lattice width estimates.

The algorithm will make use of the flatness theorem and the construction we have presented. The complexity will be evaluated using the bounds presented in the previous sections. We recall that the construction, as well as the complexity estimates, depend on a given vector u in which we evaluate flatness. The next result shows that a good enough bound is obtained by taking any u with unit ∞-norm. This allows us to make the results independent of a hard to compute vector u.

LEMMA 4.1. *Suppose that u is such that $\|u\|_\infty \leq 1$. Then*

$$\bar{C}_I(u, P(d)) \leq \sqrt{n}.$$

Proof. As $v_i, i = 1, \dots, n$, are eigenvectors of Q, they form an orthonormal basis. Let $\gamma_i, i = 1, \dots, n$, be such that $u = \sum_{i=1}^n \gamma_i v_i$. Then

$$\bar{C}_I(u, P(d)) = \left(\sum_{i=1}^n (v_i^T u)^2 \right)^{1/2} = \left(\sum_{i=1}^n (\gamma_i)^2 \right)^{1/2} = \|u\|_2 \leq \sqrt{n},$$

as we wanted to prove. □

4.1. The algorithm. Lenstra's algorithm is well known; it is based on the central idea of the flatness theorem. We develop here a specific description of the procedure, adopting the use of ellipses based on the logarithmic barrier function for the polyhedron. Our description defines several subroutines which are used by the algorithm. The algorithm can operate in two modes: In feasibility mode, the algorithm executes the subroutine **FEAS** recursively on the original instance $d = (A, b)$. In optimization mode, the algorithm uses the main subroutine **OPTIM** to perform the optimization.

The following routine basically applies the flatness theorem to the problem of deciding consistency of a system of the form $Ax \leq b$.

Subroutine **FLAT**(A, b).

Input: A polyhedron $P(d) = \{x : Ax \leq b\}$.

Output: $\hat{y}$ integer such that $A\hat{y} \leq b$; OR $\hat{w}$ integer, approximate flatness direction; OR the conclusion that $P(d)$ is empty.

```
1. If $P(d)$ is empty, STOP(return: "$P(d)$ is inconsistent").
2. Compute $x^0$ the analytic center of $P(d)$.
3. Let $Q$ be the matrix of the Dikin ellipse centered on $x^0$.
4. Let $\hat{y} = round(x^0)$.
5. If $A\hat{y} \leq b$, STOP(return $\hat{y}$).
6. If not, select $\hat{w} \in \mathbf{R}^n$ such that $\|\hat{w}\|_\infty = 1$ as a flatness direction.
7. STOP with $\hat{w}$ as an approximate direction in which the polyhedron is flat.
```

This subroutine is essentially the implementation of the algorithmic implications of the flatness theorem. In the combinatorial analysis, the determination of a potential integral point in the polyhedron is done using a basis reduction procedure (see Lovász [15]), which runs in polynomial time in the bit complexity model. The rounding of the polyhedron is done using the ellipsoidal algorithm. In our case, the operation is different: step 1 of the procedure, the construction of the rounding, requires the solution of a convex optimization problem. However, it suffices with an approximation to the analytic center, and this approximation can be computed efficiently with a small number of Newton iterations, as we discuss later. Step 4 searches for a potential integral point in $P(d)$ using a very naive approach: just look for the closest point with integral components. Step 6 provides an integral vector to estimate the lattice width by just taking any w with $0, 1, -1$ coordinates. This is consistent with the bound in Lemma 4.1 but also allow us to simplify the computation of the whole procedure. A more precise approach will be to solve in step 6 the problem

$$\begin{array}{ll} \min & w^T Q^{-1} w \\ \text{s.t.} & \\ w \in Z^n & \\ \|w\|_\infty = 1, & \end{array}$$

giving a better flatness direction. However, as we stated before, this is an NP-hard problem. From a more practical point of view, one might conceive to approximate a solution to the above problem using some kind of heuristic global optimization procedure. The worst case bound, however, is still the one we present in the following.

The next step of the algorithm is to apply this routine recursively through a range of values. Let $r^+ = \max\{w^T x : Ax \leq b\}$ and $r^- = \min\{w^T x : Ax \leq b\}$. The search will proceed by considering the lower dimension polyhedrons $Ax \leq b, w^T x = \gamma$, with $r^- \leq \gamma \leq r^+$. The problem, then, reduces to deciding lattice feasibility of $Ax \leq b, w^T x = \gamma$. The next proposition shows that this is equivalent to determining lattice feasibility of a corresponding inequality system in lower dimension.

PROPOSITION 4.2. *Let u be an integral vector such that $\|u\|_\infty = 1$, and suppose that there exists a coordinate $u_k \neq 0$ and that γ is integer. The system $Ax \leq b, u^T x = \gamma$ is lattice feasible if and only if the system $\tilde{A}x \leq \tilde{b}$ is lattice feasible, where*

$$\tilde{A} = A_N - \frac{1}{u_k} A_{\cdot k} u_N^T, \qquad 2\tilde{b} = b - \frac{1}{u_k} A_{\cdot k} \gamma,$$

with A_N being the matrix with $n-1$ columns corresponding to A with the kth column removed and u_N being the corresponding part of the vector u.

Proof. It is easy to see that $\tilde{A}$ and $\tilde{b}$ result from A and b by eliminating the kth variable, using the equation $u^T x = \gamma$. Now, Let $\tilde{x}$ be an integral point in R^{n-1} such that $\tilde{b} - \tilde{A}\tilde{x} \geq 0$. We have that

$$\begin{aligned}
\tilde{b} - \tilde{A}\tilde{x} &= b - \frac{1}{u_k} A_{\cdot k}\gamma - \left(A_N - \frac{1}{u_k} A_{\cdot k} u_N^T\right)\tilde{x} \\
&= b - A_N \tilde{x} - A_{\cdot k}\left(\frac{1}{u_k}\gamma - \frac{u_N^T \tilde{x}}{u_k}\right) \\
&= b - A\bar{x},
\end{aligned}$$

where $\bar{x} \in R^n$ is defined using all components of $\tilde{x}$ in all positions, except the kth, which is equal to $\frac{1}{u_k}(\gamma - u_N^T \tilde{x})$. Now, γ and $u_N^T \tilde{x}$ are integers, so $\gamma - u_N^T \tilde{x}$ is an integer. Furthermore, given that u is chosen such that $\|u\|_\infty = 1$ (and, hence, $u_k \in \{1, -1\}$), $\frac{1}{u_k} \in \{-1, 1\}$ and $\bar{x}_k$ is an integer.

Reciprocally, if $\bar{x}$ is an integer vector in R^n such that $A\bar{x} \leq b, u^T \bar{x} = \gamma$, it is easy to see that $\tilde{x}$ defined as the vector consisting of all components of $\bar{x}$ but the kth is feasible for $\tilde{A}$ and $\tilde{b}$. □

The next subroutine applies **FLAT** to check feasibility of a polyhedron.

SUBROUTINE **FEAS**(A, b).

Input: A polyhedron $P(d) = \{x : Ax \leq b\}$.

Output: $\hat{y}$ integer such that $A\hat{y} \leq b$; OR the conclusion that $P(d)$ is lattice free with $\hat{w}$, an approximate flatness direction; OR the conclusion that $P(d)$ is empty.

```
1. Apply FLAT$(A, b)$.
2. If the output is an integral $\hat{y} \in P(d)$, STOP(return $\hat{y}$).
3. If the output is "inconsistent," STOP(return "$P(d)$ is inconsistent").
4. If the output is a flatness direction $w$ then
5. let $r^+ = \max\{w^T x : Ax \leq b\}$ and $r^- = \min\{w^T x : Ax \leq b\}$.
6. Repeat for all integers $\gamma$, $r^- \leq \gamma \leq r^+$, the following until stopping or until
   covering the whole range:
   (a) Consider the system $Ax \leq b, w^T x = \gamma$. Transform the system as spec-
       ified in Proposition 4.2 into a system in $n-1$ variables of the form
       $\tilde{A}v \leq \tilde{b}$.
   (b) Call FEAS$(\tilde{A}, \tilde{b})$.
```

```
    (c) If the output is an integral $\tilde{x} \in \mathbf{R}^{n-1}$,
        • reconstruct the corresponding point $\bar{x}$ associated to the change of
          variable, as specified in Proposition 4.2. $\bar{x}$ is an integer;
        • STOP(return $\bar{x}$);
        • if not (FEAS returns lattice infeasibility), continue with the next
          $\gamma$.
7. If the whole range of $\gamma$ was covered and no integral point was detected,
   STOP(return $P(d)$ is lattice free).
```

It can be noticed that the call is defined recursively. The correctness of the above procedure depends on the fact that for each "slice" of the polyhedron, the feasibility problem is reduced to one in less dimensions. Observe that, due to the constraint imposed to the selection of w, the reconstructed point in step (6c) is always guaranteed to be an integer, as specified in Proposition 4.2.

The following routine is the main optimizer.

```
SUBROUTINE OPTIM$(A, b, c)$.
Input: An integer program of the form $\max\{c^T x : Ax \leq b, x \in Z^n\}$, with $c$ integral.
Output: An integral point in $P(d)$, the solution to the optimization problem, or the
assertion that $P(d)$ is lattice free.
1. Let $R^+ = \max\{c^T x : Ax \leq b\}$ and $R^- = \min\{c^T x : Ax \leq b\}$.
2. For all integers $\beta$, with $R^- \leq \beta \leq R^+$, from $R^+$ down to $R^-$ and until the
   answer is YES, do the following:
    (a) Consider the system $Ax \leq b, c^T x \geq \beta$. Let
        $$\bar{A} = \begin{bmatrix} A \\ -c \end{bmatrix}, \qquad \bar{b} = \begin{bmatrix} b \\ -\beta \end{bmatrix}.$$
    (b) Call FEAS$(\bar{A}, \bar{b})$.
    (c) If the answer is an integral $\bar{x} \in \mathbf{R}^n$, STOP(return $\bar{x}$, optimal solu-
        tion).
    (d) If not, continue with the next $\beta$.
3. If the whole range of $\beta$ was covered and no integral point was detected,
   STOP(return "$P(d)$ is lattice free").
```

It is worth pointing out that our main optimization routine begins by searching for an integral point in subsets of the original polyhedron, restricted by the constraint $c^T x \geq \beta$. Unlike **FEAS** these polyhedrons are of the same dimension as $Ax \leq b$. Subsequent calls will slice them into lower dimensional objects.

4.2. Complexity of Lenstra's algorithm in feasibility mode.

In this subsection and the following we analyze the computational effort of the above algorithm, using the estimates of section 3 for the lattice width of the polyhedron. First we consider the complexity of Lenstra's algorithm in "feasibility mode," that is, when it is used to decide lattice feasibility of a closed polyhedron. The reasoning is just the customary for Lenstra's algorithm: We observe that the main procedure **FEAS** applies **FLAT** to the whole set $P(d)$ first, using a direction u to evaluate flatness, and then applies itself recursively on the sets $P(\tilde{d})$, where $\tilde{d}$ is the instance in $\mathbf{R}^{m\times(n-1)+m}$ obtained by eliminating one variable, as in Proposition 4.2. Each call requires the application of **FLAT**. The total number of calls can be bounded by the lattice width of $P(d)$ in the direction u, and for each one of those, there are $w(u, P(\tilde{d}))$ calls in a polyhedron with one less dimension. Let N be the total number of calls to **FLAT**, and let N_k be an upper bound on the number of calls in the k-dimensional problems.

Then

$$N \leq N_n \times N_{n-1} \times N_{n-2} \times \cdots, \tag{5}$$

where, for an instance $\tilde{d}$ in the space of dimension $k \leq n$, using Theorem 3.8 we can put a bound $N_k \leq w(u, P(\tilde{d}))$. So, the whole analysis reduces to bounding $w(u, P(d))$ for instances in different dimensions. First, we introduce some additional notation and some results which relate elements in the bounds for one problem to the bounds in the reduced ones. The results are valid for any norm. Let

$$\bar{\sigma} = \max_{x \in P(d)} \|b - Ax\|_\infty. \tag{6}$$

We observe that $\|b - Ax^0\|_\infty \leq \bar{\sigma}$.

Recall that $\tilde{A}$ and $\tilde{b}$ correspond to the transformed problem data in $n-1$ dimensions, obtained from A and b.

Lemma 4.3. *Let $\tilde{x}$ be such that $\tilde{x} \in P(\tilde{d})$. Then*

$$\|\tilde{b} - \tilde{A}\tilde{x}\|_\infty \leq \bar{\sigma}.$$

Proof. Let $\tilde{x}$ be such that $\tilde{b} - A\tilde{x} \geq 0$. From Proposition 4.2,

$$\begin{aligned}\tilde{b} - \tilde{A}\tilde{x} &= b - \frac{1}{u_k} A_{\cdot k}\gamma - \left(A_N - \frac{1}{u_k} A_{\cdot k} u_N^T\right)\tilde{x} \\ &= b - A_N\tilde{x} - A_{\cdot k}\left(\frac{1}{u_k}\gamma - \frac{u_N^T\tilde{x}}{u_k}\right) \\ &= b - A\bar{x},\end{aligned}$$

where $\bar{x}$ is defined using all components of $\tilde{x}$ in all positions, except the kth, which is equal to $\frac{1}{u_k}\gamma - \frac{u_N^T\tilde{x}}{u_k}$. Hence, $\bar{x} \in P(d)$ and

$$\|\tilde{b} - \tilde{A}\tilde{x}\|_\infty \leq \max_{x \in P(d)} \|b - Ax\|_\infty = \bar{\sigma}$$

and the result follows. □

We point out that, although the definition of $\bar{\sigma}$ uses the infinity norm, we could have used any other norm, and the result of Lemma 4.3 is still valid. The following result establishes a relation between $\rho_U(d)$ and $\rho_U(\tilde{d})$ for the lower dimensional polyhedrons generated in the algorithm. Notice that these are distances to ill-posedness in different data spaces.

Lemma 4.4. *Let $\rho_U(\tilde{d})$ be the distance to ill-posedness for the data $\tilde{d} = (\tilde{A}, \tilde{b})$ as generated in step* 6(a) *of routine* **FEAS**. *Then $\rho_U(d) \leq \rho_U(\tilde{d})$.*

Proof. Consider the instance $\tilde{d} = (\tilde{A}, \tilde{b})$ and suppose that the polyhedron $P(\tilde{d})$ is bounded. Let $\Delta\tilde{d} = (\tilde{E}, \tilde{e})$ be such that $\tilde{d} + \Delta\tilde{d}$ is ill posed and $\rho_U(\tilde{d}) = \|\Delta\tilde{d}\| \geq \|\tilde{E}\|_2$. Then there exists $\tilde{v} \in \mathbf{R}^{n-1}$, $\tilde{v} \neq 0$, such that $(\tilde{A} + \tilde{E})\tilde{v} \leq 0$, implying that the perturbed polyhedron is ill-posed, that is, unbounded. Hence,

$$\left(A_N - \frac{1}{u_k} A_{\cdot k} u_N^T + \tilde{E}\right)\tilde{v} \leq 0 \implies (A_N + \tilde{E})\tilde{v} + A_{\cdot k}\left(-\frac{1}{u_k} u_N^T\tilde{v}\right) \leq 0.$$

Let $\bar{x} \in \mathbf{R}^n$ be defined as follows: All components except the kth are equal to the corresponding components of $\tilde{v}$, and the kth component is equal to $-\frac{1}{u_k} u_N^T\tilde{v}$. Then

$\bar{x} \neq 0$. Let $E \in \mathbf{R}^{m \times n}$ be a matrix equal to $\tilde{E}$ in all columns except the kth, which is zero. We have then that $(A + E)\bar{x} \leq 0$. This implies that

$$\rho_U(d) \leq \|E\|_2 = \|\tilde{E}\|_2 \leq \rho_U(\tilde{d}),$$

as we wanted to prove. □

The next result establishes an upper bound for $w(u, P(\tilde{d}))$ for any $\tilde{d}$ generated.

PROPOSITION 4.5. *For any $u \in Z^{n-1}$ such that $\|u\|_\infty = 1$*

$$w(u, P(\tilde{d})) \leq 2(m+1)\sqrt{n-1}\left(\frac{\bar{\sigma}}{\rho_U(d)}\right)$$

for all instances $\tilde{d}$ generated in step 6(a) *of subroutine* **FEAS**.

Proof. We first observe from Theorem 3.8 that

$$w(u, P(d)) \leq 2(m+1)\bar{C}_I(u, P(d))\left(\frac{\|b - Ax^0\|_\infty}{\rho_U(d)}\right).$$

Now, from Lemma 4.1, we have that $\bar{C}_I(u, P(d)) \leq \sqrt{n}$ for a u such that $\|u\|_\infty = 1$. Then if v^0 is the center of the ellipse for the problem $\tilde{A}x \leq \tilde{b}$, we have

$$\begin{aligned} w(u, P(\tilde{d})) &\leq 2(m+1)\sqrt{n-1}\left(\frac{\|\tilde{b} - \tilde{A}v^0\|_\infty}{\rho_U(\tilde{d})}\right) \\ &\leq 2(m+1)\sqrt{n-1}\left(\frac{\bar{\sigma}}{\rho_U(d)}\right), \end{aligned}$$

where the last inequality follows from Lemmas 4.3 and 4.4. □

These results allow us to prove the following theorem.

THEOREM 4.6. *Lenstra's algorithm will detect an integral point in $P(d)$ or certify that none exists in at most*

$$2^n(m+1)^n(n!)^{1/2}\left(\frac{\bar{\sigma}}{\rho_U(d)}\right)^n$$

calls to subroutine **FLAT**.

Proof. This follows from repeated application of Proposition 4.5 to relation (5). □

The theorem gives an upper bound on the number of calls to subroutine **FLAT**. The actual effective complexity for a specific problem instance might be much smaller. If we recall Theorem 3.8, we presented a bound in which the spatial orientation of the polyhedron might favor a better estimate for $w(u, P(d))$. Moreover, if the algorithm uses a good approximation to the shortest vector problem as the slicing direction, the overall search procedure might be sped up significantly.

It is also worth pointing out here that if $P(d)$ is lattice free, the algorithm will have to examine all slices $A \leq b, u^T x = \gamma$, where γ is within the range of the lattice width. On the other hand, if the polyhedron is in fact lattice feasible, Lenstra's algorithm might be fortunate enough to find an integral point fairly earlier. This illustrates a certain asymmetry in both problems, although they have the same "hard" complexity. Proving lattice infeasibility might be, in general, harder than proving lattice feasibility, at least using these algorithms as a proving tool.

4.3. Complexity of Lenstra's algorithm in optimization mode. Lenstra's algorithm in optimization mode uses the routine **OPTIM** as main control, searching first in the direction of the objective vector c. To complete the analysis we recall the following definition.

Definition 4.7. $\delta = \max\{c^T x : Ax \leq b\} - \max\{c^T x : Ax \leq b, x \in Z^n\}$ *is known as the integrality gap of the integer problem.*

In the first level, this subroutine will call **FEAS** a maximum of $\lceil \delta \rceil$ times, where δ is the integrality gap of the problem, which is the difference between the optimal value of the integer program and the optimal value of the associated linear relaxation. Each time, **FEAS** is called with the system $Ax \leq b, c^T x \geq \beta$ for some β as an argument. Let as assume that $\beta_1, \beta_2, \ldots, \beta_p$ are the values of β generated, and let $N(\beta)$ be the number of iterations (calls to **FLAT**) required by **FEAS** on the system $Ax \leq b, c^T x \geq \beta$. Then the total number of iterations required to optimize is

$$N(\beta_1) \times N(\beta_2) \times \cdots \times N(\beta_p).$$

We need to compute an upper bound to $N(\beta)$.

Lemma 4.8. *Let $\rho_U(d)$ be the distance to ill-posedness for the polyhedron defined by $Ax \leq b$. Let $d = (A, b)$ denote the corresponding data. Let $\bar{d} = (\bar{A}, \bar{b})$ be the data corresponding to the system $Ax \leq b, c^T x \geq \beta$, where*

$$\bar{A} = \begin{bmatrix} A \\ -c^T \end{bmatrix}, \qquad \bar{b} = \begin{bmatrix} b \\ -\beta \end{bmatrix}.$$

Let $\rho_U(\bar{d})$ be the distance to ill-posedness for $(\bar{A}, \bar{b})$. Then

$$\rho_U(d) \leq \rho_U(\bar{d}).$$

Proof. Let $(\bar{E}, \bar{e})$ be a perturbation of the data $\bar{d}$ such that $(\bar{A}, \bar{b}) + (\bar{E}, \bar{e})$ is ill-posed and $\rho_U(\bar{d}) = \|(\bar{E}, \bar{e})\|$. That is, the polyhedron defined by the system $(\bar{A}+\bar{E})x \leq (\bar{b} + \bar{e})$ is unbounded. This means that there exists $u \neq 0$ such that $(\bar{A} + \bar{E})u \leq 0$. Let E be the first m rows of $\bar{E}$. Then $(A+E)u \leq 0$. This means that the polyhedron defined by the data $(A + E, b)$ is unbounded. Hence,

$$\rho_U(d) \leq \|E\| \leq \|(\bar{E}, \bar{e})\| = \rho_U(\bar{d})$$

as we wanted to prove. □

Proposition 4.9. *Let $\bar{d} = (\bar{A}, \bar{b})$ be the data corresponding to the system $Ax \leq b, c^T x \geq \beta$. Let u be an integral vector such that $\|u\|_\infty = 1$. Then*

$$w(u, P(\bar{d})) \leq 2(m+2)\sqrt{n} \left(\frac{\tilde{\sigma}}{\rho_U(d)} \right),$$

where

$$\tilde{\sigma} = \max_{x \in P(\bar{d})} \|\bar{b} - \bar{A}x\|_\infty.$$

Proof. This follows immediately from Theorem 3.8 and Lemmas 4.1 and 4.8. □

With these results we can prove the following theorem.

Theorem 4.10. *Consider the application of the modified Lenstra algorithm to the integer programming problem*

$$\max\{c^T x : Ax \leq b, x \in Z^n\}.$$

Let δ be the integrality gap of the problem. Assume that the polyhedron $K = \{x : Ax \leq b\}$ is nonempty and contains an integral point (this can be checked in feasibility mode). Then the procedure will find an optimal solution in at most

$$\lceil\delta\rceil 2^n (m+2)^n (n!)^{1/2} \left(\frac{\max\{\bar{\sigma}, \lceil\delta\rceil + 1\}}{\rho_U(d)}\right)^n$$

calls to subroutine **FLAT**, *where $\bar{\sigma}$ is defined as in* (6).

Proof. In the first level, the subroutine performs at most $w(u, P(\bar{d}))$ calls to **FEAS** if an integral point is not detected immediately. Now, each one of those calls corresponds to the application of the subroutine to a polyhedron in $n-1$ variables, and we can use the bound in Theorem 4.6, with $\tilde{\sigma}$ instead of $\bar{\sigma}$, as now **FEAS** receives $\bar{d}$ as imput. Hence, for all possible i,

$$(7)\ N(\beta_i) \leq 2(m+2)\sqrt{n}\left(\frac{\tilde{\sigma}}{\rho_U(d)}\right) \times 2^{n-1}(m+2)^{n-1}((n-1)!)^{1/2}\left(\frac{\tilde{\sigma}}{\rho_U(d)}\right)^{n-1}.$$

Now, we obtain a bound for $\tilde{\sigma}$. We have that

$$\tilde{\sigma} = \max_{x \in P(\bar{d})} \|\bar{b} - \bar{A}x\|_\infty.$$

From the definition of $(\bar{A}, \bar{b})$ we have that

$$\|\bar{b} - \bar{A}x\|_\infty \leq \max\{\bar{\sigma}, \max_{x \in P(\bar{d})} |\beta - c^T x|\}.$$

Now, when the procedure stops, it will be after detecting an integral point in the polyhedron defined by the system $Ax \leq b, c^T x \geq \beta$. This means that the polyhedron defined by $Ax \leq b, c^T x \geq \beta + 1$ was lattice infeasible, and as β decreases by one unit on each iteration, we can be sure that $|c^T x - \beta| \leq \lceil\delta\rceil + 1$ for all $x \in P(\bar{d})$. Hence,

$$\tilde{\sigma} \leq \max\{\bar{\sigma}, \lceil\delta\rceil + 1\}$$

and replacing in (7) we have

$$N(\beta_i) \leq 2^n (m+2)^n (n!)^{1/2} \left(\frac{\max\{\bar{\sigma}, \lceil\delta\rceil + 1\}}{\rho_U(d)}\right)^n.$$

As there are at most δ values β_i, the bound follows. □

Notice that this result assumes the existence of integer solutions to the problem; otherwise, δ is not defined. However, the existence of integral solutions can be checked with one single call to **FEAS** before calling **OPTIM**. Both in this case and in feasibility mode, we are giving an estimate of the computational effort in terms of calls to the subroutine **FLAT**. This routine is where the main effort is incurred. The computation of an approximate analytic center requires the application of the Newton method to the minimization of a barrier function, although this can be carried out efficiently. Actually, the effort is similar to the one incurred by an interior point algorithm to approximately solve an optimization problem.

4.4. The impact on the complexity of the branch and bound algorithm. The standard algorithm to solve integer programming problems is the well-known branch and bound procedure. In this section we give some ideas regarding possible implications of the previous results on the actual complexity of the branch and bound algorithm. Practitioners know very well that in the worst case the procedure will have an exponential behavior, but they also know that some cleverly chosen branching rules might have a considerable impact on the actual performance.

We now give a worst case estimate for the complexity of the branch and bound algorithm. This estimate is similar to others presented, for instance, in Schrijver [25].

Proposition 4.11. *Consider the integer programming problem*

$$\min\{c^T x : x \in P(d), x \in Z^n\},$$

where $P(d)$ is a bounded polyhedron and c is an integer vector. Let $e_1, \dots, e_n$ be the unitary coordinate directions. Let $w(e_i, P(d))$ be the integer width range of P measured along the objective function e_i. Then the number of nodes to be examined by the branch and bound procedure can be bounded by

$$\prod_{i=1}^{n}(w(e_i, P(d)) + 1) \leq (\max_i w(e_i, P(d)) + 1)^n. \tag{8}$$

Proof. This estimate is based on the fact that in the worst case, the procedure will branch on all integer values available per each coordinate, and that number is clearly bounded by the above expression. □

An easy bound on $w(e_i, P(d))$ can be obtained from all our previous development, and an overall bound is the following.

Theorem 4.12. *The worst case complexity of the branch and bound algorithm can be bounded by*

$$2^n(m+1)^n n^{n/2}\left(\frac{\bar{\sigma}}{\rho_U(d)}\right)^n.$$

Proof. This follows by using Theorem 3.8 and Lemma 4.1 with $u = e_i$, bounding $\bar{C}_I(u, P(d))$ by $\sqrt{n}$, as $\|u\|_\infty = 1$, and using the definition of $\bar{\sigma}$ in the bound. □

It is tempting to compare this worst case bound with the one obtained for Lenstra's algorithm in the previous subsection. There is no indication of the tightness of the bounds, so no precise comparison is possible. However, one can still argue on the order of the size of the elements which enter in the bounds (5) and (8). First, notice that although we bounded each of the terms $w_L(P(\tilde{d}))$, they will tend to be smaller for lower dimensional problems $\tilde{d}$ and also even smaller if a good approximation is used in the shortest vector problem. On the other hand, the terms $w(e_i, P(d))$ in the branch and bound estimate might all be of roughly the same size. This implies that Lenstra's algorithm might be favored by the fact that the slices are taken precisely in the direction of minimum lattice width. The branch and bound algorithm, on the other hand, will be favored if the orientation of the polyhedron tends to make the minimum lattice width vector coincide with one of the coordinate directions. Moreover, if the objective function happens to coincide with the minimum width direction, this will certainly improve the running time.

5. An expected value comparison with a combinatorial analysis.

The results obtained so far present a different approach to bounding the lattice width of a polyhedron and hence provide some alternative bound on complexity. As mentioned in section 3, the flatness theorem has been extensively investigated, and there are several bounds of a combinatorial nature. All bounds associated to a polynomial time calculation are of the form

$$w(u, P(d)) \leq n^{\beta} 2^{n},$$

where β is a numerical constant independent of the problem data. The bound shown in this work depends on very different parameters, which are instance dependent. So, for fixed dimensions n and m, our bound will give different values. How does it compare with the above? This is a difficult question, but one possibility is to perform some kind of mean value analysis of the factors entering in the definition of the bounds of section 3. We do this now by relaying a recent analysis of average performance of condition measures presented by Dunagan, Spielman, and Teng [27].

Let us consider for this purpose the polyhedron defined by $d = (A, b)$. In the previously cited work, the authors analyze several condition measures for linear programming formats, including the problem of finding a nontrivial solution to the system $Ax \leq 0, x \neq 0$. The format of the problem under consideration here is $Ax \leq b$, but as the cited authors point out, this can be reduced to the homogeneous case $Ax - x^0 b \leq 0, x^0 > 0$, preserving distances to ill-posedness. Moreover, in our specific case, the system $Ax \leq b$ is unbounded if and only if the system $Ax \leq 0, x \neq 0$, is infeasible. This allows us to translate the results to our format. Let

$$C_U(d) = \frac{\|d\|}{\rho_U(d)}.$$

The smoothed analysis presented by Dunagan, Spielman, and Teng is a "local" analysis of expected values. Suppose that $\|(\bar{A}, \bar{b})\|_F \leq 1$ in the Frobenius norm. We also assume that (A, b) are samples of independent Gaussian random variables of variance σ^2, centered at $(\bar{A}, \bar{b})$. The results allows us to claim that with probability $1 - \nu$,

$$C_U(d) = O\left(\frac{n^2 m^2}{\sigma^2 \nu}\left[\frac{n^2 m^2}{\sigma^2 \nu}\right]^{O(1)}\right). \tag{9}$$

This result says that the condition number $C_U(d)$ of a particular instance, on average, around a neighborhood of some instance is not exponentially large.

To use this result in our bounds we need to introduce $C_U(d)$ explicitly. We can do that with the following.

LEMMA 5.1. *Let $x \in P(d)$. Then $\|x\|_2 \leq C_U(d)$.*

Proof. We have that $Ax \leq b$. If $x = 0$, the bound holds trivially. Hence, we assume $x \neq 0$. Let $\bar{e}$ be such that $\|\bar{e}\|_2 = 1$ and $\|x\|_2 = \bar{e}^T x$. Let $\Delta A = -b\bar{e}^T / \|x\|_2$. Then $(A + \Delta A)x \leq 0$. This means that the instance could be made unbounded with an arbitrarily small perturbation, and hence

$$\rho_U(d) \leq \|\Delta A\|_2 \leq \|b\|_2 \|\bar{e}\|_2 / \|x\|_2.$$

This implies that

$$\|x\|_2 \leq \frac{\|b\|_2}{\rho_U(d)} \leq \frac{\|d\|}{\rho_U(d)} = C_U(d),$$

as we wanted to prove. □

We now consider the term $\bar{\sigma}$ which appears in the complexity estimate.

LEMMA 5.2.

$$\bar{\sigma} \leq 2\|d\| C_U(d).$$

Proof. This follows by observing that if $\bar{x}$ is the point where the value of $\bar{\sigma}$ is attained, then

$$\bar{\sigma} = \|b - A\bar{x}\|_2 \leq \|b\|_2 + \|A\bar{x}\|_2 \leq \|b\|_2 + \|A\|_2\|\bar{x}\|_2.$$

From Lemma 5.1, the definition of $\|d\|$, and the fact that $C_U(d) \geq 1$ (as an appropriate perturbation of size $\|d\|$ will make the instance unbounded for sure), the result follows. $\square$

Combining Lemmas 5.1 and 5.2 with Theorem 3.8 we have the following.

PROPOSITION 5.3.

$$w_L(P(d)) \leq 4(m+1)\sqrt{n}C_U(d)^2.$$

Now, based on the smoothed analysis discussed, we can estimate $C_U(d)$ for a fixed probability ν by a number of the form

$$\beta\left(\frac{mn}{\sigma}\right)^{O(1)},$$

where β is a numerical constant (recall that σ is the standard deviation of the data). This gives an approximate estimate for $w_L(P(d))$ in the order of

$$\beta^2\left(\frac{n^3m^3}{\sigma^2}\right)^{O(1)}.$$

This value is much smaller than the worst case one and the ones obtained by the combinatorial analysis. This is, of course, only a vague indication of the possible average impact of the conditioning of the polyhedron into the real actual complexity of algorithms.

6. Conclusions and further discussion.

In this paper we have presented several results which correspond to an alternative analysis of the complexity of an integer programming algorithm, namely Lenstra's algorithm. We also gave some suggestions on the implications for branch and bound procedures. The main result of the paper gives a bound on the lattice width of a polyhedron which depends on some specific properties of the data related to the conditioning of the polyhedron. This bound is used to derive a bound on the running time of the algorithms. The fact that the data instance is close to defining an unbounded polyhedron affects the estimate of the lattice width.

We now address various topics regarding the assumptions of the analysis, its limitations, and potential extensions.

The use of barrier functions. The analyses of the flatness theorem found in the literature, in general, make use of the idea of rounding the polyhedron by using appropriate ellipses. We are doing the same, but the difference is that we have used the Dikin ellipse constructed from the logarithmic barrier function of the polyhedron. This has the advantage of connecting easily with continuous condition measures. It has the disadvantage that for the proper operation of the algorithm, the knowledge of a strict interior point is required, something which might be considered restrictive. Also, the ellipses will be affected if the formulation of the problem contains redundant constraints, although the shape of the polyhedron will still be the same. In [5] it is shown that the condition measures of polyhedra are also connected to inscribed and circumscribed ellipses generated by the ellipsoidal algorithm. This relation could be used to reformulate our algorithms and obtain possibly similar bounds to the one presented here but using the ellipsoidal algorithms as a driving machine instead. This analysis will be closer to other existing ones for the flatness theorem in the combinatorial optimization literature (see Lovász [15] and Groetschel, Lovász, and Schrijver [9]).

Tightness and computability of the bounds. One might ask, how tight are the bounds, and it is easy to realize that they are not tight. First, they are based on a rounding of a polyhedron using ellipses, a fact which already introduces a difference. Also, the exact bound might not be achievable as they require the computation of a shortest integral vector, a hard problem. For the purpose of the algorithm, the problem in step 6 of subroutine **FLAT** could be approximated by any practical and fast procedure or just evaluated on the coordinate unitary vectors, which gives the bound of Lemma 4.1.

Analytic center computation and complexity in terms of the Newton method. The conceptual algorithm requires the computation of the analytic center, but it suffices with an approximation. We suggest now how the analysis could be extended to exhibit a complexity bound depending on Newton iterations used to approximate the analytical center. As a main tool, we can make use of the following well-known result from the interior point literature (see Renegar [23]).

PROPOSITION 6.1. *Let $K = \{x : Ax \leq b\}$ and assume that K is consistent and has nonempty interior. Let x^0 be such that $Ax^0 < b$. Let $\epsilon > 0$ be given. Then an approximation $\tilde{x}$ to x^0 such that $\|\tilde{x} - \hat{x}\|_Q \leq \epsilon$ can be computed after at most*

$$O\left(\sqrt{n}\log\frac{1}{\epsilon\mathrm{dist}(x^0, \partial K)}\right)$$

iterations of the Newton method, where $\|u\|_Q = \sqrt{u^TQu}$ and $\mathrm{dist}(x^0, \partial K)$ denotes the distance from x^0 to the boundary of the K.

To use this result we must assume that we know in advance a point x^0 in the interior of $P(d) = \{x : Ax \leq b\}$. Following the construction in Proposition 4.2, we will also need to argue that from x^0 it is possible to construct a point $\tilde{x}$ in the interior of any polyhedron $\tilde{A}x \leq \tilde{b}$ generated during the successive slicing of the original polyhedron. This can be achieved as follows: Consider the system $Ax \leq b, u^Tx = \gamma$, as in Proposition 4.2. Solve the optimization problem $\max\{u^Tx : Ax \leq b\}$, and let x' be an optimal solution. Let $\bar{x}$ be the point in the line segment between x^0 and x' contained in the hyperplane $u^Tx = \gamma$. Using the arguments in the proof of Proposition 4.2, one can show that the point $\tilde{x}$ obtained from $\bar{x}$ by eliminating the kth component satisfies $\tilde{A}\tilde{x} < \tilde{b}$. Now we will need to argue that there exists a constant η such that $\eta \geq \mathrm{dist}(\tilde{x}, \partial P(\tilde{d}))$ for all polyhedrons $P(\tilde{d})$ generated, where $\tilde{x}$ is the suitably chosen starting interior point. Under those assumptions, we can choose, for instance, $\epsilon = 1/3$, as has been customary in the interior point methods literature, and modify subroutine **FLAT** to compute only an ϵ approximation to the analytic center. Then the complexity estimates in Theorem 4.6 and 4.10 will be modified only by the factor $(m+1+1/3)$, as we are shifting the interior reference point by a relative distance of $1/3$, in the norm of the matrix Q. We can then combine those theorems with Proposition 6.1 to claim an overall bound on the number of Newton iterations of the form

$$O\left(2^n(m+2)^n(n!)^{1/2}\sqrt{n}\log\left(\frac{3}{\eta}\right)\left(\frac{\bar{\sigma}}{\rho_U(d)}\right)^n\right)$$

for the feasibility problem. A similarly extended bound will follow from Theorem 4.10. In both cases, the analysis and the bound depend on the existence of an appropriate lower bound η.

In the combinatorial analysis, a polynomial time result is given for the application of the flatness theorem, based on the ellipsoidal algorithm and the basis reduction algorithm, which run in polynomial time. We are not presenting here a formal polynomial time result, but the meaning of efficiency is more of a continuous nature in the sense that only a finite number of Newton iterations are required, as in the above bound.

Potential extensions. The approach presented here suggests that information from the lattice width of a polyhedron could be used to solve integer programs more efficiently. We are conducting research on selection rules for branch and bound based on this information; see [3]. Also Mehrotra, Sheng, and Li [16] have implemented Lenstra's algorithm using disjunctive cuts originating on information from the rounding of the polyhedron. Our results could also be used to analyze the complexity of that implementation.

Acknowledgments. The authors would like to thank an anonymous referee for the very helpful comments which allowed us to improve the first version of the paper. We also would like to thank FONDECYT for the funding of this research. The first author would also like to thank Prof. James Renegar for directing him to the literature on the flatness theorem.

REFERENCES

[1] I. Bárány, H. Scarf, and D. Shallcross, *The topological structure of maximal lattice free convex bodies: The general case*, Math. Program., 80 (1998), pp. 1–16.

[2] L. Blum, F. Cucker, M Schub, and S. Smale, *Complexity and Real Computation*, Springer-Verlag, Berlin, 1998.

[3] I. Derpich and J. Vera, *Improving the efficiency of the branch and bound algorithm for integer programming based on "flatness" information*, European J. Oper. Res., to appear.

[4] M. Epelman and R. Freund, *Condition number complexity of an elementary algorithm for computing a reliable solution of a conic linear system* Math. Program., 88 (2000), pp. 451–485.

[5] R. Freund and J. Vera, *Some characterizations and properties of distance to ill-posedness and the condition measure of a conic linear system*, Math. Program., 86 (1999), pp. 225–260.

[6] R. Freund and J. Vera, *Condition-based complexity of optimization in conic linear form via the ellipsoidal algorithm*, SIAM J. Optim., 10 (1999), pp. 155–176.

[7] R. Freund and J. Vera, *On the complexity of estimating condition measures for conic convex optimization*, Math. Oper. Res., 28 (2003), pp. 625–648.

[8] J. Golub and C. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, 1989.

[9] A. Groetschel, L. Lovász, and A. Schrijver, *Geometric Algorithms in Combinatorial Optimization*, 2nd ed., Springer-Verlag, Berlin, 1992.

[10] F. John, *Extremum problems with inequalities as subsidiary conditions*, in Studies and Essays Presented to R. Courant on His 60th Birthday, January 8, 1948, Intersciences, New York, 1948, pp. 187–204.

[11] R. Kannan and L. Lovász, *Covering minima and lattice-point-free convex bodies*, Ann. of Math., 128 (1988), pp. 577–602.

[12] R. Kannan, L. Lovász, and H. Scarf, *The shape of polyhedra*, Math. Oper. Res., 15 (1990), pp. 364–380.

[13] A. Khintchine, *A quantitative formulation of Kronecker's theory of approximation*, Izv. Akad. Nauk. SSSR. Ser. Mat., 12 (1948), pp. 113–122, in Russian.

[14] H. W. Lenstra, Jr., *Integer programming with a fixed number of variables*, Math. Oper. Res., 8 (1983), pp. 538–548.

[15] L. Lovász, *Algorithmic Theory of Numbers, Graphs, and Convexity*, CBMS-NSF Regional Conf. Ser. in Appl. Math. 50, SIAM, Philadelphia, 1986.

[16] S. Mehrotra, H. Sheng, and Z. Li, *A modified Lenstra method for mixed integer programming*, INFORMS Annual Meeting, San José, 2002.

[17] D. Micciancio, *The shortest vector in a lattice is hard to approximate to within some constant*, SIAM J. Comput., 30 (2001), pp. 2008–2035.

[18] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM Stud. Appl. Math. 13, SIAM, Philadelphia, 1994.

[19] M. Nuñez and R. Freund, *Condition measures and properties of the central trajectory of a linear program*, Math. Program., 83 (1998), pp. 1–28.

[20] J. Peña, *Understanding the Geometry of Infeasible Perturbations of a Conic Linear System*, SIAM J. Optim., 10 (2000), pp. 534–550.

[21] J. Renegar, *Incorporating condition measures into the complexity theory of linear programming*, SIAM J. Optim., 5 (1995), pp. 506–524.

[22] J. Renegar, *Some perturbation theory for linear programming*, Math. Program., 65 (1994), pp. 73–91.

[23] J. Renegar, *Linear programming, complexity theory, and elementary functional analysis*, Math. Program., 70 (1995), pp. 279–351.

[24] J. Renegar, *Condition numbers, the barrier method, and the conjugate-gradient method*, SIAM J. Optim., 6 (1996), pp. 879–912.

[25] A. Schrijver, *Theory of Linear and Integer Programming*, John Wiley and Sons, New York, 1986.

[26] S. Smale, *Some remarks on the foundations of numerical analysis*, SIAM Rev., 32 (1990), pp. 211–220.

[27] J. Dunagan, D. Spielman, and S. Teng, *Smoothed Analysis of Renegar's Condition Number for Linear Programming*, http://math.mit.edu/spielman/SmoothedAnalysis (2002).

[28] J. Vera, *Ill-Posedness in Mathematical Programming and Problem Solving with Approximate Data*, Ph.D. dissertation, Cornell University, Ithaca, NY, 1992.

[29] J. Vera, *Ill-posedness and the complexity of deciding existence of solutions to linear programs*, SIAM J. Optim., 6 (1996), pp. 549–569.

[30] J. Vera, *On the complexity of linear programming under finite precision arithmetic*, Math. Program., 80 (1998), pp. 91–123.

[31] H. Wolkowicz, R. Saigal, and L. Vandenberghe, *Handbook of Semidefinite Programming*, Kluwer Academic Publishers, Boston, 2000.

[32] L. A. Wolsey, *Integer Programming*, John Wiley and Sons, New York, 1998.

SIAM J. OPTIM.
Vol. 16, No. 4, pp. 986–1006

FAST ALGORITHMS FOR PROJECTION ON AN ELLIPSOID[*]

YU-HONG DAI[†]

Abstract. Several fast algorithms are proposed for the problem of projecting a point onto a general ellipsoid. To avoid the direct estimation of the spectral radius in the Lin–Han algorithm, we provide the maximal 2-dimensional inside ball algorithm and the sequential 2-dimensional projection algorithm. However, we find that the solution procedure of the former algorithm may tend to generate some 2-dimensional reduced ellipsoids, and the latter algorithm may produce zigzags. Therefore we investigate the hybrid use of the two algorithms. Our numerical experiments show that all the algorithms, even the hybrid algorithms, are suitable for large-scale problems and much faster than the Lin–Han algorithm. Linear convergence of the algorithms is established. Possible extensions of the algorithms are also discussed.

Key words. projection, ellipsoid, large-scale, hybrid algorithm, linear convergence

AMS subject classifications. 65K05, 90C06

DOI. 10.1137/040613305

1. Introduction.

The problem of projection on a general convex set

$$\begin{array}{ll} \min & d(\mathbf{a},\mathbf{x}) \\ \text{s.t.} & \mathbf{x} \in \mathcal{C}, \end{array} \tag{1.1}$$

where $d(\cdot,\cdot)$ is some distance function and $\mathcal{C}$ is some convex set in $\mathcal{R}^n$, is one of the fundamental problems in convex analysis. It is also an important inertia of projection methods for nonlinear programming, variational inequality problems, etc. For example, Birgin, Martínez, and Raydan [1] established efficient spectral projected gradient algorithms for optimization over convex sets. Evidently, the performance of their algorithms is very related to the subprojection algorithm on the convex set. Although the problem (1.1) has been well studied in theory, little is known about how to solve the problem except when $\mathcal{C}$ is some special set such as a ball, a box, a box with a singly linear constraint (for example, see [12, 2]), or an order simplex (for example, see [6]).

In this paper, we consider the following problem of projecting a point onto a general ellipsoid:

$$\begin{array}{ll} \min & d(\mathbf{a},\mathbf{x}) = \|\mathbf{x}-\mathbf{a}\| \\ \text{s.t.} & \mathbf{x} \in \mathcal{E} := \{\mathbf{x} \in \mathcal{R}^n : q(\mathbf{x}) \le \alpha\}, \end{array} \tag{1.2}$$

where $\mathbf{a} \in \mathcal{R}^n$ is a point to be projected, $q(\mathbf{x}) = \mathbf{x}^T A\mathbf{x} + 2\mathbf{b}^T\mathbf{x}$, A is a positive definite matrix in $\mathcal{R}^{n\times n}$, and $\|\cdot\|$ means the 2-norm. Note that the convex set $\mathcal{C}$ can usually be written as

$$\mathcal{C} = \bigcap_{i=1}^{m} \{\mathbf{x} \in \mathcal{R}^n : \ g_i(\mathbf{x}) \le 0\}, \tag{1.3}$$

[*] Received by the editors August 7, 2004; accepted for publication (in revised form) June 10, 2005; published electronically January 26, 2006. This work was supported by the EPRSC in the UK (GR/R87208/01) and by Chinese NSF grants 10171104 and 40233029.

http://www.siam.org/journals/siopt/16-4/61330.html

[†] Department of Mathematics, University of Dundee, Dundee DD1 4HN, Scotland, UK, and LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Box 2719, Beijing 100080, China (dyh@lsec.cc.ac.cn).

where m is some positive integer and $g_i(\mathbf{x})$ $(i = 1, \ldots, m)$ is some concave function in $\mathcal{R}^n$. Since a suitable local approximation to a nonlinear function is a quadratic function, the problem (1.2) is fundamental in solving the problem (1.1) with $\mathcal{C}$ given by (1.3). If the problem with $m = 1$ is well solved, one can then use methods such as those in [4] and [7] to solve the general problem with any m. The problem (1.2) with $\mathbf{b} = \mathbf{0}$ is also related to the trust region subproblem in nonlinear optimization.

To solve the problem (1.2), Lin and Han [9] proposed a simple and geometric algorithm for the problem (1.2) with $\mathbf{b} = \mathbf{0}$ with attractive convergence properties. Suppose that the current iteration is $\mathbf{x}_k$ that belongs to the boundary of $\mathcal{E}$. The basic idea of their algorithm is to construct an n-dimensional ball that lies inside the ellipsoid $\mathcal{E}$ and is tangent to the boundary of $\mathcal{E}$ at $\mathbf{x}_k$, and then take $\mathbf{x}_{k+1}$ to be the intersection of the boundary of $\mathcal{E}$ and the line segment connecting $\mathbf{a}$ and the center of the ball $\mathbf{c}_k$. Consequently, they have to estimate the spectral radius of A in some way. As analyzed in section 3, however, a lower estimate of this quantity may deteriorate the performance of the algorithm greatly. Nevertheless, if we consider the choice of $\mathbf{x}_{k+1}$ on the 2-dimensional linear manifold $\mathcal{S}_k = \mathbf{x}_k + \mathrm{Span}\{\mathbf{a} - \mathbf{x}_k, A\mathbf{x}_k + \mathbf{b}\}$, then much faster algorithms can be obtained.

The rest of this paper is organized as follows. In the next section, we present a general framework for all the algorithms considered in this paper. In section 3, a numerical analysis of the Lin–Han algorithm is provided. Two new algorithms, namely, a maximal 2-dimensional inside ball algorithm and a sequential 2-dimensional projection algorithm, are proposed in sections 4 and 5, respectively. In section 6, we investigate the hybrid use of the two algorithms and propose the simple hybrid projection algorithm and the general hybrid projection algorithm. A linear convergence result is established in section 7 for the general hybrid projection algorithm, which has the Lin–Han algorithm, maximal 2-dimensional inside ellipsoid algorithm, and sequential 2-dimensional projection algorithm as its special cases. Numerical results with the algorithms are reported in section 8. Discussion is given in the last section.

2. The general algorithm. Throughout this paper, we assume that $q(\mathbf{a}) > \alpha$, because otherwise the projection of $\mathbf{a}$ on the ellipsoid $\mathcal{E}$ is itself. We also assume that

$$\alpha > \min\{q(\mathbf{x}) : \mathbf{x} \in \mathcal{R}^n\} = -\mathbf{b}^T A^{-1}\mathbf{b}, \tag{2.1}$$

so that $\mathcal{E}$ exists and is not a singleton. In this section, we describe a general algorithm whose diagram is shown in Figure 1. This algorithm requires a feasible initial point $\mathbf{x}_0$ and generates a sequence $\{\mathbf{x}_k\} \subset \Omega(\mathcal{E})$, where $\Omega(\mathcal{E})$ is the boundary of $\mathcal{E}$,

$$\Omega(\mathcal{E}) = \{\mathbf{x} \in \mathcal{R}^n : q(\mathbf{x}) = \alpha\}. \tag{2.2}$$

Suppose that a feasible point $\mathbf{x}_k$ is obtained at the kth iteration. Denote $\mathbf{u}_k = \nabla q(\mathbf{x}_k)/2 = A\mathbf{x}_k + \mathbf{b}$. The algorithm calculates an intermediate point $\mathbf{c}_k$ along the negative gradient of q at $\mathbf{x}_k$:

$$\mathbf{c}_k = \mathbf{x}_k - \gamma_k \mathbf{u}_k, \tag{2.3}$$

where $\gamma_k > 0$ is so chosen that $\mathbf{c}_k \in \mathcal{E}$, namely, $q(\mathbf{c}_k) \le \alpha$. For any $\mathbf{x}$, $\mathbf{y} \in \mathcal{R}^n$ with $\mathbf{x} \neq \mathbf{y}$, denote by $\mathcal{L}(\mathbf{x}, \mathbf{y})$ the line segment connecting $\mathbf{x}$ and $\mathbf{y}$,

$$\mathcal{L}(\mathbf{x}, \mathbf{y}) = \{\mathbf{x} + \eta\,(\mathbf{y} - \mathbf{x}) : \eta \in [0, 1]\}. \tag{2.4}$$

The algorithm takes $\mathbf{x}_{k+1}$ as the minimizer of the distance $\|\mathbf{x} - \mathbf{a}\|$ on the set $\mathcal{L}(\mathbf{a}, \mathbf{c}_k) \cap \mathcal{E}$. Equivalently, defining $\mathbf{w}_k = \mathbf{c}_k - \mathbf{a}$, the algorithm calculates

$$\mathbf{x}_{k+1} = \mathbf{a} + \eta_k\, \mathbf{w}_k, \tag{2.5}$$

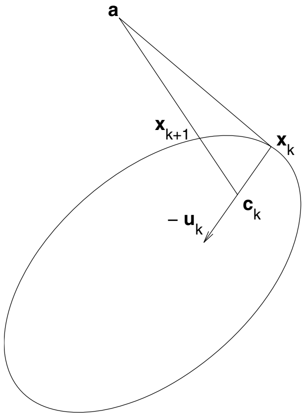


FIG. 1. *Diagram of the general algorithm*

where $\eta_k \in (0, 1)$ is such that $\mathbf{x}_{k+1} \in \Omega(\mathcal{E})$. The above procedure is then repeated until some convergence criterion is satisfied.

Let us denote $\mathbf{g}_a = \nabla q(\mathbf{a})/2 = A\,\mathbf{a} + \mathbf{b}$; the requirement $\mathbf{x}_{k+1} \in \Omega(\mathcal{E}_k)$ asks η_k to satisfy

$$\alpha = q(\mathbf{x}_{k+1}) = q(\mathbf{a} + \eta_k \mathbf{w}_k) = (\mathbf{w}_k^T A \mathbf{w}_k)\,\eta_k^2 + 2\,(\mathbf{g}_a^T \mathbf{w}_k)\,\eta_k + q(\mathbf{a}) := \psi(\eta_k). \tag{2.6}$$

Notice that $\psi(0) = q(\mathbf{a}) > \alpha$, $\psi(1) = q(\mathbf{a} + \mathbf{w}_k) = q(\mathbf{c}_k) \le \alpha$, and $\psi(\eta) \to +\infty$ as $\eta \to +\infty$. Therefore the quadratic equation $\psi(\eta) = \alpha$ has one root in $(0, 1)$ and one root in $(1, +\infty)$. The smaller one is taken for η_k, namely,

$$\eta_k = -\frac{\mathbf{g}_a^T \mathbf{w}_k}{\mathbf{w}_k^T A \mathbf{w}_k} - \sqrt{\left(\frac{\mathbf{g}_a^T \mathbf{w}_k}{\mathbf{w}_k^T A \mathbf{w}_k}\right)^2 - \frac{q(\mathbf{a}) - \alpha}{\mathbf{w}_k^T A \mathbf{w}_k}}. \tag{2.7}$$

Define $\mathbf{v}_k = \mathbf{a} - \mathbf{x}_k$. The following criterion is used for the termination of the algorithm:

$$1 - \frac{\mathbf{u}_k^T \mathbf{v}_k}{\|\mathbf{u}_k\|\,\|\mathbf{v}_k\|} \le \varepsilon, \tag{2.8}$$

where $\varepsilon > 0$ is some tolerance error. Now we provide a detailed description of the general algorithm.

GENERAL ALGORITHM.

1. *Given a starting point* $\mathbf{x}_0 \in \Omega(\mathcal{E})$ *and* $\varepsilon > 0$. *Set* $k := 0$.
2. *Calculate* γ_k *in some way,* $\mathbf{u}_k = A\mathbf{x}_k + \mathbf{b}$, *and* $\mathbf{c}_k$ *by* (2.3).
3. *Calculate* $\mathbf{w}_k = \mathbf{c}_k - \mathbf{a}$ *and* $\mathbf{x}_{k+1}$ *by* (2.5) *and* (2.7).
4. *If* (2.8) *does not hold, set* $k := k + 1$ *and go to step* 2.

As will be seen, all the algorithms in this paper are special cases of the above general algorithm, but differ on the choice of γ_k.

3. A numerical analysis of the Lin–Han algorithm. The algorithm by Lin and Han [9], which consists of constructing a ball that lies inside the ellipsoid $\mathcal{E}$ and intersects $\Omega(\mathcal{E})$ at the point $\mathbf{x}_k$, is a special case of the General Algorithm. More

Table 1
Performances of the Lin-Han algorithm with $\gamma_k = 0.01\,\zeta$.

ζ	1	0.5	0.2	0.1	0.05	0.02	0.01
#iter	142	284	710	1420	2839	7098	14197

exactly, their algorithm aims to find a positive number γ_k such that the n-dimensional inside ball

$$\mathcal{B}(\gamma_k) = \{\|\mathbf{x} - \mathbf{c}_k\| \le \gamma_k \|\mathbf{u}_k\| : \mathbf{x} \in \mathcal{R}^n\} \subset \mathcal{E}. \tag{3.1}$$

Consequently, Lin and Han require the choice of γ_k to satisfy the condition

$$\tau \le \gamma_k \le (\rho(A))^{-1}, \tag{3.2}$$

where τ is some small positive constant and $\rho(A)$ is the spectral radius of A.

As analyzed in [9], the first inequality can provide a sufficient decrease of $d(\mathbf{a}, \mathbf{x}_k)$, and hence the global convergence of the algorithm can be established. The function of the second inequality is to guarantee that the n-dimensional ball $\mathcal{B}(\gamma_k)$ lies inside the ellipsoid $\mathcal{E}$. For this purpose, the authors need to estimate some matrix norm $|||A||| \ge \rho(A)$, and the 1-norm or ∞-norm is suggested. As will be shown, the numerical performance of their algorithm heavily relies on the estimation of the spectral radius $\rho(A)$ and an underestimation of this quantity may deteriorate the algorithm greatly.

Consider the following 10-dimensional example.

Example 1. $\mathcal{E} = \{\mathbf{x} \in \mathcal{R}^{10} : q(\mathbf{x}) := \sum_{i=1}^{10} i^2 x_i^2 = 385\}$, $\mathbf{a} = (a_i)$ with $a_i = 10i^2 + 1\,(i = 1, \dots, 10)$. The initial point is $\mathbf{x}_0 = \sqrt{\frac{385}{q(\mathbf{a})}}\,\mathbf{a}$. In this example, the projection $\mathbf{x}^*$ of $\mathbf{a}$ on $\mathcal{E}$ is $\mathbf{x}^* = (1, 1, \dots, 1)^T$.

Since in this example $A = \mathrm{diag}(1, 4, \dots, 100)$, we have that $\rho(A) = 100$. We tested the Lin–Han algorithm using $\gamma_k = \zeta\,(\rho(A))^{-1} = 0.01\,\zeta$ with different values of $\zeta \le 1$. The tolerance error ε in (2.8) was set to 10^{-6}. Table 1 lists the numbers of iterations required by the algorithm with different values of ζ.

From Table 1, we see that the number of iterations required by the Lin–Han algorithm is almost linearly dependent on the value ζ. A good estimation of the spectral radius $\rho(A)$ may accelerate the algorithm significantly. This example even suggests that it is worthwhile to do so before the projection calculations if a good approximation can be obtained with relatively low cost. As will be seen in the following sections, however, this estimation procedure is not necessary, and algorithms much faster than the Lin–Han algorithm even with $\gamma_k = (\rho(A))^{-1}$ can be obtained.

4. Maximal 2-dimensional inside ball algorithm.

Our new algorithms are based on the following observation: all the points $\mathbf{a}$, $\mathbf{x}_k$, $\mathbf{c}_k$, and $\mathbf{x}_{k+1}$ lie in the 2-dimensional linear manifold

$$\mathcal{S}_k = \{\mathbf{x}_k + (\mathbf{u}_k, \mathbf{v}_k)\,\mathbf{r} : \mathbf{r} \in \mathcal{R}^2\}, \tag{4.1}$$

where $(\mathbf{u}_k, \mathbf{v}_k)$ stands for a matrix whose columns are $\mathbf{u}_k$ and $\mathbf{v}_k$. Thus at the kth iteration we may consider just the 2-dimensional linear manifold $\mathcal{S}_k$ instead of the whole space $\mathcal{R}^n$.

Define $\mathcal{E}_k = \mathcal{E} \cap \mathcal{S}_k$ to be the 2-*dimensional reduced ellipsoid of* $\mathcal{E}$ and $\Omega(\mathcal{E}_k) = \Omega(\mathcal{E}) \cap \mathcal{S}_k$ to be the boundary of $\mathcal{E}_k$. A direct extension of the Lin–Han algorithm is to construct a 2-dimensional inside ball

$$\mathcal{B}_2(\gamma_k) = \{\|\mathbf{x} - \mathbf{c}_k\| \le \gamma_k \|\mathbf{u}_k\| : \mathbf{x} \in \mathcal{S}_k\} \subset \mathcal{E}_k. \tag{4.2}$$

In addition, the numerical analysis of the Lin–Han algorithm in the previous section suggests that the larger γ_k, the more efficient the algorithm. Therefore it is natural for us to choose the maximum 2-dimensional inside ball and propose the following algorithm.

ALGORITHM 1 (maximal 2-dimensional inside ball algorithm). *At step* 2 *of the General Algorithm, calculate the maximal* γ_k *such that* (4.2) *holds.*

Since the dimension of the ellipsoid $\mathcal{E}_k$ is only two, we can directly calculate the radius of the maximum inside ball of $\mathcal{E}_k$ at $\mathbf{x}_k$ and then decide the value of γ_k in the above algorithm. To do this, we orthonormalize the vectors $\mathbf{v}_k$ and $\mathbf{u}_k$ as follows:

$$\mathbf{p}_k = \frac{\mathbf{v}_k}{\|\mathbf{v}_k\|}, \quad \mathbf{q}_k = \frac{\mathbf{z}_k}{\|\mathbf{z}_k\|}, \tag{4.3}$$

where $\mathbf{z}_k = \mathbf{u}_k - \frac{\mathbf{u}_k^T\mathbf{v}_k}{\mathbf{v}_k^T\mathbf{v}_k}\mathbf{v}_k$. Denote

$$H_k = (\mathbf{p}_k, \mathbf{q}_k) \in \mathcal{R}^{n\times 2}, \tag{4.4}$$

which satisfies $H_k^T H_k = I$. The linear manifold $\mathcal{S}_k$ in (4.1) can be expressed by

$$\mathcal{S}_k = \{\mathbf{x}_k + H_k\mathbf{l} : \mathbf{l} \in \mathcal{R}^2\}. \tag{4.5}$$

Consequently, the 2-dimensional reduced ellipsoid $\mathcal{E}_k$ can be expressed in the vector $\mathbf{l}$ as follows:

$$\mathcal{E}_k^{(l)} = \{\mathbf{l} \in \mathcal{R}^2 : \mathbf{l}^T A_k \mathbf{l} + 2\mathbf{b}_k^T\mathbf{l} \le 0\}, \tag{4.6}$$

where

$$A_k = H_k^T A H_k = \begin{pmatrix} \frac{\mathbf{v}_k^T A\mathbf{v}_k}{\|\mathbf{v}_k\|^2} & \frac{\mathbf{v}_k^T A\mathbf{z}_k}{\|\mathbf{v}_k\|\,\|\mathbf{z}_k\|} \\ \frac{\mathbf{z}_k^T A\mathbf{v}_k}{\|\mathbf{v}_k\|\,\|\mathbf{z}_k\|} & \frac{\mathbf{z}_k^T A\mathbf{z}_k}{\|\mathbf{z}_k\|^2} \end{pmatrix}, \quad \mathbf{b}_k = H_k^T\mathbf{u}_k = \begin{pmatrix} \frac{\mathbf{u}_k^T\mathbf{v}_k}{\|\mathbf{v}_k\|} \\ \frac{\mathbf{u}_k^T\mathbf{z}_k}{\|\mathbf{z}_k\|} \end{pmatrix}. \tag{4.7}$$

At the same time, $\mathbf{x}_k$ corresponds to the origin in the $\mathbf{l}$ subspace. Our problem is then to compute the radius r_k of the maximal inside ball of the ellipsoid $\mathcal{E}_k^{(\mathbf{l})}$ at the origin.

To this aim, for any $t > 0$ we consider the ball

$$\mathcal{B}_2^{(l)}(t) = \{\mathbf{l} \in \mathcal{R}^2 : \|\mathbf{l} + t\mathbf{b}_k\| \le t\|\mathbf{b}_k\|\}$$

that is tangent with the boundary of $\mathcal{E}_k^{(l)}$ at the origin. For any $\mathbf{l}$ on the boundary of $\mathcal{B}_2^{(l)}(t)$, we have that $\|\mathbf{l} + t\mathbf{b}_k\|^2 = t^2\|\mathbf{b}_k\|^2$ and hence

$$\mathbf{l}^T\mathbf{l} + 2t\mathbf{b}_k^T\mathbf{l} = 0. \tag{4.8}$$

If $t \le (\rho(A_k))^{-1}$, we can get by this, (4.8), and (4.6) that

$$\mathbf{l}^T A_k\mathbf{l} + 2\mathbf{b}_k^T\mathbf{l} = \mathbf{l}^T A_k\mathbf{l} - t^{-1}\,\mathbf{l}^T\mathbf{l} \le \mathbf{l}^T A_k\mathbf{l} - \rho(A_k)\mathbf{l}^T\mathbf{l} \le 0,$$

which means $\mathbf{l} \in \mathcal{E}_k^{(l)}$ and hence $r_k \ge (\rho(A_k))^{-1}\|\mathbf{b}_k\|$. On the other hand, for any $t > (\rho(A_k))^{-1}$, consider the point

$$\bar{\mathbf{l}} = -2t(\mathbf{b}_k^T\bar{\mathbf{u}})\bar{\mathbf{u}},$$

where $\bar{\mathbf{u}}$ is one unit eigenvector of the matrix A_k corresponding to $\rho(A_k)$. By direct check, we know that

$$\bar{\mathbf{l}} \in \mathcal{B}_2^{(l)}(t) \quad \text{but} \quad \bar{\mathbf{l}} \notin \mathcal{E}_k.$$

Hence we also have that $r_k \leq (\rho(A_k))^{-1}\|\mathbf{b}_k\|$. To sum up, $r_k = (\rho(A_k))^{-1}\|\mathbf{b}_k\|$ is exactly the radius of the maximal inside ball of $\mathcal{B}_2^{(\mathbf{l})}(r)$ of $\mathcal{E}_k$ at $\mathbf{x}_k$.

By direct calculations, we know that the spectral radius of the matrix A_k is

$$(4.9) \qquad \rho(A_k) = \frac{1}{2}\left[\frac{\mathbf{v}_k^T A \mathbf{v}_k}{\mathbf{v}_k^T \mathbf{v}_k} + \frac{\mathbf{z}_k^T A \mathbf{z}_k}{\mathbf{z}_k^T \mathbf{z}_k} + \sqrt{\left(\frac{\mathbf{v}_k^T A \mathbf{v}_k}{\mathbf{v}_k^T \mathbf{v}_k} - \frac{\mathbf{z}_k^T A \mathbf{z}_k}{\mathbf{z}_k^T \mathbf{z}_k}\right)^2 + \frac{4\,(\mathbf{v}_k^T A \mathbf{z}_k)^2}{\mathbf{v}_k^T \mathbf{v}_k\, \mathbf{z}_k^T \mathbf{z}_k}}\,\right].$$

On the other hand, we have by $H_k^T H_k = I$ that $\|\mathbf{b}_k\| = \|H_k^T \mathbf{u}_k\| = \|\mathbf{u}_k\|$. Thus the spectral radius of the maximal 2-dimensional inside ball of $\mathcal{E}_k$ at $\mathbf{x}_k$ is $(\rho(A_k))^{-1}\,\|\mathbf{u}_k\|$, and the value of γ_k in Algorithm 1 is

$$(4.10) \qquad \gamma_k = (\rho(A_k))^{-1}.$$

In the implementation of Algorithm 1, we need not store and compute the vectors $\mathbf{p}_k$ and $\mathbf{q}_k$ since only the value $\rho(A_k)$ is required. We counted that Algorithm 1 requires 1 matrix-vector multiplication and 12 vector-vector operations or scalar-vector multiplications. (Here note that to calculate $\rho(A_k)$ by (4.9), we can obtain $A\mathbf{z}_k$ and $A\mathbf{v}_k$ by $A\mathbf{u}_k$ and $\mathbf{u}_k$ and hence only require one matrix-vector multiplication, that is $A\mathbf{u}_k$, at each iteration.) Comparing with the Lin–Han algorithm, Algorithm 1 requires only 1 more vector-vector operation. However, Algorithm 1 avoids the direct estimate of the spectral radius $\rho(A)$. Even if $\rho(A)$ is available, we may expect that Algorithm 1 is better than the Lin–Han algorithm with $\gamma_k = (\rho(A))^{-1}$ because it follows from $A_k = H_k^T A H_k$ and $H_k^T H_k = I$ that

$$(4.11) \qquad \rho(A_k)^{-1} \geq \rho(A)^{-1}.$$

Example 1 in section 3 has been used for a quick check, and it is found that Algorithm 1 requires only 111 iterations to achieve a solution with the same precision. More numerical comparisons will be provided in section 8.

5. Sequential 2-dimensional projection algorithm.

A numerical drawback of Algorithm 1 is that, even in the case of two dimensions, if the ellipsoid $\mathcal{E}$ is flat and the point $\mathbf{a}$ to be projected is close to the sharp area, the algorithm may take a large number of iterations. Consider the following example.

Example 2. $\mathcal{E} = \{\mathbf{x} \in \mathcal{R}^2 : x_1^2 + 10000x_2^2 = 2\}$, $\mathbf{a} = (1,\ 100.01)^T$. The initial point $\mathbf{x}_0$ is either $(\sqrt{2},\ 0)^T$ or $(-1,\ 0.01)^T$. The exact projection of $\mathbf{a}$ onto $\mathcal{E}$ is $\mathbf{x}^* = (1,\ 0.01)^T$.

If $\mathbf{x}_0 = (\sqrt{2},\ 0)^T$, Algorithm 1 takes 7361 iterations to reach the stopping condition (2.8) with $\varepsilon = 10^{-6}$. If $\mathbf{x}_0 = (-1,\ 0.01)^T$, Algorithm 1 requires 12,858 iterations to find a satisfactory point. In this example, we have that $\gamma_k \equiv 10^{-4}$. This drawback of Algorithm 1 still exists in the higher-dimensional case. Take the 10-dimensional example in section 3 as an instance. Denoting by M_k the matrix with columns formed by $\mathbf{e}_{k+i} = \frac{\mathbf{x}_{k+i} - \mathbf{x}^*}{\|\mathbf{x}_{k+i} - \mathbf{x}^*\|}$ $(i = 0, 1, 2)$, we found that the determinant of $M_k^T M_k$ eventually tends to zero, which means that the solution procedure of Algorithm 1 tends to some 2-dimensional reduced ellipsoid. At the same time, the γ_k tends monotonically increasingly to some value, which is approximately $1.2804e{-}2$.

To overcome the above drawback of Algorithm 1, we propose another algorithm that consists of calculating the exact projection of $\mathbf{a}$ onto the 2-dimensional ellipsoid $\mathcal{E}_k$ at each iteration. The following lemma, together with the invariance of the projection under orthogonal transformations, indicates that the projection onto any 2-dimensional ellipsoid can be obtained via a quartic equation.

LEMMA 1. *Consider the* 2*-dimensional ellipsoid*

$$\mathcal{E}^{(h)} = \{\mathbf{h} \in \mathcal{R}^2 : \mathbf{h}^T D\mathbf{h} \le \beta\}, \tag{5.1}$$

where $\beta > 0$ *and* $D = \mathrm{diag}(\lambda_1, \lambda_2)$ *with* $\lambda_1, \lambda_2 > 0$*. For any* $\mathbf{h} = (h_1, h_2)^T$ *with* $\mathbf{h}^T D\mathbf{h} > \beta$*, denote by* $\mathbf{h}^* = (h_1^*, h_2^*)^T$ *the projection of* $\mathbf{h}$ *onto* $\mathcal{E}^{(h)}$*. Then*

$$h_2^* = \frac{\lambda_1 h_2}{(\lambda_1 - \lambda_2)h_1^* + \lambda_2 h_1}\, h_1^*, \tag{5.2}$$

where h_1^* *satisfies the quartic equation*

$$\left[(\lambda_1 - \lambda_2)h_1^* + \lambda_2 h_1\right]^2 \left[\lambda_1 (h_1^*)^2 - \beta\right] + \lambda_1^2 \lambda_2 h_2^2 (h_1^*)^2 = 0. \tag{5.3}$$

Proof. By the Karush–Kuhn–Tucker condition (for example, see Fletcher [5]), there exists some $\mu > 0$ such that $\mathbf{h} - \mathbf{h}^* = \mu D\mathbf{h}^*$, or equivalently,

$$\begin{cases} h_1 - h_1^* = \mu\lambda_1 h_1^*, \\ h_2 - h_2^* = \mu\lambda_2 h_2^*. \end{cases} \tag{5.4}$$

It follows that

$$\lambda_1 h_1^*(h_2 - h_2^*) = \lambda_2 h_2^*(h_1 - h_1^*), \tag{5.5}$$

which implies the truth of (5.2). In addition, by the feasibility condition,

$$\lambda_1 (h_1^*)^2 + \lambda_2 (h_2^*)^2 = \beta. \tag{5.6}$$

Substituting (5.2) into (5.6), we then know that h_1^* satisfies (5.3). □

The quartic equation (5.3) can be solved in an analytical way or easily by some numerical methods (in our implementation with MATLAB, we use the function *roots*). From (5.4) and the positiveness of the multiplier μ, we can get that

$$\mu = \frac{h_1 - h_1^*}{\lambda_1 h_1^*} = \frac{h_2 - h_2^*}{\lambda_2 h_2^*} > 0. \tag{5.7}$$

The above relations and (5.2) can help us to pick up the correct value for h_1^* among the four roots of (5.3). The h_2^* is then determined by (5.2).

Note that the computational burden of projecting a point onto a 2-dimensional ellipsoid is negligible when n is relatively large. We propose the following algorithm for projecting onto an n-dimensional ellipsoid.

ALGORITHM 2 (sequential 2-dimensional projection algorithm). *At the kth iteration, having* $\mathbf{x}_k \in \Omega(\mathcal{E})$*, we take the projection of* $\mathbf{a}$ *on the* 2*-dimensional reduced ellipsoid* $\mathcal{E}_k = \mathcal{E} \cap \mathcal{S}_k$ *to be* $\mathbf{x}_{k+1}$*.*

Now we describe how to calculate $\mathbf{x}_{k+1}$ in Algorithm 2. Notice that the linear manifold $\mathcal{S}_k$ in (4.1) can be expressed by (4.5), where A_k and $\mathbf{b}_k$ are still given by (4.7), (4.4), and (4.3). Also notice that the point $\mathbf{a}$ in $\mathcal{R}^n$ corresponds to $\mathbf{a}_l = (\|\mathbf{v}_k\|, 0)^T$

in the $\mathbf{l}$ space. Due to the invariance property of the projection under orthogonal transformations and the fact that $H_k^T H_k = I$, if the projection $\mathbf{a}_l^*$ of $\mathbf{a}_l$ onto the ellipsoid $\mathcal{E}_k^{(l)}$ in (4.6) is obtained, the $\mathbf{x}_{k+1}$ in Algorithm 2 is given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + H_k \mathbf{a}_l^*. \tag{5.8}$$

Therefore our calculation of $\mathbf{x}_{k+1}$ in Algorithm 2 can be divided into two steps: the first step is to compute the projection $\mathbf{a}_l^*$ of $\mathbf{a}_l = (\|\mathbf{v}_k\|, 0)^T$ onto $\mathcal{E}_k^{(\mathbf{l})}$ in (4.6), and the second step is to calculate $\mathbf{x}_{k+1}$ from $\mathbf{a}_l^*$.

At the first step, to compute $\mathbf{a}_l^*$, we assume that the eigendecomposition of the 2×2 matrix A_k in (4.7) is

$$A_k = Q^T D Q, \quad \text{where } D \text{ is diagonal and } Q^T Q = I. \tag{5.9}$$

Under the orthogonal transformation $\mathbf{l} \to \mathbf{h} = Q\mathbf{l} + D^{-1}Q\mathbf{b}$, the ellipsoid $\mathcal{E}_k^{(\mathbf{l})}$ can be expressed in the form (5.1) with $\beta = (Q\mathbf{b}_k)^T D^{-1} (Q\mathbf{b}_k)$. The point $\mathbf{a}_l$ in the $\mathbf{l}$ space corresponds to $\mathbf{a}_h = Q\mathbf{a}_l + D^{-1}Q\mathbf{b}_k$ in the $\mathbf{h}$ space. Denote by $\mathbf{a}_h^*$ the projection of $\mathbf{a}_h$ onto $\mathcal{E}^{(h)}$; we also have that $\mathbf{a}_h^* = Q\mathbf{a}_l^* + D^{-1}Q\mathbf{b}_k$. Consequently, we have that

$$\mathbf{a}_l^* = Q^T(\mathbf{a}_h^* - D^{-1}Q\mathbf{b}_k). \tag{5.10}$$

By Lemma 1, the projection $\mathbf{a}_h^*$ of $\mathbf{a}_h$ onto $\mathcal{E}^{(h)}$ can be obtained via a quartic equation. Therefore after A_k and $\mathbf{b}_k$ have been obtained, the computational work to obtain the projection $\mathbf{a}_l^*$ of $\mathbf{a}_l$ onto the 2-dimensional ellipsoid $\mathcal{E}_k^{(l)}$ is again negligible when n is relatively large.

At the second step, we may calculate $\mathbf{x}_{k+1}$ from $\mathbf{a}_l^*$ directly by (5.8). To avoid the explicit storage of H_k, however, we express $\mathbf{x}_{k+1}$ in the form (2.5) and treat Algorithm 2 as a special case of the General Algorithm described in section 3. Assume that $\mathbf{a}_l^* = (a_{l,1}^*, a_{l,2}^*)^T$. It follows from (5.8), the definition of H_k, and (4.3) that

$$\mathbf{x}_{k+1} = \mathbf{x}_k + a_{l,1}^*\mathbf{p}_k + a_{l,2}^*\mathbf{q}_k = \mathbf{x}_k + \frac{a_{l,2}^*}{\|\mathbf{z}_k\|}\mathbf{u}_k + \left[\frac{a_{l,1}^*}{\|\mathbf{v}_k\|} - \frac{\mathbf{u}_k^T\mathbf{v}_k}{\mathbf{v}_k^T\mathbf{v}_k}\frac{a_{l,2}^*}{\|\mathbf{z}_k\|}\right]\mathbf{v}_k. \tag{5.11}$$

On the other hand, the definitions of $\mathbf{w}_k$, $\mathbf{c}_k$, and $\mathbf{v}_k$ in section 2 indicate that

$$\mathbf{w}_k = \mathbf{c}_k - \mathbf{a} = \mathbf{x}_k - \gamma_k\,\mathbf{u}_k - \mathbf{a} = -\gamma_k\,\mathbf{u}_k - \mathbf{v}_k. \tag{5.12}$$

By (5.12) and the definition of $\mathbf{v}_k$, the $\mathbf{x}_{k+1}$ in (2.5) can be expressed as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{v}_k + \eta_k\,\mathbf{w}_k = \mathbf{x}_k - \eta_k\,\gamma_k\,\mathbf{u}_k + (1-\eta_k)\,\mathbf{v}_k. \tag{5.13}$$

Comparing (5.11) and (5.13), we can then calculate $\mathbf{x}_{k+1}$ by (2.5) and (5.12) with

$$\eta_k = 1 - \frac{a_{l,1}^*}{\|\mathbf{v}_k\|} + \frac{\mathbf{u}_k^T\mathbf{v}_k}{\mathbf{v}_k^T\mathbf{v}_k}\frac{a_{l,2}^*}{\|\mathbf{z}_k\|} \tag{5.14}$$

and

$$\gamma_k = -\frac{a_{l,2}^*}{\|\mathbf{z}_k\|}\frac{1}{\eta_k}. \tag{5.15}$$

The above equivalent treatment also facilitates us in designing a safeguard for Algorithm 2. If the γ_k in (5.15) is negative or tiny (this is sometimes the case in our

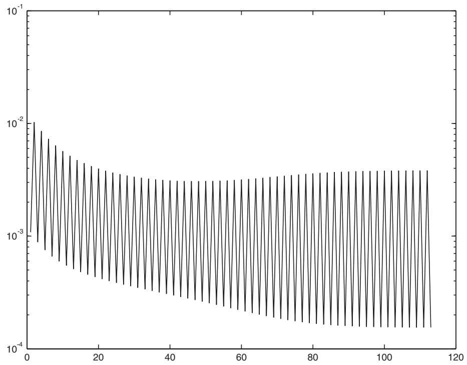


FIG. 2. *The* $\{\gamma_k\}$ *by Algorithm* 2.

numerical experiments, although seldom), we can turn to using (4.10) and carry out one step by the maximal 2-dimensional inside ball algorithm.

Using (4.4) and (4.3), the matrix A_k and $\mathbf{b}_k$ in the expression of $\mathcal{E}_k^{(l)}$ can be also calculated without the explicit storage of H_k. If we do not consider the difference in the calculation of γ_k, the computation amount of Algorithm 2 and that of Algorithm 1 are identical since they require the same vector operations; namely, both of them require only 1 matrix-vector multiplication and 12 vector-vector operations or scalar-vector multiplications. Due to the minimal property, however, Algorithm 2 is expected to perform better than 1. When we use Algorithm 2 to solve the example in section 3, a solution with the same precision is achieved at the 79th iteration.

6. Hybrid projection algorithms. Algorithms 1 and 2 avoid the direct estimate to the spectral radius $\rho(A)$ and are more efficient than the Lin–Han algorithm. Algorithm 2 seems to be optimal since the distance function $d(\mathbf{a}, \mathbf{x})$ achieves the maximal decrease in the 2-dimensional ellipsoid $\mathcal{S}_k$ at every iteration. However, the following example shows that Algorithm 2 produces some kind of zigzags. Consider the 3-dimensional example that follows.

Example 3. $\mathcal{E} = \{\mathbf{x} \in \mathcal{R}^3 : \, x_1^2 + 100\, x_2^2 + 10000\, x_3^2 = 1.0101\}$, $\mathbf{a} = (2,\, 1.01,\, 1.0001)^T$. The initial point is $\mathbf{x}_0 = (0.01,\, 0.01,\, 0.01)^T$. The exact projection of $\mathbf{a}$ onto $\mathcal{E}$ is $\mathbf{x}^* = (1,\, 0.01,\, 0.0001)^T$.

Even for this 3-dimensional example, Algorithm 2 takes 113 iterations to reach the stopping condition (2.8) with $\varepsilon = 10^{-6}$. Denote again $\mathbf{e}_k = \frac{\mathbf{x}_k - \mathbf{x}^*}{\|\mathbf{x}_k - \mathbf{x}^*\|}$ and the matrices

$$\bar{M}_k = [\mathbf{e}_{2k},\, \mathbf{e}_{2k+2},\, \mathbf{e}_{2k+4}], \qquad \tilde{M}_k = [\mathbf{e}_{2k+1},\, \mathbf{e}_{2k+3},\, \mathbf{e}_{2k+5}].$$

We found that both the determinants of $\bar{M}_k$ and $\tilde{M}_k$ tend to zero as k increases. This shows that the iterations generalized by Algorithm 2 tend to two 2-dimensional reduced ellipsoids alternately. Meanwhile, the sequence $\{\gamma_k\}$ also tends to two different values, as shown in Figure 2. The same phenomenon is observed for Algorithm 2 in Example 1 in section 3.

Instead of establishing strict theoretical results for the above observations, we are interested in this paper in finding more efficient algorithms. A naive idea to avoid the zigzagging phenomenon of Algorithm 2 is to use one iteration of Algorithm 1 after every two iterations of Algorithm 2. We then obtain the following simple hybrid projection algorithm, where for some given $\mathbf{x}_k$, $\gamma_k^{(1)}$ and $\gamma_k^{(2)}$ stand for the values of

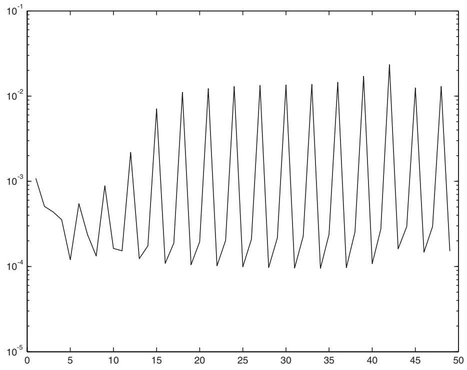


FIG. 3. *The $\{\gamma_k\}$ by Algorithm* 3.

γ_k given by Algorithm 1 and Algorithm 2, respectively.

ALGORITHM 3 (simple hybrid projection algorithm). *At step* 2 *in the General Algorithm, set $\gamma_{3k+i} = \gamma^{(2)}_{3k+i}$ for $i = 0$, 1 and $\gamma_{3k+2} = \gamma^{(1)}_{3k+2}$.*

Although its basic idea is simple, our numerical experiments on a collection of test problems showed that Algorithm 3 is much more efficient than Algorithms 1 and 2. For instance, to solve Example 3 with the same precision, Algorithm 3 requires only 49 iterations, which is significantly smaller than the number required by Algorithm 2. To solve Example 1 in section 3, Algorithm 3 takes only 27 iterations. At the same time, we notice that the value of γ_k changes frequently in Algorithm 3 (see Figure 3 for $\{\gamma_k\}$ in Example 3).

A possible explanation for the success of Algorithm 3 is that the inexactness in the projection of $\mathbf{a}$ on the 2-dimensional reduced ellipsoid $\mathcal{E}_{3k+2}$ introduced by Algorithm 1 can make $\mathcal{E}_{3k+3}$ and $\mathcal{E}_{3k+4}$ be much different from $\mathcal{E}_{3k}$ and $\mathcal{E}_{3k+1}$, and hence help Algorithm 2 continuously achieve big decreases in the distance function $d(\mathbf{a}, \mathbf{x}_k)$. It is interesting to note that a similar idea has been used in the steepest descent method and leads to significant numerical improvement (see [3]).

The degree of inexactness in Algorithm 3 depends on the values of $\gamma^{(1)}_{3k+2}$ and $\gamma^{(2)}_{3k+2}$. If $\gamma^{(1)}_{3k+2} \approx \gamma^{(2)}_{3k+2}$, Algorithm 3 fails to bring enough inexactness. On the other hand, we see that there are many other ways to control the inexactness (for example, by multiplying $\gamma^{(2)}_k$ by some positive constant less than 1). In addition, from Figure 3 we have some worry that the sequences $\{\gamma_k\}$ and $\{\mathbf{e}_k\}$ in Algorithm 3 may also sink into some type of cycle. Therefore we propose the following general hybrid projection algorithm.

ALGORITHM 4 (general hybrid projection algorithm). *At step* 2 *of the General Algorithm, compute γ_k by some positive function $\psi(\gamma^{(1)}_k, \gamma^{(2)}_k)$ of $\gamma^{(1)}_k$ and $\gamma^{(2)}_k$.*

The above algorithm includes Algorithms 1, 2, and 3 as its components. Now we discuss how to choose the function ψ. To guarantee the existence of $\mathbf{x}_{k+1}$ and $d(\mathbf{a}, \mathbf{x}_{k+1}) < d(\mathbf{a}, \mathbf{x}_k)$, we impose the following condition:

$$\gamma_k \le \max(\gamma^{(1)}_k, \gamma^{(2)}_k). \tag{6.1}$$

If this relation holds, it is easy to know by continuity that the line segment connecting $\mathbf{a}$ and $\mathbf{c}_k = \mathbf{x}_k - \gamma_k \mathbf{u}_k$ must have an intersection point with $\Omega(\mathcal{E}_k)$. By (6.1) and the positivity of ψ, we can express γ_k as

$$\gamma_k = c_k^{(1)}\gamma_k^{(1)} + c_k^{(2)}\gamma_k^{(2)}, \tag{6.2}$$

where $c_k^{(1)}$ and $c_k^{(2)}$ are such that

$$c_k^{(1)} \geq 0, \quad c_k^{(2)} \geq 0, \quad c_k^{(1)} + c_k^{(2)} \leq 1. \tag{6.3}$$

To ensure the convergence of the algorithm, we require that

$$c_k^{(1)} + c_k^{(2)} \geq \tau \quad \text{for some } \tau \in (0,\, 1] \text{ and all } k. \tag{6.4}$$

Under these requirements on $c_k^{(1)}$ and $c_k^{(2)}$, we will show in the next section that the algorithm is globally convergent and the convergence is linear.

In this paper, we are particularly interested in the following 4-parameter family of hybrid projection algorithms:

$$\gamma_k = \begin{cases} \gamma_k^{(2)} & \text{if } \mathrm{mod}(k, m_1 + m_2) < m_1, \\ c_1\gamma_k^{(1)} + c_2\gamma_k^{(2)} & \text{otherwise,} \end{cases} \tag{6.5}$$

where $m_1 \geq 1$ and $m_2 \geq 1$ are integers and c_1 and c_2 are nonnegative constants satisfying $0 < c_1 + c_2 \leq 1$. The formula (6.5) indicates that the algorithm will carry out m_2 inexact 2-dimensional projection steps after every m_1 steps of Algorithm 2. In section 8, we will find that some methods in the family (6.5) are more efficient than the simple hybrid projection algorithm. Here we would like to note that Algorithm (6.5) with the choice (8.2) requires only 23 iterations for Example 1.

7. Linear convergence. Lin and Han [9] proved global convergence for their algorithm under the condition (3.2) on γ_k. In the following we will establish the linear convergence of the general hybrid projection algorithm with γ_k given by (6.2) under the assumptions (6.3) and (6.4) on $c_1^{(k)}$ and $c_2^{(k)}$. Consequently, the Lin–Han algorithm and Algorithms 1–3 are all linearly convergent.

For any nonzero vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathcal{R}^n$, define the angle between $\mathbf{x}$ and $\mathbf{y}$ as

$$\theta(\mathbf{x}, \mathbf{y}) = \arccos\left(\frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}\right), \quad 0 \leq \theta(\mathbf{x}, \mathbf{y}) \leq \pi. \tag{7.1}$$

For any $\mathbf{x}_k \in \mathcal{S}(\mathbf{a}, \mathcal{E})$, we denote the angles

$$\nu_k = \theta(\mathbf{a} - \mathbf{x}_k, A\mathbf{x}_k + \mathbf{b}), \quad \theta_k = \theta(\mathbf{a} - \mathbf{x}_k, \mathbf{x}^* - \mathbf{a}), \tag{7.2}$$

where $\mathbf{x}^*$ is the projection of $\mathbf{a}$ on $\mathcal{E}$ as before. In the following, Lemmas 2 and 3 aim to provide a lower bound for the decrease $d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}_{k+1})$ by the angle ν_k. Lemma 5, which calls Lemma 4, estimates the upper bound for the distance $d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}^*)$ by the angle θ_k. Then using the relation $\nu_k \geq \theta_k$, as shown in Lemma 6, we establish the linear convergence of the general hybrid projection algorithm in Theorem 7.

Denote by κ and $\lambda_{min}(A)$ the condition number and the minimal eigenvalue of A, respectively. Define

$$\bar{\alpha} = \alpha + \mathbf{b}^T A^{-1}\mathbf{b}. \tag{7.3}$$

Under the transformation $\mathbf{x} \to \mathbf{y} = \mathbf{x} + A^{-1}\mathbf{b}$, we can express $\Omega(\mathcal{E})$ as

$$\bar{\Omega} = \{\mathbf{y} \in \mathcal{R}^n : \ \mathbf{y}^T A\mathbf{y} = \bar{\alpha}\}. \tag{7.4}$$

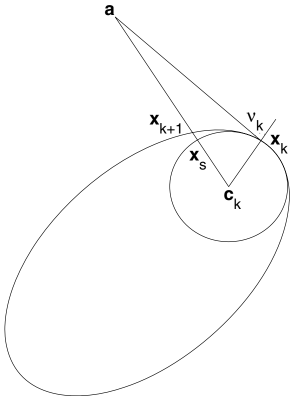


FIG. 4. *Diagram of the proof of Lemma* 2.

Then we can obtain

$$\begin{aligned}\min_{\mathbf{x}\in\Omega(\mathcal{E})}\|A\mathbf{x}+\mathbf{b}\| &= \min_{\mathbf{y}\in\bar{\Omega}}\|A\mathbf{y}\| = \min_{\mathbf{y}\in\bar{\Omega}}\sqrt{(A^{\frac{1}{2}}\mathbf{y})^T A(A^{\frac{1}{2}}\mathbf{y})}\\ &= \min_{\mathbf{y}\in\bar{\Omega}}\sqrt{\lambda_{\min}(A)\,\|A^{\frac{1}{2}}\mathbf{y}\|} = \sqrt{\lambda_{\min}(A)\,\bar{\alpha}}. \qquad (7.5)\end{aligned}$$

LEMMA 2. *For Algorithm* 1*, there exists some positive constant* c_3 *such that*

$$d(\mathbf{a},\mathbf{x}_k) - d(\mathbf{a},\mathbf{x}_{k+1}) \geq c_3 \sin^2\frac{\nu_k}{2} \quad \textit{for all } k, \qquad (7.6)$$

where

$$c_3 = \frac{2\,d(\mathbf{a},\,\mathbf{x}^*)}{1+c_4\,d(\mathbf{a},\,\mathbf{x}^*)} \qquad \textit{and} \qquad c_4 = \left(\frac{\rho(A)}{\bar{\alpha}}\right)^{\frac{1}{2}}\kappa^{\frac{1}{2}}. \qquad (7.7)$$

Proof. Denote by $\mathbf{x}_s$ the intersection of the line segment $\mathcal{L}(\mathbf{a},\mathbf{c}_k)$ and the boundary of the 2-dimensional ball $\mathcal{B}_2(\gamma_k)$ in (4.2) (see Figure 4). Then we have that $\|\mathbf{x}_s - \mathbf{c}_k\| = \|\mathbf{x}_k - \mathbf{c}_k\| = \gamma_k\,\|\mathbf{u}_k\|$. Noting that $\mathcal{B}_2(\gamma_k) \subset \mathcal{E}_k$ and considering the triangle formed by the points $\mathbf{a}$, $\mathbf{c}_k$, and $\mathbf{x}_k$, we can get that

$$\begin{aligned} d(\mathbf{a},\mathbf{x}_k) - d(\mathbf{a},\mathbf{x}_{k+1}) &= [d(\mathbf{a},\mathbf{x}_k) + \|\mathbf{x}_k - \mathbf{c}_k\|] - [d(\mathbf{a},\mathbf{x}_{k+1}) + \|\mathbf{x}_s - \mathbf{c}_k\|]\\ &\geq [d(\mathbf{a},\mathbf{x}_k) + \|\mathbf{x}_k - \mathbf{c}_k\|] - \|\mathbf{a}-\mathbf{c}_k\|\\ &\geq \frac{[d(\mathbf{a},\mathbf{x}_k) + \|\mathbf{x}_k - \mathbf{c}_k\|]^2 - \|\mathbf{a}-\mathbf{c}_k\|^2}{2\,[d(\mathbf{a},\mathbf{x}_k) + \|\mathbf{x}_k - \mathbf{c}_k\|]}\\ &= \frac{d(\mathbf{a},\mathbf{x}_k)\,\|\mathbf{x}_k - \mathbf{c}_k\|\,[1-\cos(\pi-\nu_k)]}{d(\mathbf{a},\mathbf{x}_k) + \|\mathbf{x}_k - \mathbf{c}_k\|}\\ &= \frac{2\,\sin^2\frac{\nu_k}{2}}{[d(\mathbf{a},\mathbf{x}_k)]^{-1} + [\gamma_k\|\mathbf{u}_k\|]^{-1}}. \qquad (7.8)\end{aligned}$$

From the above relation, (4.10), (4.11), the definition of $\mathbf{u}_k$, and (7.5), we know that (7.6) holds. □

LEMMA 3. *Consider Algorithm* 4 *with* γ_k *given by* (6.2)*. If* $c_1^{(k)}$ *and* $c_2^{(k)}$ *satisfy* (6.3) *and* (6.4)*, we have that*

$$d(\mathbf{a},\mathbf{x}_k) - d(\mathbf{a},\mathbf{x}_{k+1}) \geq c_3\,\tau\,\sin^2\frac{\nu_k}{2} \quad \textit{for all } k. \qquad (7.9)$$

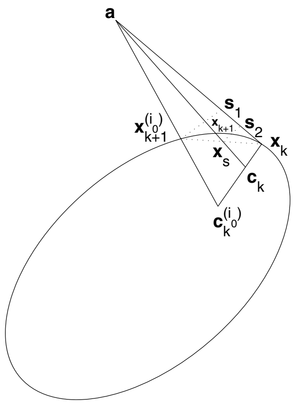


FIG. 5. *Diagram of proof of Lemma* 3.

Proof. For any fixed $\mathbf{x}_k$, denote by $\mathbf{x}_{k+1}^{(1)}$ and $\mathbf{x}_{k+1}^{(2)}$ the points generated by Algorithms 1 and 2, respectively. Noting that $d(\mathbf{a}, \mathbf{x}_{k+1}^{(2)}) = \min\{d(\mathbf{a}, \mathbf{x}) : \mathbf{x} \in \mathcal{E}_k\} \le d(\mathbf{a}, \mathbf{x}_{k+1}^{(1)})$, we have by this and Lemma 2 that

$$d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}_{k+1}^{(i)}) \ge c_3 \sin^2 \frac{\nu_k}{2} \quad \text{for } i = 1, 2 \text{ and all } k. \tag{7.10}$$

The relations (6.2) and (6.3) imply that $\gamma_k \le \max(\gamma_k^{(1)}, \gamma_k^{(2)})$. Define $\mathbf{x}_{k+1}(\gamma) = \mathcal{L}(\mathbf{a}, \mathbf{x}_k - \gamma \mathbf{u}_k) \cap \Omega(\mathcal{E}_k)$. Since $\gamma_k^{(2)}$ is the unique minimizer of the function $d(\mathbf{a}, \mathbf{x}_{k+1}(\gamma))$ such that $\mathbf{x}_{k+1}(\gamma) \subset \Omega(\mathcal{E}_k)$, we know that $d(\mathbf{a}, \mathbf{x}_{k+1}(\gamma))$ is monotonically decreasing as γ moves from $\gamma_k^{(1)}$ to $\gamma_k^{(2)}$. Consequently, if

$$\gamma_k \ge \min(\gamma_k^{(1)}, \gamma_k^{(2)}), \tag{7.11}$$

we have that $d(\mathbf{a}, \mathbf{x}_{k+1}) \le d(\mathbf{a}, \mathbf{x}_{k+1}^{(1)})$ and hence (7.9) is true.

If (7.11) does not hold, we have by (6.2) and (6.3) that

$$\tau\, \gamma_k^{(i_0)} \le \gamma_k \le \gamma_k^{(i_0)}, \tag{7.12}$$

where $i_0 \in \{0,\, 1\}$ is such that $\gamma_k^{(i_0)} = \min\,(\gamma_k^{(1)}, \gamma_k^{(2)})$. Equivalently, we have that

$$\tau\, \|\mathbf{c}_k^{(i_0)} - \mathbf{x}_k\| \le \|\mathbf{c}_k - \mathbf{x}_k\| \le \|\mathbf{c}_k^{(i_0)} - \mathbf{x}_k\|, \tag{7.13}$$

where $\mathbf{c}_k^{(i_0)} = \mathbf{x}_k - \gamma_k^{(i_0)} \mathbf{u}_k$. Denote again by $\mathbf{x}_s$ the intersection of $\mathcal{L}(\mathbf{a}, \mathbf{c}_k)$ and the line segment connecting $\mathbf{x}_k$ and $\mathbf{x}_{k+1}^{(i_0)}$ (see Figure 5). Due to the convexity of $\Omega(\mathcal{E}_k)$, we know that $\mathbf{x}_s$ belongs to the interior of $\mathcal{E}_k$ and hence

$$d(\mathbf{a}, \mathbf{x}_{k+1}) < d(\mathbf{a}, \mathbf{x}_s). \tag{7.14}$$

For convenience, for any given vectors $\mathbf{z}_1$, $\mathbf{z}_2$, and $\mathbf{z}_3$, we denote by $\angle \mathbf{z}_1\mathbf{z}_2\mathbf{z}_3$ the angle between $\mathbf{z}_1 - \mathbf{z}_2$ and $\mathbf{z}_3 - \mathbf{z}_2$. Note that $\angle \mathbf{a}\mathbf{c}_k\mathbf{x}_k > \angle \mathbf{a}\mathbf{c}_k^{(i_0)}\mathbf{x}_k$. If introducing a supplementary point $\mathbf{s} \subset \mathcal{L}(\mathbf{x}_k, \mathbf{x}_{k+1}^{(i_0)})$ such that $\mathcal{L}(\mathbf{s}, \mathbf{c}_k)$ is parallel to $\mathcal{L}(\mathbf{x}_{k+1}^{(i_0)}, \mathbf{c}_k^{(i_0)})$, we can see that

$$\frac{\|\mathbf{x}_s - \mathbf{x}_k\|}{\|\mathbf{x}_{k+1}^{(i_0)} - \mathbf{x}_k\|} > \frac{\|\mathbf{c}_k - \mathbf{x}_k\|}{\|\mathbf{c}_k^{(i_0)} - \mathbf{x}_k\|}. \tag{7.15}$$

Now we introduce a supplementary point $\mathbf{s}_1 \subset \mathcal{L}(\mathbf{a}, \mathbf{x}_k)$ such that $\|\mathbf{a} - \mathbf{s}_1\| = \|\mathbf{a} - \mathbf{x}_{k+1}^{(i_0)}\|$. Then we have that

$$\angle \mathbf{a}\mathbf{x}_{k+1}^{(i_0)}\mathbf{s}_1 = \angle \mathbf{a}\mathbf{s}_1\mathbf{x}_{k+1}^{(i_0)} = \angle \mathbf{s}_1\mathbf{x}_{k+1}^{(i_0)}\mathbf{x}_k + \angle \mathbf{a}\mathbf{x}_k\mathbf{x}_{k+1}^{(i_0)}, \tag{7.16}$$

$$\angle \mathbf{a}\mathbf{x}_{k+1}^{(i_0)}\mathbf{s}_1 + \angle \mathbf{a}\mathbf{s}_1\mathbf{x}_{k+1}^{(i_0)} + \angle \mathbf{x}_k\mathbf{a}\mathbf{x}_{k+1}^{(i_0)} = \pi. \tag{7.17}$$

Substituting (7.16) into (7.17), we get that

$$2\angle \mathbf{s}_1\mathbf{x}_{k+1}^{(i_0)}\mathbf{x}_k = (\pi - 2\angle \mathbf{a}\mathbf{x}_k\mathbf{x}_{k+1}^{(i_0)}) - \angle \mathbf{x}_k\mathbf{a}\mathbf{x}_{k+1}^{(i_0)}. \tag{7.18}$$

Similarly, if we introduce another supplementary point $\mathbf{s}_2 \subset \mathcal{L}(\mathbf{a}, \mathbf{x}_k)$ such that $\|\mathbf{a} - \mathbf{s}_2\| = \|\mathbf{a} - \mathbf{x}_s\|$, we have that

$$2\angle \mathbf{s}_2\mathbf{x}_{k+1}^{(i_0)}\mathbf{x}_k = (\pi - 2\angle \mathbf{a}\mathbf{x}_k\mathbf{x}_{k+1}^{(i_0)}) - \angle \mathbf{x}_k\mathbf{a}\mathbf{x}_s. \tag{7.19}$$

The relations (7.18), (7.19), and $\angle \mathbf{x}_k\mathbf{a}\mathbf{x}_s < \angle \mathbf{x}_k\mathbf{a}\mathbf{x}_{k+1}^{(i_))}$ imply that $\angle \mathbf{s}_2\mathbf{x}_{k+1}^{(i_0)}\mathbf{x}_k > \angle \mathbf{s}_1\mathbf{x}_{k+1}^{(i_0)}\mathbf{x}_k$. Similarly to (7.15), we can prove that

$$\frac{\|\mathbf{s}_2 - \mathbf{x}_k\|}{\|\mathbf{s}_1 - \mathbf{x}_k\|} > \frac{\|\mathbf{x}_s - \mathbf{x}_k\|}{\|\mathbf{x}_{k+1}^{(i_0)} - \mathbf{x}_k\|}. \tag{7.20}$$

Therefore by (7.14), (7.20), (7.15), and (7.12), we obtain

$$\begin{aligned} \frac{d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}_{k+1})}{d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}_{k+1}^{(i_0)})} &> \frac{d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}_s)}{d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}_{k+1}^{(i_0)})} = \frac{\|\mathbf{s}_2 - \mathbf{x}_k\|}{\|\mathbf{s}_1 - \mathbf{x}_k\|} \\ &> \frac{\|\mathbf{x}_s - \mathbf{x}_k\|}{\|\mathbf{x}_{k+1}^{(i_0)} - \mathbf{x}_k\|} > \frac{\|\mathbf{c}_k - \mathbf{x}_k\|}{\|\mathbf{c}_k^{(i_0)} - \mathbf{x}_k\|} \geq \tau, \end{aligned} \tag{7.21}$$

which, with (7.10), indicates the truth of (7.9). ☐

To estimate the distance between $d(\mathbf{a}, \mathbf{x}_k)$ and $d(\mathbf{a}, \mathbf{x}^*)$, we require the following lemma.

LEMMA 4. *Consider the n-dimensional ellipsoid $\mathcal{E}$ in* (1.2). *For any $\mathbf{x}$, $\mathbf{y} \in \Omega(\mathcal{E})$ with $\mathbf{x} \neq \mathbf{y}$, we have that*

$$\cos\theta(A\mathbf{x} + \mathbf{b}, \mathbf{x} - \mathbf{y}) \leq \frac{1}{2}c_4\|\mathbf{x} - \mathbf{y}\|, \tag{7.22}$$

where c_4 is given in (7.7).

Proof. Without loss of generality, we assume that $n = 2$; otherwise consider the reduced ellipsoid of $\mathcal{E}$ restricted to the 2-dimensional linear manifold $\{\mathbf{x} + (A\mathbf{x} + \mathbf{b}, \mathbf{x} - \mathbf{y})\,\mathbf{r} : \mathbf{r} \in \mathcal{R}^2\}$. Further, by making the transformation $\mathbf{x} \to \mathbf{x} + A^{-1}\mathbf{b}$ and some orthogonal transformation, we assume that

$$\mathcal{E} = \{\mathbf{x} \in R^2 : \mathbf{x}^T A\mathbf{x} \leq \bar{\alpha}\}, \qquad \text{where } A = \mathrm{diag}(\beta^2, \delta^2) \text{ with } 0 < \beta \leq \delta, \tag{7.23}$$

and $\bar{\alpha}$ is still given in (7.3). Let $\varrho = \bar{\alpha}^{\frac{1}{2}}\beta^{-1}\delta^{-1}$. Then we can express any $\mathbf{x}$, $\mathbf{y} \subset \Omega(\mathcal{E})$ as

$$\mathbf{x} = \varrho \begin{pmatrix} \delta\cos\alpha_1, \\ \beta\sin\alpha_1 \end{pmatrix}, \quad \mathbf{y} = \varrho \begin{pmatrix} \delta\cos\alpha_2, \\ \beta\sin\alpha_2 \end{pmatrix}.$$

Denoting $\alpha_3 = \frac{\alpha_1+\alpha_2}{2}$ and $\alpha_4 = \frac{\alpha_1-\alpha_2}{2}$, we have by direct calculations that

$$\begin{aligned}\|\mathbf{x}-\mathbf{y}\|^2 &= \varrho^2\,[\delta^2(\cos\alpha_1-\cos\alpha_2)^2+\beta^2(\sin\alpha_1-\sin\alpha_2)^2]\\ &= 4\varrho^2\sin^2\alpha_4(\delta^2\cos^2\alpha_3+\beta^2\sin^2\alpha_3)\\ &\geq 4\varrho^2\beta^2\sin^2\alpha_4\end{aligned} \tag{7.24}$$

and

$$\begin{aligned}(\mathbf{x}-\mathbf{y})^TA\mathbf{x} &= \varrho^2\beta^2\delta^2[\cos\alpha_1(\cos\alpha_1-\cos\alpha_2)+\sin\alpha_1(\sin\alpha_1-\sin\alpha_2)]\\ &= \varrho^2\beta^2\delta^2[1-(\cos\alpha_1\cos\alpha_2+\sin\alpha_1\sin\alpha_2)]\\ &= \varrho^2\beta^2\delta^2[1-\cos(\alpha_1+\alpha_2)]\\ &= 2\varrho^2\beta^2\delta^2\sin^2\alpha_4.\end{aligned} \tag{7.25}$$

In addition, we can get that

$$\|A\mathbf{x}\| = \varrho\beta\delta\sqrt{\beta^2\cos^2\alpha_1+\delta^2\sin^2\alpha_1} \geq \varrho\beta^2\delta. \tag{7.26}$$

Thus by (7.1), $\mathbf{b}=\mathbf{0}$, (7.24)–(7.26), and the definition of ϱ, we obtain

$$\cos\theta(A\mathbf{x}+\mathbf{b},\mathbf{x}-\mathbf{y}) = \frac{(\mathbf{x}-\mathbf{y})^TA\mathbf{x}}{\|\mathbf{x}-\mathbf{y}\|^2\,\|A\mathbf{x}\|}\,\|\mathbf{x}-\mathbf{y}\| \leq \frac{\delta}{2\varrho\beta^2}\|\mathbf{x}-\mathbf{y}\| \leq \frac{\delta^2}{2\bar{\alpha}^{\frac{1}{2}}\beta}\,\|\mathbf{x}-\mathbf{y}\|.$$

If $n=2$, we have that $\delta=\sqrt{\rho(A)}$ and $\beta=\sqrt{\lambda_{\min}(A)}$. If $n\geq 3$, in which case a 2-dimensional reduced ellipsoid is considered, we have similarly to (7.16) that $\delta\leq\sqrt{\rho(A)}$ and $\beta\geq\sqrt{\lambda_{\min}(A)}$. Consequently, (7.22) is always true. □

With the help of Lemma 3, we can now estimate the distance between $d(\mathbf{a},\mathbf{x}_k)$ and $d(\mathbf{a},\mathbf{x}^*)$ by the angle θ_k in (7.2).

LEMMA 5. *Denote* $\mathcal{S}(\mathbf{a},\mathcal{E}) = \{\mathbf{x}\in\Omega(\mathcal{E}) : \theta(\mathbf{a}-\mathbf{x}, A\mathbf{x}+\mathbf{b}) \leq \frac{\pi}{2}\}$. *If* $\mathbf{x}_k\in\mathcal{S}(\mathbf{a},\mathcal{E})$, *there exists some positive constant* c_5 *such that*

$$d(\mathbf{a},\mathbf{x}_k) - d(\mathbf{a},\mathbf{x}^*) \leq c_5\sin^2\frac{\theta_k}{2}. \tag{7.27}$$

Proof. Define

$$\phi(\mathbf{x}) = \begin{cases}\theta(\mathbf{a}-\mathbf{x}^*,\mathbf{x}-\mathbf{x}^*) & \text{if } \mathbf{x}\neq\mathbf{x}^*,\\ \frac{\pi}{2} \quad \text{if } \mathbf{x}=\mathbf{x}^*.\end{cases}$$

It is easy to see that $\phi^* := \max\{\phi(\mathbf{x}) : \mathbf{x}\in\mathcal{S}(\mathbf{a},\mathcal{E})\} \geq \frac{\pi}{2}$ since $\mathbf{x}^*\subset\mathcal{S}(\mathbf{a},\mathcal{E})$. In addition, notice that the point $\bar{\mathbf{x}}$ in $\mathcal{E}$ satisfying $\theta(\mathbf{a}-\mathbf{x}^*,\bar{\mathbf{x}}-\mathbf{x}^*)=\pi$ does not belong to $\mathcal{S}(\mathbf{a},\mathcal{E})$. Then we have by the compactness of $\mathcal{S}(\mathbf{a},\mathcal{E})$ that $\phi^*<\pi$. Further, denote $\sigma_k=\theta(\mathbf{a}-\mathbf{x}^*,\,\mathbf{x}_k-\mathbf{x}^*)$ (see Figure 6). In a similar way, we can show that $\frac{\pi}{2}\geq\pi-(\theta_k+\sigma_k)\geq\xi^*$. It follows that for any $\mathbf{x}_k\in\mathcal{S}(\mathbf{a},\mathcal{E})$,

$$\sin\sigma_k\geq\sin\phi^* \quad\text{and}\quad \sin(\theta_k+\sigma_k)\geq\sin\xi^*. \tag{7.28}$$

From the triangle formed by $\mathbf{a}$, $\mathbf{x}^*$, and $\mathbf{x}_k$, we have that

$$\frac{\|\mathbf{x}^*-\mathbf{x}_k\|}{\sin\theta_k} = \frac{d(\mathbf{a},\mathbf{x}_k)}{\sin\sigma_k} = \frac{d(\mathbf{a},\mathbf{x}^*)}{\sin(\theta_k+\sigma_k)}. \tag{7.29}$$

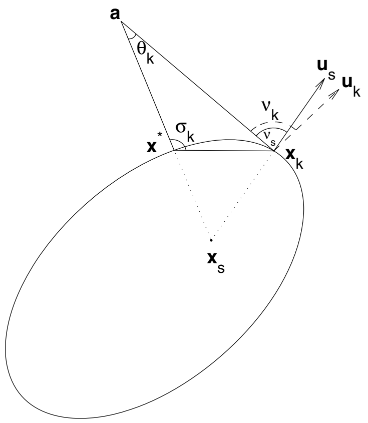


Fig. 6. *Diagram of proof of Lemmas* 5 *and* 6.

Noting that $\mathbf{a} - \mathbf{x}^*$ is parallel to $A\mathbf{x}^* + \mathbf{b}$, we have by Lemma 4 and the fact that $\mathbf{a} - \mathbf{x}^*$ is parallel to $A\mathbf{x}^* + \mathbf{b}$ that

$$-\cos\sigma_k = \cos(\pi - \sigma_k) \le c_4\|\mathbf{x}^* - \mathbf{x}_k\|. \tag{7.30}$$

Now, by (7.29), (7.30), and (7.28), we can obtain

$$\begin{aligned}
\frac{d(\mathbf{a},\mathbf{x}_k) - d(\mathbf{a},\mathbf{x}^*)}{d(\mathbf{a},\mathbf{x}^*)} &= \frac{\sin\sigma_k - \sin(\theta_k + \sigma_k)}{\sin(\theta_k + \sigma_k)} \\
&= \frac{\sin\sigma_k(1 - \cos\theta_k) - \cos\sigma_k \sin\theta_k}{\sin(\theta_k + \sigma_k)} \\
&\le \frac{2\sin\sigma_k \sin^2\frac{\theta_k}{2} + \frac{1}{2}c_4\|\mathbf{x}^* - \mathbf{x}_k\| \sin\theta_k}{\sin(\theta_k + \sigma_k)} \\
&= \frac{2\sin^2\sigma_k \sin^2\frac{\theta_k}{2} + \frac{1}{2}c_4\, d(a,\mathbf{x}_k)\sin^2\theta_k}{\sin(\theta_k + \sigma_k)\sin\sigma_k} \\
&= \frac{2\sin^2\sigma_k + 2c_4\, d(a,\mathbf{x}_k)\cos^2\frac{\theta_k}{2}}{\sin(\theta_k + \sigma_k)\sin\sigma_k} \sin^2\frac{\theta_k}{2} \\
&\le \frac{2(1 + c_4 d_{\max})}{\sin\xi^* \, \sin\phi^*} \sin^2\frac{\theta_k}{2}. \qquad (7.31)
\end{aligned}$$

In the above, $d_{\max} = \max_{\mathbf{x}\in\mathcal{E}} d(\mathbf{a},\mathbf{x}) < +\infty$. Therefore (7.27) holds with

$$c_5 = \frac{2(1 + c_4 d_{\max})\, d(a,\mathbf{x}^*)}{\sin\xi^* \, \sin\phi^*}, \tag{7.32}$$

which completes the proof. □

To establish the linear convergence of the algorithm, we now need a relation between the angle ν_k and θ_k. As in the proof of Theorem 3.8 in [9], we can show that $\nu_k \geq \theta_k$. In the following, we present a geometrical proof of this result (see also Figure 6 for the diagram of the proof).

LEMMA 6. *For any* $\mathbf{x}_k \in \mathcal{S}(\mathbf{a}, \mathcal{E})$ *with* $\mathbf{x}_k \neq \mathbf{x}^*$, *we have that* $\nu_k \geq \theta_k$.

Proof. Denote $\xi_k = \theta(\mathbf{x}^* - \mathbf{x}_k,\, \mathbf{u}_k)$. Since $\mathbf{u}_k$ is the normal direction of q at $\mathbf{x}_k$, we know that $\xi_k > \frac{\pi}{2}$. Denote by $\bar{\mathcal{S}}_k$ the 2-dimensional linear manifold including $\mathbf{a}$, $\mathbf{x}^*$, and $\mathbf{x}_k$. Note that the direction $\mathbf{u}_k$ does not necessarily lie in $\bar{\mathcal{S}}_k$. We then introduce a supplementary direction $\mathbf{u}_s \in \bar{\mathcal{S}}_k$ such that the angle $\theta(\mathbf{x}^* - \mathbf{x}_k,\, \mathbf{u}_s)$ has the same size as ξ_k. Meanwhile, we denote by $\bar{\mathcal{C}}_k$ the cone $\mathbf{x}_k \cup \{\mathbf{y} \neq \mathbf{x}_k : \theta(\mathbf{x}^* - \mathbf{x}_k,\, \mathbf{y} - \mathbf{x}_k) = \xi_k\}$. Then we can see that

$$\nu_s \doteq \theta(\mathbf{a} - \mathbf{x}_k,\, \mathbf{u}_s) = \min\{\theta(\mathbf{a} - \mathbf{x}_k,\, \mathbf{y} - \mathbf{x}_k) : \mathbf{y} \in \bar{\mathcal{C}}_k \backslash \{x_k\}\} \leq \nu_k. \tag{7.33}$$

On the other hand, since $\mathbf{x}^*$ is the projection of $\mathbf{a}$ on the ellipsoid, we have that $\theta(\mathbf{a} - \mathbf{x}^*, \mathbf{x}_k - \mathbf{x}^*) > \frac{\pi}{2}$. Consequently, the straight line passing $\mathbf{a}$ and $\mathbf{x}^*$ and the other one $\{\mathbf{x}_k + t\,\mathbf{u}_s : t \in \mathcal{R}^1\}$ must cross at some point, still say $\mathbf{x}_s$. From the triangle formed by $\mathbf{a}$, $\mathbf{x}_s$, and $\mathbf{x}_k$, we can get that

$$\nu_s \geq \theta_k. \tag{7.34}$$

Combining (7.33) and (7.34), we know the truth of this lemma. □

Now we are able to give the main theorem.

THEOREM 7. *Consider Algorithm* 4 *with* γ_k *given by* (6.2). *If* $c_k^{(1)}$ *and* $c_k^{(2)}$ *satisfy* (6.3) *and* (6.4), *there exists some positive constant* $c_6 < 1$ *such that*

$$\frac{d(\mathbf{a}, \mathbf{x}_{k+1}) - d(\mathbf{a}, \mathbf{x}^*)}{d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}^*)} \leq 1 - c_6. \tag{7.35}$$

Proof. By Lemmas 3, 5, and 6, we have that

$$\frac{d(\mathbf{a}, \mathbf{x}_{k+1}) - d(\mathbf{a}, \mathbf{x}^*)}{d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}^*)} = 1 - \frac{d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}_{k+1})}{d(\mathbf{a}, \mathbf{x}_k) - d(\mathbf{a}, \mathbf{x}^*)} \leq 1 - \frac{c_3 \tau}{c_5}. \tag{7.36}$$

Substituting the values of c_i, we know that (7.35) holds with

$$c_6 = \frac{c_3 \tau}{c_5} = \frac{\tau\, \sin \xi^*\, \sin \sigma^*}{\left[1 + (\rho(A)/\bar{\alpha})^{\frac{1}{2}}\, \kappa^{\frac{1}{2}}\, d(\mathbf{a}, \mathbf{x}^*)\right] \left[1 + (\rho(A)/\bar{\alpha})^{\frac{1}{2}}\, \kappa^{\frac{1}{2}}\, d_{\max}\right]}. \tag{7.37}$$

The proof is then complete. □

When $\mathbf{x}_k \to \mathbf{x}^*$, we have that $\sigma_k \to \frac{\pi}{2}$, $\theta_k + \sigma_k \to \frac{\pi}{2}$, and $d(\mathbf{a}, \mathbf{x}_k) \to d(\mathbf{a}, \mathbf{x}^*)$. Consequently, from the proof of Lemma 5, the linear convergence constant in (7.35) can be approximated by $1 - \bar{c}_6$, where

$$\bar{c}_6 = \frac{\tau}{\left[1 + (\rho(A)/\bar{\alpha})^{\frac{1}{2}}\, \kappa^{\frac{1}{2}}\, d(\mathbf{a}, \mathbf{x}^*)\right]^2}. \tag{7.38}$$

The relation (7.38) indicates that the convergence becomes slower when the condition number of A becomes larger or the point $\mathbf{a}$ to be projected is farther from the ellipsoid.

8. Numerical experiments. A 10-dimensional example has been used before to show the efficiency of new projection algorithms. Now we provide some numerical results for higher-dimensional problems. To facilitate our observation, we assume that the matrix A is diagonal and that its diagonal entries are given by $a_{ii} = 10^{\frac{i-1}{n-1} ncond}$ $(i = 1, \ldots, n)$ and *ncond* controls the condition number of the matrix A. The vector $\mathbf{b}$ is set to $\mathbf{0}$ in our tests, although the algorithms can apply to the case of nonzero $\mathbf{b}$. In case of nonzero $\mathbf{b}$, we need to find a feasible initial point in $\mathcal{E}$. We set $c = 0$ and $\alpha = 1$ so that the ellipsoid $\mathcal{E}$ lies in the unit ball at the origin. Equivalently, given n and *ncond*, the ellipsoid used in our test is

$$\mathcal{E} = \left\{ \mathbf{x} \in \mathcal{R}^n : \sum_{i=1}^{n} 10^{\frac{i-1}{n-1} ncond} x_i^2 = 1 \right\}.$$

For the choice of $\mathbf{a}$, we first generate a point $\tilde{\mathbf{a}} = (\tilde{a}_i)$ by

$$\tilde{a}_i = a_{ii}^{-\omega}, \tag{8.1}$$

where $\omega \geq 0$ is some parameter. Then we ask $\mathbf{x}^* = (\tilde{\mathbf{a}}^T A \tilde{\mathbf{a}})^{-\frac{1}{2}}\, \tilde{\mathbf{a}}$ to be the projection of $\mathbf{a}$ in the ellipsoid $\mathcal{E}$. It is easy to see that the larger ω is, the more $\mathbf{x}^*$ tends to an eigenvector of the matrix A corresponding to its small eigenvalue. For each $\mathbf{x}^*$, we choose different ϑ's for $\mathbf{a}$ such that $\|\mathbf{a}\| = \vartheta$. Given the size ϑ of $\mathbf{a}$, the point $\mathbf{a}$ can be calculated by

$$\mathbf{a} = \mathbf{x}^* + \frac{\vartheta^2 - \|\mathbf{x}^*\|^2}{1 + \sqrt{1 + \|A\mathbf{x}^*\|^2 \left[\vartheta^2 - \|\mathbf{x}^*\|^2\right]}} A\mathbf{x}^*.$$

The above strategy enables us not only to control both the size and the direction of $\mathbf{a}$ (by the parameters ω and ϑ) but also to know its exact projection $\mathbf{x}^*$. To sum up, the construction of our test problems depends on the four parameters n, *ncond*, ω, and ϑ. In our tests, we fix $n = 10^4$ and vary the other parameters:

$$ncond \in \{2,\ 3,\ 4,\ 5\}, \quad \omega \in \left\{0,\ \frac{1}{8},\ \frac{1}{4},\ \frac{1}{2}\right\}, \quad \vartheta \in \{2,\ 5,\ 10,\ 50\}.$$

We tested the Lin–Han algorithm and Algorithms 1–4 with the MATLAB language (version 6.5.0). For all cases, the initial point is set to $\mathbf{x}_0 = \vartheta^{-\frac{1}{2}}\mathbf{a}$. The stopping condition is (2.8) with $\varepsilon = 10^{-6}$. For the Lin–Han algorithm, we set $\gamma_k = (\rho(A))^{-1} = 10^{-ncond}$. As analyzed in section 3, this choice of γ_k favors the comparison of the Lin–Han algorithm since any underestimation of this value may deteriorate the performance of the algorithm. The parameters in Algorithm 4 are chosen as follows:

$$m_1 = 1, \quad m_2 = 1, \quad c_1 = 0.1, \quad c_2 = 0.8. \tag{8.2}$$

Nevertheless, good numerical results are obtained with other values of $(m_1,\ m_2,\ c_1,\ c_2)$, for example (1, 1, 0.05, 0.9). Generally, if $m_1 = m_2 = 1$ are fixed, the suggested arrangements for c_1 and c_2 are that

$$c_2 \in [0.7,\ 0.9], \quad c_1 \in [0,\ 0.95 - c_2].$$

The iteration numbers required by the algorithms for each case are noted in Table 2, where LH and Ai stand for the Lin–Han algorithm and Algorithm i, respectively.

Table 2
Numerical comparisons of five projection algorithms.

		$ncond = 2$							$ncond = 3$		
LH	A1	A2	A3	A4	ω	ϑ	LH	A1	A2	A3	A4
72	49	40	19	17	0	2	209	141	118	44	43
110	74	62	25	25	0	5	316	212	184	62	57
134	90	76	29	29	0	10	382	256	231	77	65
163	109	94	35	31	0	50	434	290	280	86	77
82	54	45	19	19	1/8	2	269	173	148	52	41
127	82	70	26	29	1/8	5	435	278	246	83	48
157	101	88	32	31	1/8	10	559	357	324	104	51
194	124	110	40	35	1/8	50	716	456	440	137	87
94	58	50	22	21	1/4	2	344	208	186	65	49
146	89	80	31	31	1/4	5	580	349	322	104	64
181	110	100	37	33	1/4	10	773	465	436	137	121
225	136	126	43	39	1/4	50	1076	647	630	193	97
121	63	64	37	27	1/2	2	579	293	304	136	62
187	96	100	52	35	1/2	5	994	501	534	223	115
228	117	124	62	37	1/2	10	1337	672	728	295	166
278	142	152	73	43	1/2	50	1899	953	1052	409	165
		$ncond = 4$							$ncond = 5$		
LH	A1	A2	A3	A4	ω	ϑ	LH	A1	A2	A3	A4
479	321	282	92	75	0	2	809	541	529	158	77
621	415	396	125	85	0	5	697	466	529	137	82
623	416	429	125	88	0	10	580	388	440	116	70
524	350	377	104	74	0	50	506	338	364	104	69
734	469	414	128	70	1/8	2	1732	1104	1018	290	141
1136	724	672	200	97	1/8	5	2255	1437	1466	386	141
1362	868	854	239	101	1/8	10	2069	1318	1546	362	126
1298	828	932	236	105	1/8	50	1299	828	1008	236	94
1068	643	586	188	76	1/4	2	3020	1815	1684	509	151
1834	1102	1036	311	120	1/4	5	5015	3012	2920	833	175
2485	1493	1452	425	173	1/4	10	6430	3862	3972	1058	247
3261	1959	2122	557	197	1/4	50	5404	3245	4288	899	203
2357	1183	1246	535	171	1/2	2	8700	4357	4622	1858	429
4352	2181	2354	937	253	1/2	5	16715	8367	9074	3415	546
6352	3182	3488	1312	379	1/2	10	25653	12839	14178	5071	775
10769	5391	6128	2119	360	1/2	50	49765	24902	29210	9349	800

Since the number of calculations per iteration required by each algorithm is similar, the algorithmic performance can basically be evaluated by the required iteration numbers. From Table 2, we make the following comments.

Regarding influence of ncond, ω, and ϑ. In general, we see that the problem becomes more difficult as *ncond* and ω increase. In other words, when the ellipsoid becomes more *flat*, it is more difficult to project those points close to the flat part of the ellipsoid. The influence of ϑ, namely the size of $\mathbf{a}$, is different. If *ncond* and ω are fixed, the increase of ϑ leads to more projection iterations in most cases. However, for quite many cases such as $ncond = 4$, $\omega \in \{0, 1/8\}$, and $ncond = 5$, $\omega \in \{0, 1/8, 1/4\}$, the iterations required for $\vartheta = 50$ are generally fewer than those for $\vartheta = 10$. It seems to us that for each case of *ncond* and ω, the problem becomes eventually more difficult as $\mathbf{a}$ gets farther away from the ellipsoid and then eventually easier after $\mathbf{a}$ exceeds some distance.

Regarding efficiency of the five projection algorithms. It is evident that the Lin–Han algorithm is the worst and Algorithm 4 is the best. Further, we see that the gain achieved by Algorithm 4 is bigger as the problem becomes more difficult. In the most difficult case, when $ncond = 5$, $\omega = 1/2$, and $\vartheta = 50$, the iteration numbers required by Algorithm 4 are only about one sixty-third of the number required by the Lin–Han algorithm. In other words, Algorithm 4 is less influenced by the difficulty of the problem. Algorithm 3 is the second best among the five algorithms. Comparing Algorithms 1 and 2, we see that in *easy* cases, Algorithm 2 is better than Algorithm 1, whereas Algorithm 1 requires fewer iterations than Algorithm 2 in *difficult* cases.

9. Discussion.

In this paper we have proposed several new algorithms for projection on a general ellipsoid by considering the 2-dimensional reduced ellipsoid at each iteration. To avoid the direct estimation of the spectral radius $\rho(A)$ in the Lin–Han algorithm, we provided the maximal 2-dimensional inside ball algorithm (Algorithm 1) and the sequential 2-dimensional projection algorithm (Algorithm 2). However, we found that the solution procedure of Algorithm 1 tends to some 2-dimensional reduced ellipsoid. For Algorithm 2, the iterations tend to two 2-dimensional reduced ellipsoids alternately. Therefore we investigated the hybrid use of the two algorithms and proposed the simple hybrid projection algorithm (Algorithm 3) and the general hybrid projection algorithm (Algorithm 4). Our numerical experiments show that Algorithms 1–4, even Algorithm 4, are much faster than the Lin–Han algorithm even when the spectral radius $\rho(A)$ is exactly known. To further improve Algorithm 4, we feel that one possible approach is to impose some conjugacy on the 2-dimensional reduced ellipsoids $\{\mathcal{E}_k\}$.

One disadvantage of the algorithms of this paper is that they require a feasible point of the ellipsoid $\mathcal{E}$. Assume that a feasible point, $\mathbf{x}_f$ say, has been found. Then one can choose the intersection of $\mathcal{E}$ and the line segment $\mathcal{L}(\mathbf{a}, \mathbf{x}_f)$ as an initial point. If the right-hand-side term $\mathbf{b} = \mathbf{0}$, a feasible point can be easily found in $\mathcal{E}$. However, this is not always the case with nonzero $\mathbf{b}$. In that case, to find a feasible point of $\mathcal{E}$ and use the algorithms, one might need to reduce the function $q(\mathbf{x})$ from some infeasible point with the help of some minimization algorithm. Therefore it may be interesting to find some infeasible projection algorithms in which a feasible point is not necessary.

It is obvious that the maximal 2-dimensional inside ball algorithm can be extended to the problem of calculating the distance between two ellipsoids considered in Lin and Han [10],

$$\begin{array}{ll} \min & \|\mathbf{x} - \mathbf{y}\| \\ \text{s.t.} & \mathbf{x} \in \mathcal{E},\ \mathbf{y} \in \bar{\mathcal{E}}. \end{array}$$

Hence the similar disadvantage of estimating the spectral radius of some ellipsoid in their algorithm can be avoided. Some kind of extension of the sequential 2-dimensional ellipsoid projection algorithm to such a problem is also possible. For example, assuming that $\mathbf{x}_k \in \mathcal{E}$ and $\mathbf{y}_k \in \bar{\mathcal{E}}$ have been obtained at the kth iteration, we can construct a maximal 2-dimensional inside ball of $\mathcal{E}$ at $\mathbf{x}_k$. Define the center of this ball to be c_k. Then we can take $\mathbf{y}_{k+1} \in \bar{\mathcal{E}}$ to be the projection of $\mathbf{c}_k$ on $\bar{\mathcal{E}}$ and then let $\mathbf{x}_{k+1} = \Omega(\mathcal{E}) \cap \mathcal{L}(\mathbf{c}_k, \mathbf{y}_{k+1})$. Therefore faster algorithms for the above problem should also be able to be obtained. Nevertheless, it still remains to study how to design the most efficient algorithms. In addition, it may be also interesting to investigate how to extend the idea of the algorithms proposed in this paper to solve the projection problem on a general convex set. More recent work on this aspect can be seen in Lin [8] and Lin and Han [11].

Acknowledgments. The author is very grateful to Professor Roger Fletcher for many useful discussions with him and for his kindness while the author was visiting the University of Dundee. Many thanks are also due to the two anonymous referees, whose comments and suggestions improved the quality of this paper greatly.

REFERENCES

[1] E. G. Birgin, J. M. Martínez, and M. Raydan, *Nonmonotone spectral projected gradient methods on convex sets*, SIAM J. Optim., 10 (2000), pp. 1196–1211.

[2] Y. H. Dai and R. Fletcher, *New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds*, Math. Program., 103 (2005), pp. 541–559.

[3] Y. H. Dai and Y. Yuan, *Alternate minimization gradient method*, IMA J. Numer. Anal., 23 (2003), pp. 377–393.

[4] R. L. Dykstra, *An algorithm for restricted least-squares regression*, J. Amer. Statist. Assoc., 78 (1983), pp. 837–842.

[5] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., Wiley, New York, 1991.

[6] S. J. Grotzinger and C. Witzgall, *Projections onto order simplexes*, Appl. Math. Optim., 12 (1984), pp. 247–270.

[7] S.-P. Han and G. Lou, *A parallel algorithm for a class of convex programs*, SIAM J. Control Optim., 26 (1988), pp. 345–355.

[8] A. Lin, *A class of methods for projection on a convex set*, Adv. Modeling Optim., 5 (2003), pp. 211–221.

[9] A. Lin and S.-P. Han, *Projection on an Ellipsoid*, Research report, Department of Mathematical Sciences, The Johns Hopkins University, Baltimore, MD, 2001.

[10] A. Lin and S.-P. Han, *On the distance between two ellipsoids*, SIAM J. Optim., 13 (2002), pp. 298–308.

[11] A. Lin and S.-P. Han, *A class of methods for projection on the intersection of several ellipsoids*, SIAM J. Optim., 15 (2004), pp. 129–138.

[12] P. M. Pardalos and N. Kovoor, *An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds*, Math. Program., 46 (1990), pp. 321–328.

SIAM J. OPTIM.
Vol. 16, No. 4, pp. 1007–1023

A PROXIMAL BUNDLE METHOD WITH APPROXIMATE SUBGRADIENT LINEARIZATIONS*

KRZYSZTOF C. KIWIEL†

Abstract. We give a proximal bundle method for minimizing a convex function f over a closed convex set. It only requires evaluating f and its subgradients with an accuracy $\epsilon > 0$, which is fixed but possibly unknown. It asymptotically finds points that are ϵ-optimal. When applied to Lagrangian relaxation, it allows for ϵ-accurate solutions of Lagrangian subproblems and finds ϵ-optimal solutions of convex programs.

Key words. nondifferentiable optimization, convex programming, proximal bundle methods, approximate subgradients, Lagrangian relaxation

AMS subject classifications. 65K05, 90C25

DOI. 10.1137/040603929

1. Introduction.

We consider the convex constrained minimization problem

$$f_* := \inf\{\, f(x) : x \in S \,\}, \tag{1.1}$$

where S is a nonempty closed convex set in the Euclidean space $\mathbb{R}^n$ with inner product $\langle \cdot, \cdot \rangle$ and norm $|\cdot|$, and $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function. We assume that for fixed *accuracy tolerances* $\epsilon_f \geq 0$ and $\epsilon_g \geq 0$, for each $y \in S$ we can find an *approximate value* f_y and an *approximate subgradient* g_y of f that produce the *approximate linearization* of f:

$$\bar{f}_y(\cdot) := f_y + \langle g_y, \cdot - y \rangle \leq f(\cdot) + \epsilon_g \quad \text{with} \quad \bar{f}_y(y) = f_y \geq f(y) - \epsilon_f. \tag{1.2}$$

Thus $f_y \in [f(y) - \epsilon_f, f(y) + \epsilon_g]$ estimates $f(y)$, while $g_y \in \partial_\epsilon f(y)$ for the *total accuracy tolerance* $\epsilon := \epsilon_f + \epsilon_g$; i.e., g_y is a member of the ϵ-subdifferential of f at y,

$$\partial_\epsilon f(y) := \{\, g : f(\cdot) \geq f(y) - \epsilon + \langle g, \cdot - y \rangle \,\}.$$

The above assumption is realistic in many applications. For instance, if f is a max-type function of the form

$$f(y) := \sup\{\, F_z(y) : z \in Z \,\}, \tag{1.3}$$

where each $F_z : \mathbb{R}^n \to \mathbb{R}$ is convex and Z is an infinite set, then it may be impossible to calculate $f(y)$. However, we may still consider the following two cases. In the first case of *controllable accuracy*, for each positive $\tilde{\epsilon}$ one can find an $\tilde{\epsilon}$-maximizer of (1.3), i.e., an element $z_y \in Z$ satisfying $F_{z_y}(y) \geq f(y) - \tilde{\epsilon}$; in the second case, this may be possible only for some fixed (and possibly unknown) $\tilde{\epsilon} < \infty$. In both cases we may set $f_y := F_{z_y}(y)$ and take g_y as any subgradient of F_{z_y} at y to satisfy (1.2) with $\epsilon_f := \tilde{\epsilon}$, $\epsilon_g := 0$; then $\epsilon = \tilde{\epsilon}$.

*Received by the editors February 6, 2004; accepted for publication (in revised form) July 1, 2004; published electronically January 26, 2006. This research was supported by the State Committee for Scientific Research under grant 4T11A00622.

http://www.siam.org/journals/siopt/16-4/60392.html

†Systems Research Institute, Polish Academy of Sciences, Newelska 6, 01–447 Warsaw, Poland (kiwiel@ibspan.waw.pl).

A special case of (1.3) arises in *Lagrangian relaxation* [Ber99, section 5.5.3], [HUL93, Chap. XII], where problem (1.1) with $S := \mathbb{R}^n_+$ is the Lagrangian dual of the primal problem

$$\text{(1.4)} \qquad \sup\ \psi_0(z) \quad \text{s.t.} \quad \psi_j(z) \geq 0,\ j = 1\colon n,\ z \in Z,$$

with $F_z(y) := \psi_0(z) + \langle y, \psi(z)\rangle$ for $\psi := (\psi_1, \dots, \psi_n)$. Then, for each multiplier $y \geq 0$, we need only find $z_y \in Z$ such that $f_y := F_{z_y}(y) \geq f(y) - \epsilon$ in (1.3) to use $g_y := \psi(z_y)$. For instance, if (1.4) is a semidefinite program with each ψ_j affine and Z the set of symmetric positive semidefinite matrices of order m with unit trace, then $f(y)$ is the maximum eigenvalue of a symmetric matrix $M(y)$ depending affinely on y [Tod01, section 6.3], and z_y can be found by computing an approximate eigenvector corresponding to the maximum eigenvalue of $M(y)$ via the Lanczos method [HeK02], [HeR00].

This paper extends the proximal bundle method of [Kiw90] and its variants [Hin01], [ScZ92], [HUL93, section XV.3] to the inexact setting of (1.2) with *unknown* ϵ_f and ϵ_g. Our extension is natural and simple: the original method is run as if the linearizations were exact until a *predicted descent test* discovers their inaccuracy; then the method is restarted with a decreased proximity weight. Since our descent test (or a similar one) is employed as a stopping criterion by the existing implementations of proximal bundle methods, our analysis also sheds light on the implementations' behavior in the inexact case (cf. section 4.5).

We show that our method asymptotically estimates the optimal value f_* of (1.1) with accuracy ϵ and finds ϵ-optimal points. In Lagrangian relaxation, under standard convexity and compactness assumptions on problem (1.4) (see section 5), it finds ϵ-optimal primal solutions by combining partial Lagrangian solutions, even when Lagrange multipliers don't exist. This seems to be the first such result on primal recovery in Lagrangian relaxation.

We now comment briefly on other relations with the literature.

The setting of (1.2) subsumes those in [Hin01], [Kiw85], [Kiw95a]. Indeed, suppose that for some nonnegative tolerances $\tilde{\epsilon}_f^-$, $\tilde{\epsilon}_f^+$, and $\tilde{\epsilon}_g$, for each $y \in S$ we can find some

$$\text{(1.5)} \qquad f_y \in \left[f(y) - \tilde{\epsilon}_f^-, f(y) + \tilde{\epsilon}_f^+ \right] \quad \text{and} \quad g_y \in \partial_{\tilde{\epsilon}_g} f(y).$$

Then (1.2) holds with $\epsilon_f := \tilde{\epsilon}_f^-$ and $\epsilon_g := \tilde{\epsilon}_f^+ + \tilde{\epsilon}_g$. We add that $\tilde{\epsilon}_f^- = \tilde{\epsilon}_f^+ = \tilde{\epsilon}_g$ in [Kiw85], [Hin01] uses $\tilde{\epsilon}_f^- = \tilde{\epsilon}_f^+ = 0$, i.e., exact values $f_y = f(y)$, whereas [Kiw95a] employs (1.2) with $\epsilon_g = 0$ (corresponding to $\tilde{\epsilon}_f^- := \tilde{\epsilon}_g := \epsilon_f = \epsilon$ and $\tilde{\epsilon}_f^+ := 0$ in (1.5)).

First, our method is more widely applicable than those in [Hin01], [Kiw85], [Kiw95a], since [Kiw85], [Kiw95a] assume that the $\tilde{\epsilon}$-tolerances in (1.5) are controllable and can be driven to 0, whereas [Hin01] needs exact f-values. Thus only our method can handle Lagrangian relaxation with subproblem solutions of unknown accuracy. Second, our convergence results are stronger than those in [Hin01], since they handle constraints and practicable stopping criteria (cf. section 4.2). Third, our method is much simpler than that of [Hin01].

Finally, the method of [Sol03] works in the setting of (1.2) with $\epsilon_g = 0$ and known (possibly varying) tolerances ϵ_f employed in its stopping criterion and the descent test. If the tolerances are below a fraction of a stopping threshold $\Delta > 0$, the method terminates, ensuring that the traditional stopping criterion of bundle methods is met for this Δ. In turn, the framework of [Mil01, section 4.5] is related to those in [Kiw85], [Kiw95a].

The paper is organized as follows. In section 2 we present our proximal bundle method. Its convergence is analyzed in section 3. Several modifications are given in section 4. Applications to Lagrangian relaxation of convex and nonconvex programs are studied in section 5.

2. The inexact proximal bundle method.

We may regard (1.1) as an unconstrained problem $f_* = \min f_S$ with the *essential objective*

$$f_S := f + i_S, \tag{2.1}$$

where i_S is the *indicator* function of S ($i_S(x) = 0$ if $x \in S$, ∞ if $x \notin S$).

Our method generates a sequence of *trial points* $\{y^k\}_{k=1}^{\infty} \subset S$ for evaluating the approximate values $f_y^k := f_{y^k}$, subgradients $g^k := g_{y^k}$, and linearizations $f_k := \bar{f}_{y^k}$ such that

$$f_k(\cdot) = f_y^k + \langle g^k, \cdot - y^k\rangle \le f(\cdot) + \epsilon_g \quad \text{with} \quad f_k(y^k) = f_y^k \ge f(y^k) - \epsilon_f, \tag{2.2}$$

as stipulated in (1.2). Iteration k uses the polyhedral *cutting-plane model* of f

$$\check{f}_k(\cdot) := \max_{j \in J^k} f_j(\cdot) \quad \text{with} \quad k \in J^k \subset \{1, \ldots, k\} \tag{2.3}$$

for finding

$$y^{k+1} := \arg\min \left\{ \phi_k(\cdot) := \check{f}_k(\cdot) + i_S(\cdot) + \tfrac{1}{2t_k}|\cdot - x^k|^2 \right\}, \tag{2.4}$$

where $t_k > 0$ is a *stepsize* that controls the size of $|y^{k+1} - x^k|$ and the *prox center* $x^k := y^{k(l)}$ has the value $f_x^k := f_y^{k(l)}$ for some $k(l) \le k$ (usually $f_x^k = \min_{j=1}^k f_y^j$). Note that, by (2.2),

$$f(x^k) - \epsilon_f \le f_x^k \le f(x^k) + \epsilon_g. \tag{2.5}$$

However, we may have $f_x^k < \check{f}_k(x^k) = \phi_k(x^k)$ in (2.4), in which case the *predicted descent*

$$v_k := f_x^k - \check{f}_k(y^{k+1}) \tag{2.6}$$

may be nonpositive; then t_k is increased and y^{k+1} is recomputed to decrease $\check{f}_k(y^{k+1})$ until $v_k > 0$ (specific tests on v_k for increasing t_k are discussed below and in section 4.3). A *descent* step to $x^{k+1} := y^{k+1}$ with $f_x^{k+1} := f_y^{k+1}$ occurs if $f_y^{k+1} \le f_x^k - \kappa v_k$ for a fixed $\kappa \in (0,1)$. Otherwise, a *null* step $x^{k+1} := x^k$ improves the next model $\check{f}_{k+1}$ with f_{k+1} (cf. (2.3)).

For choosing J^{k+1}, note that by the optimality condition $0 \in \partial\phi_k(y^{k+1})$ for (2.4),

$$\exists p_f^k \in \partial\check{f}_k(y^{k+1}) \text{ such that } p_S^k := -(y^{k+1} - x^k)/t_k - p_f^k \in \partial i_S(y^{k+1}) \tag{2.7}$$

and there are multipliers ν_j^k, $j \in J^k$, also known as *convex weights,* such that

$$p_f^k = \sum_{j \in J^k} \nu_j^k g^j, \ \sum_{j \in J^k} \nu_j^k = 1, \ \nu_j^k \ge 0, \ \nu_j^k \left[\check{f}_k(y^{k+1}) - f_j(y^{k+1})\right] = 0, \ j \in J^k. \tag{2.8}$$

Let $\hat{J}^k := \{j \in J^k : \nu_j^k \ne 0\}$. To save storage without impairing convergence, it suffices to choose $J^{k+1} \supset \hat{J}^k \cup \{k+1\}$ (i.e., we may drop inactive linearizations f_j with $\nu_j^k = 0$ that do not contribute to the trial point y^{k+1}).

The subgradient relations in (2.7) enable us to derive an optimality estimate from the following *aggregate linearizations* of $\check{f}_k$ and f, i_S, $\check{f}_S^k := \check{f}_k + i_S$ and f_S, respectively:

$$\bar{f}_k(\cdot) := \check{f}_k(y^{k+1}) + \langle p_f^k, \cdot - y^{k+1}\rangle \le \check{f}_k(\cdot) \le f(\cdot) + \epsilon_g, \tag{2.9}$$

$$\bar{\imath}_S^k(\cdot) := \langle p_S^k, \cdot - y^{k+1}\rangle \le i_S(\cdot), \tag{2.10}$$

$$\bar{f}_S^k(\cdot) := \bar{f}_k(\cdot) + \bar{\imath}_S^k(\cdot) \le \check{f}_S^k(\cdot) := \check{f}_k(\cdot) + i_S(\cdot) \le f_S(\cdot) + \epsilon_g, \tag{2.11}$$

where the final inequalities follow from (2.1)–(2.3). Adding (2.9)–(2.10) and using (2.11) and the linearity of

$$\bar{f}_S^k(\cdot) = \check{f}_k(y^{k+1}) + \langle p_f^k + p_S^k, \cdot - y^{k+1}\rangle, \tag{2.12}$$

we get

$$f_x^k + \langle p^k, \cdot - x^k\rangle - \alpha_k = \bar{f}_S^k(\cdot) \le \check{f}_S^k(\cdot) \le f_S(\cdot) + \epsilon_g, \tag{2.13}$$

where

$$p^k := p_f^k + p_S^k = (x^k - y^{k+1})/t_k \quad \text{and} \quad \alpha_k := f_x^k - \bar{f}_S^k(x^k) \tag{2.14}$$

are the *aggregate subgradient* (cf. (2.7)) and the *aggregate linearization error*, respectively. The aggregate subgradient inequality (2.13) yields the *optimality estimate*

$$f_x^k \le f(x) + \epsilon_g + |p^k||x - x^k| + \alpha_k \quad \text{for all } x \in S. \tag{2.15}$$

Combined with $f(x^k) - \epsilon_f \le f_x^k$ (cf. (2.5)), the optimality estimate (2.15) says that the point x^k is *ϵ-optimal* (i.e., $f(x^k) - f_* \le \epsilon := \epsilon_f + \epsilon_g$) if the *optimality measure*

$$V_k := \max\left\{\, |p^k|, \alpha_k \,\right\} \tag{2.16}$$

is zero; x^k is approximately ϵ-optimal if V_k is small.

Thus we would like V_k to vanish asymptotically. Hence it is crucial to bound V_k via the predicted descent v_k, since normally bundling and descent steps drive v_k to 0. To this end, we first highlight some elementary properties of α_k and v_k; see Figure 2.1.

In other words, (2.13) and (2.5) mean that the model $\check{f}_S^k$ and its linearization $\bar{f}_S^k$ may overshoot the objective f_S by at most ϵ_g, whereas f_x^k may underestimate $f(x^k)$ by at most ϵ_f. Hence the linearization error α_k of (2.14) may drop below 0 by no more than $\epsilon := \epsilon_f + \epsilon_g$:

$$\alpha_k \ge f_x^k - \check{f}_S^k(x^k) \ge f_x^k - f(x^k) - \epsilon_g \ge -\epsilon_f - \epsilon_g = -\epsilon. \tag{2.17}$$

The predicted descent v_k (cf. (2.6)) may be expressed in terms of p^k and α_k as

$$v_k = t_k|p^k|^2 + \alpha_k = |d^k|^2/t_k + \alpha_k \quad \text{with} \quad d^k := y^{k+1} - x^k = -t_k p^k \tag{2.18}$$

being the *search direction*. Indeed, $|y^{k+1} - x^k|^2/t_k = t_k|p^k|^2$ by (2.14), whereas by (2.12)

$$\check{f}_k(y^{k+1}) = \bar{f}_S^k(y^{k+1}) = \bar{f}_S^k(x^k) + \langle p^k, y^{k+1} - x^k\rangle = \bar{f}_S^k(x^k) - |y^{k+1} - x^k|^2/t_k,$$

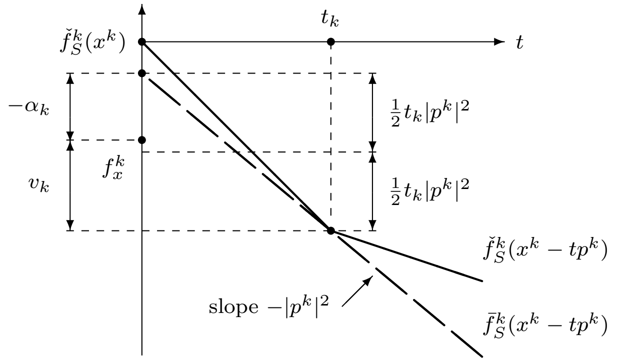


FIG. 2.1. *Predicted descent domination:* $v_k \geq -\alpha_k \Leftrightarrow \frac{1}{2}t_k|p^k|^2 \geq -\alpha_k \Leftrightarrow v_k \geq \frac{1}{2}t_k|p^k|^2$.

so $v_k := f_x^k - \check f_k(y^{k+1}) = \alpha_k + t_k|p^k|^2$ by (2.14). Note that $v_k \geq \alpha_k$.

Since $V_k := \max\{|p^k|, \alpha_k\}$, $v_k = t_k|p^k|^2 + \alpha_k$, and $-\alpha_k \leq \epsilon$ (cf. (2.16)–(2.18)), we have

$$V_k = \max\left\{ [(v_k - \alpha_k)/t_k]^{1/2}, \alpha_k \right\}, \tag{2.19}$$

$$V_k \leq \max\left\{ (2v_k/t_k)^{1/2}, v_k \right\} \qquad \text{if} \quad v_k \geq -\alpha_k, \tag{2.20}$$

$$V_k < (-2\alpha_k/t_k)^{1/2} \leq (2\epsilon/t_k)^{1/2} \quad \text{if} \quad v_k < -\alpha_k. \tag{2.21}$$

The bound (2.21) will imply that if x^k isn't ϵ-optimal (so that V_k can't vanish as t_k increases), then $v_k \geq -\alpha_k$ and the bound (2.20) hold for t_k large enough; on the other hand, the bound (2.20) suggests that t_k shouldn't decrease unless V_k is small enough.

We now have the necessary ingredients to state our method in detail.

ALGORITHM 2.1.

Step 0 (initialization). Select $x^1 \in S$, a *descent parameter* $\kappa \in (0,1)$, a *stepsize bound* $T_1 > 0$, and a stepsize $t_1 \in (0, T_1]$. Set $y^1 := x^1$, $f_x^1 := f_y^1$ (cf. (2.2)), $g^1 := g_{y^1}$, $J^1 := \{1\}$, $i_t^1 := 0$, $k := k(0) := 1$, $l := 0$ ($k(l) - 1$ will denote the iteration of the lth descent step).

Step 1 (trial point finding). Find y^{k+1} and multipliers ν_j^k such that (2.7)–(2.8) hold.

Step 2 (stopping criterion). If $V_k = 0$ (cf. (2.15)–(2.16)), stop ($f_x^k \leq f_* + \epsilon_g$).

Step 3 (stepsize correction). If $v_k < -\alpha_k$, set $t_k := 10t_k$, $T_k := \max\{T_k, t_k\}$, $i_t^k := k$ and loop back to Step 1; else set $T_{k+1} := T_k$.

Step 4 (descent test). Evaluate f_y^{k+1} and g^{k+1} (cf. (2.2)). If the *descent test* holds:

$$f_y^{k+1} \leq f_x^k - \kappa v_k, \tag{2.22}$$

set $x^{k+1} := y^{k+1}$, $f_x^{k+1} := f_y^{k+1}$, $i_t^{k+1} := 0$, $k(l+1) := k+1$ and increase l by 1 (*descent step*); else set $x^{k+1} := x^k$, $f_x^{k+1} := f_x^k$, and $i_t^{k+1} := i_t^k$ (*null step*).

Step 5 (bundle selection). Choose $J^{k+1} \supset \hat J^k \cup \{k+1\}$, where $\hat J^k := \{j \in J^k : \nu_j^k \neq 0\}$.

Step 6 (stepsize updating). If $k(l) = k + 1$ (i.e., after a descent step), select $t_{k+1} \in [t_k, T_{k+1}]$; otherwise, either set $t_{k+1} := t_k$, or choose $t_{k+1} \in [0.1t_k, t_k]$ if $i_t^{k+1} = 0$ and

$$f_x^k - f_{k+1}(x^k) \geq V_k := \max\left\{ |p^k|, \alpha_k \right\}. \tag{2.23}$$

Step 7 (loop). Increase k by 1 and go to Step 1.

A few comments on the method are in order.

Remark 2.2.

(i) When the feasible set S is polyhedral, Step 1 may use the quadratic programming (QP) method of [Kiw94], which can efficiently solve sequences of related subproblems (2.4).

(ii) Step 2 may also use the test $f_x^k \leq \inf \check{f}_S^k$ (cf. Lemma 2.3(i)); more practicable stopping criteria are discussed in section 4.2.

(iii) In the case of exact evaluations ($\epsilon = 0$), we have $v_k \geq \alpha_k \geq 0$ (cf. (2.17)–(2.18)), Step 3 is redundant, and Algorithm 2.1 becomes essentially that of [Kiw90].

(iv) To see the need for increasing t_k at Step 3, suppose $n = 1$, $f(x) = -x$, $S = \mathbb{R}$, $x^1 = 0$, $t_1 = \epsilon = 1$, $f_x^1 = g^1 = -1$, $f_2(x) = -x$. If Step 3 were omitted and null steps were taken when $v_k \leq 0$, the method would jam with $y^{k+1} = 1$ for $k \geq 1$. Also note that decreasing t_k would not help. In fact decreasing t_k at Step 6 aims at collecting more local information about f at null steps, whereas in such cases t_k must be increased to produce descent or confirm that x^k is ϵ-optimal (let $f(x) = \max\{-x, x-2\}$ above). Hence whenever t_k is increased at Step 3, the *stepsize indicator* $i_t^k \neq 0$ prevents Step 6 from decreasing t_k after null steps until the next descent step occurs (cf. Step 4).

(v) At Step 5, one may let $J^{k+1} := J^k \cup \{k+1\}$ and then, if necessary, drop from J^{k+1} an index $j \in J^k \setminus \hat{J}^k$ with the smallest $f_j(x^k)$ to keep $|J^{k+1}| \leq M$ for some $M \geq n + 2$.

(vi) Step 6 may use the procedure of [Kiw90, section 2] for updating the proximity weight $u_k := 1/t_k$, with obvious modifications.

We now show that the loop between Steps 1 and 3 is infinite iff $f_x^k \leq \inf \check{f}_S^k < \check{f}_k(x^k)$, in which case the current iterate x^k is already ϵ-optimal.

Lemma 2.3.

(i) *If* $f_x^k \leq \inf \check{f}_S^k$, *then* $f(x^k) - \epsilon_f \leq f_x^k \leq f_* + \epsilon_g$ *and* $f(x^k) \leq f_* + \epsilon$.

(ii) *Step 2 terminates, i.e.,* $V_k := \max\{|p^k|, \alpha_k\} = 0$, *iff* $f_x^k \leq \min \check{f}_S^k = \check{f}_S^k(x^k)$.

(iii) *If the loop between Steps* 1 *and* 3 *is infinite, then* $f_x^k \leq \inf \check{f}_S^k$ ($< \check{f}_S^k(x^k)$; *cf.* (ii)). *Moreover, in this case we have* $\check{f}_S^k(y^{k+1}) \downarrow \inf \check{f}_S^k$ *as* $t_k \uparrow \infty$.

(iv) *If* $f_x^k \leq \inf \check{f}_S^k$ *at Step* 1 *and Step* 2 *does not terminate* (*i.e.,* $\inf \check{f}_S^k < \check{f}_S^k(x^k)$; *cf.* (ii)), *then an infinite loop between Steps* 3 *and* 1 *occurs.*

Proof. (i) Combine $f_* = \inf f_S$ (cf. (1.1), (2.1)) with $\inf \check{f}_S^k \leq \inf f_S + \epsilon_g$ (cf. (2.13)) and $f(x^k) - \epsilon_f \leq f_x^k$ (cf. (2.5)), and use $\epsilon := \epsilon_f + \epsilon_g$ for the second inequality.

(ii) "⇒": Since $|p^k| = 0 \geq \alpha_k$, (2.13)–(2.14) yield $\bar{f}_S^k(x^k) \leq \check{f}_S^k(\cdot)$, $y^{k+1} = x^k$, and $f_x^k \leq \bar{f}_S^k(x^k)$, whereas by (2.12), $\bar{f}_S^k(x^k) = \check{f}_k(y^{k+1}) = \check{f}_S^k(x^k)$. "⇐": Since $\check{f}_S^k(x^k) = \min \check{f}_S^k$, using $\phi_k(x^k) = \min \check{f}_S^k \leq \phi_k(y^{k+1}) \leq \phi_k(x^k)$ in (2.4) gives $y^{k+1} = x^k$, so again $\bar{f}_S^k(x^k) = \check{f}_S^k(x^k)$ by (2.12), and (2.14) yields $p^k = 0$ and $\alpha_k = f_x^k - \check{f}_S^k(x^k) \leq 0$.

(iii) At Step 3 during the loop the facts $V_k < (2\epsilon/t_k)^{1/2}$ (cf. (2.21)) and $t_k \uparrow \infty$ give $\max\{|p^k|, \alpha_k\} =: V_k \to 0$, so (2.13) yields $f_x^k \leq \inf \check{f}_S^k$. The fact that $\check{f}_S^k(y^{k+1}) \downarrow \inf \check{f}_S^k$ as $t_k \uparrow \infty$ in (2.4) is well known; see, e.g., [Kiw95b, Lem. 2.1].

(iv) By (2.11), $\check{f}_k(y^{k+1}) = \check{f}_S^k(y^{k+1}) \geq \inf \check{f}_S^k$. Thus (cf. (2.6)) $v_k \leq f_x^k - \inf \check{f}_S^k \leq 0$ and (cf. (2.18)) $v_k = t_k|p^k|^2 + \alpha_k$ yield $\alpha_k \leq -t_k|p^k|^2$ at Step 3 with $p^k \neq 0$ (since

$\max\{|p^k|, \alpha_k\} =: V_k > 0$ at Step 2). Hence $\alpha_k < -\frac{t_k}{2}|p^k|^2$, so (cf. (2.18)) $v_k < -\alpha_k$ and Step 3 loops back to Step 1, after which Step 2 can't terminate due to (ii). □

Remark 2.4. By Lemma 2.3, the algorithm may terminate if $f_x^k \le \inf \check f_S^k$. When S is polyhedral, then either $\inf \check f_S^k = -\infty$, or there is $\check t_k$ such that $\check f_S^k(y^{k+1}) = \min \check f_S^k$ whenever $t_k \ge \check t_k$; either case may be discovered by a parametric QP method [Kiw95b], and the algorithm may stop if $f_x^k \le \min \check f_S^k$, thus forestalling an infinite loop in Steps 1 through 3.

3. Convergence. In view of Lemma 2.3, we may suppose that the algorithm neither terminates nor loops infinitely between Steps 1 and 3 (otherwise x^k is ϵ-optimal). At Step 4, $y^{k+1} \in S$ and $v_k > 0$ (by (2.20), since $V_k > 0$ at Step 2), so $x^{k+1} \in S$ and $f_x^{k+1} \le f_x^k$ for all k.

Let $f_x^\infty := \lim_k f_x^k$. We shall show that $f_x^\infty \le f_* + \epsilon_g$. Because the proof is quite complex, it is broken into a series of lemmas, starting with the following two simple results. To handle loops, let V_k' denote the minimum value of V_k at each iteration k.

Lemma 3.1. *If* $\underline{\lim}_k V_k' = 0$ *(e.g.,* $\underline{\lim}_k V_k = 0$*) and* $\{x^k\}$ *is bounded, then* $f_x^\infty \le f_* + \epsilon_g$.

Proof. Pick $K \subset \{1, 2, \dots\}$ such that $V_k' \xrightarrow{K} 0$. Fix $x \in S$. Letting $k \in K$ tend to infinity in (2.15)–(2.16) with $V_k = V_k'$ yields $f_x^\infty \le f(x) + \epsilon_g$, so $f_x^\infty \le \inf_S f + \epsilon_g = f_* + \epsilon_g$. □

Lemma 3.2. *Let* $T_\infty := \lim_k T_k$ *at Step* 4. *If* $T_\infty = \infty$, *then* $\underline{\lim}_k V_k' = 0$.

Proof. Let $K \subset \{1, 2, \dots\}$ index iterations k that increase T_k at Step 3. For $k \in K$, at Step 3 on the last loop to Step 1 we have $V_k < (2\epsilon/t_k)^{1/2}$ (cf. (2.21)) with t_k such that $10t_k$ becomes the final T_k, so the facts $0 \le V_k' \le V_k$ and $T_k \xrightarrow{K} \infty$ give $V_k' \xrightarrow{K} 0$. □

In view of Lemmas 3.1–3.2, we may assume that $T_\infty < \infty$ when $\{x^k\}$ is bounded, e.g., only finitely many descent steps occur. This case is analyzed below.

Lemma 3.3. *Suppose there exists* $\bar k$ *such that for all* $k \ge \bar k$, *Step* 3 *doesn't increase* t_k *and only null steps occur with* $t_{k+1} \le t_k$ *determined by Step* 6. *Then* $v_k \to 0$.

Proof. Fix $k \ge \bar k$. We first show that $\check f_S^{k+1} \ge \bar f_S^k$. Let $\hat f_k := \max_{j \in \hat J^k} f_j$. Since $\hat J^k := \{j \in J^k : \nu_j^k \ne 0\}$ and $g^j = \nabla f_j$, $\hat f_k \le \max_{j \in J^k} f_j =: \check f_k$ and (2.8) yield $\hat f_k(y^{k+1}) = \check f_k(y^{k+1})$ and $p_f^k \in \partial \hat f_k(y^{k+1})$. Thus $\bar f_k \le \hat f_k$ by (2.9), so $\hat f_k \le \check f_{k+1}$ ($\hat J^k \subset J^{k+1}$) gives $\bar f_k \le \check f_{k+1}$. Hence (2.10)–(2.11) yield $\bar f_S^k := \bar f_k + \bar\imath_S^k \le \check f_{k+1} + i_S =: \check f_S^{k+1}$.

Next, consider the following partial linearization of the objective ϕ_k of (2.4):

$$\bar\phi_k(\cdot) := \bar f_S^k(\cdot) + \tfrac{1}{2t_k}|\cdot - x^k|^2. \tag{3.1}$$

We have $\nabla \bar\phi_k(y^{k+1}) = 0$ from $\nabla \bar f_S^k = p^k = (x^k - y^{k+1})/t_k$ (cf. (2.13)–(2.14)), and $\bar f_S^k(y^{k+1}) = \check f_k(y^{k+1})$ by (2.12), so $\bar\phi_k(y^{k+1}) = \phi_k(y^{k+1})$ (cf. (2.4)) and by Taylor's expansion,

$$\bar\phi_k(\cdot) = \phi_k(y^{k+1}) + \tfrac{1}{2t_k}|\cdot - y^{k+1}|^2. \tag{3.2}$$

By (3.1) and (2.11), we have $\bar\phi_k(x^k) = \bar f_S^k(x^k) \le f(x^k) + \epsilon_g$ (using $x^k \in S$); hence by (3.2),

$$\phi_k(y^{k+1}) + \tfrac{1}{2t_k}|y^{k+1} - x^k|^2 = \bar\phi_k(x^k) \le f(x^k) + \epsilon_g. \tag{3.3}$$

Now, using $x^{k+1} = x^k$, $t_{k+1} \le t_k$, and $\check f_S^{k+1} \ge \bar f_S^k$ in (2.4) and (3.1) gives $\phi_{k+1} \ge \bar\phi_k$, so

$$\phi_k(y^{k+1}) + \tfrac{1}{2t_k}|y^{k+2} - y^{k+1}|^2 \le \phi_{k+1}(y^{k+2}) \tag{3.4}$$

by (3.2). Since $x^k = x^{\bar{k}}$ and $t_k \le t_{\bar{k}}$ for $k \ge \bar{k}$, by (3.3)–(3.4) there exists $\phi_\infty \le f(x^{\bar{k}}) + \epsilon_g$ such that

$$\phi_k(y^{k+1}) \uparrow \phi_\infty, \quad y^{k+2} - y^{k+1} \to 0, \tag{3.5}$$

and $\{y^{k+1}\}$ is bounded. Then $\{g^k\}$ is bounded as well, since $g^k \in \partial_\epsilon f(y^k)$ with $\epsilon := \epsilon_f + \epsilon_g$ by (2.2), whereas $\partial_\epsilon f$ is locally bounded [HUL93, section XI.4.1].

We now show that the *approximation error* $\check{\epsilon}_k := f_y^{k+1} - \check{f}_k(y^{k+1})$ vanishes. Using the form (2.2) of f_{k+1}, the bound $f_{k+1} \le \check{f}_{k+1}$ (cf. (2.3)), the Cauchy–Schwarz inequality, and (2.4) with $x^k = x^{\bar{k}}$ and $t_{k+1} \le t_k$ for $k \ge \bar{k}$, we estimate

$$\begin{aligned}
\check{\epsilon}_k &:= f_y^{k+1} - \check{f}_k(y^{k+1}) = f_{k+1}(y^{k+2}) - \check{f}_k(y^{k+1}) + \langle g^{k+1}, y^{k+1} - y^{k+2}\rangle \\
&\le \check{f}_{k+1}(y^{k+2}) - \check{f}_k(y^{k+1}) + |g^{k+1}||y^{k+1} - y^{k+2}| \\
&= \phi_{k+1}(y^{k+2}) - \phi_k(y^{k+1}) + |g^{k+1}||y^{k+1} - y^{k+2}| \\
&\qquad - \tfrac{1}{2t_{k+1}}|y^{k+2} - x^{\bar{k}}|^2 + \tfrac{1}{2t_k}|y^{k+1} - x^{\bar{k}}|^2 \\
&\le \phi_{k+1}(y^{k+2}) - \phi_k(y^{k+1}) + |g^{k+1}||y^{k+1} - y^{k+2}| + \Delta_k, \qquad (3.6)
\end{aligned}$$

where

$$\begin{aligned}
\Delta_k &:= \tfrac{1}{2t_k}\left(|y^{k+1} - x^{\bar{k}}|^2 - |y^{k+2} - x^{\bar{k}}|^2\right) \\
&\le \tfrac{1}{2t_k}\left(|y^{k+1} - y^{k+2}|^2 + 2|y^{k+2} - y^{k+1}||y^{k+2} - x^{\bar{k}}|\right) \\
&\le \tfrac{1}{2t_k}|y^{k+1} - y^{k+2}|^2 + \left(\tfrac{1}{t_k}|y^{k+1} - y^{k+2}|^2 \tfrac{1}{t_{k+1}}|y^{k+2} - x^{\bar{k}}|^2\right)^{1/2}.
\end{aligned}$$

We have $\overline{\lim}_k \Delta_k \le 0$, since $\frac{1}{2t_k}|y^{k+1} - y^{k+2}|^2 \to 0$ by (3.4)–(3.5), whereas $\frac{1}{t_{k+1}}|y^{k+2} - x^{\bar{k}}|^2$ is bounded by (3.3). Hence using (3.5) and the boundedness of $\{g^{k+1}\}$ in (3.6) yields $\overline{\lim}_k \check{\epsilon}_k \le 0$. On the other hand, the null step condition $f_y^{k+1} > f_x^k - \kappa v_k$ for $k \ge \bar{k}$ gives

$$\check{\epsilon}_k = \left[\, f_y^{k+1} - f_x^k \,\right] + \left[\, f_x^k - \check{f}_k(y^{k+1}) \,\right] > -\kappa v_k + v_k = (1-\kappa)v_k \ge 0,$$

where $\kappa < 1$ by Step 0; thus $\check{\epsilon}_k \to 0$ and $v_k \to 0$. □

Using (2.18) we may relate the descent $v_k := f_x^k - \check{f}_k(y^{k+1})$ predicted by $\check{f}_k$ with the descent predicted by the augmented model ϕ_k in subproblem (2.4):

$$\begin{aligned}
w_k &:= f_x^k - \phi_k(y^{k+1}) = v_k - \tfrac{1}{2}t_k|p^k|^2 &\quad (3.7a)\\
&= \tfrac{1}{2}t_k|p^k|^2 + \alpha_k = \tfrac{1}{2}|d^k|^2/t_k + \alpha_k. &\quad (3.7b)
\end{aligned}$$

The above relations are convenient in showing that $|d^k| = O(t_k^{1/2})$ during a series of null steps that decrease t_k; this will be useful when $\underline{\lim}_k t_k = 0$.

LEMMA 3.4. *If Step* 4 *is entered with* $i_t^k = 0$, *then* $|d^k|^2 \le \left(t_{k(l)}|g^{k(l)}|^2 + 2\epsilon\right) t_k$.

Proof. First, suppose $k = k(l)$. Then (cf. Steps 0 and 4) $x^k = y^k$ and $f_x^k = f_y^k$, so using the bound $\check{f}_k \ge f_k$ (cf. (2.3)) in subproblem (2.4) and the form (2.2) of f_k gives

$$\phi_k(y^{k+1}) \ge \min\left\{\, f_k(\cdot) + \tfrac{1}{2t_k}|\cdot - x^k|^2 \,\right\} = f_x^k - \tfrac{t_k}{2}|g^k|^2.$$

Thus $w_{k(l)} \le \frac{t_{k(l)}}{2}|g^{k(l)}|^2$ by (3.7a). Next, suppose $k > k(l)$. Then (cf. Steps 3, 4, 6) $x^{j+1} = x^{k(l)}$ and $t_{j+1} \le t_j$ for $j = k(l){:}\ k-1$ due to $i_t^k = 0$, and hence $w_{j+1} \le w_j$ by

(3.4) and (3.7a). Thus $w_k \le w_{k(l)}$, and by (3.7b) and (2.17), $\frac{1}{2t_k}|d^k|^2 = w_k - \alpha_k \le w_{k(l)} + \epsilon$. □

We now use the safeguard (2.23) for analyzing the case of diminishing stepsizes.

LEMMA 3.5. *Suppose* $\underline{\lim}_k t_k = 0$ *at Step* 6 *and either only finitely many descent steps occur, or* $\sup_l t_{k(l)} < \infty$ *and* $\{x^k\}$ *is bounded. Then* $\underline{\lim}_k V_k = 0$ *at Step* 6.

Proof. Let C be the supremum of $t_{k(l)}|g^{k(l)}|^2 + 2\epsilon$ over the generated values of l. Note that $C < \infty$ since, if l is unbounded, then $\{g^{k(l)}\}$ is bounded because for $k = k(l)$ we have $x^k = y^k$ and $g^k \in \partial_\epsilon f(y^k)$ with $\epsilon := \epsilon_f + \epsilon_g$ by (2.2), whereas $\partial_\epsilon f$ is locally bounded.

Since $\underline{\lim}_k t_k = 0$, there is $K \subset \{1, 2, \dots\}$ such that $t_{k+1} \xrightarrow{K} 0$ at Step 6 with $t_{k+1} < t_k$ for all $k \in K$; thus $t_k \xrightarrow{K} 0$, since $t_k \le 10t_{k+1}$ at Step 6. For $k \in K$, at Step 6 we have (2.23), and at Step 4 we have $f_y^{k+1} > f_x^k - \kappa v_k$ and $i_t^k = 0$. Using $i_t^k = 0$, the definition of C and $t_k \xrightarrow{K} 0$ in Lemma 3.4 yields $|d^k|^2 \le Ct_k \xrightarrow{K} 0$, i.e., $d^k \xrightarrow{K} 0$. Thus, since $\{x^k\}$ is bounded, so are $\{y^{k+1} = x^k + d^k\}_{k\in K}$ and $\{g^{k+1} \in \partial_\epsilon f(y^{k+1})\}_{k\in K}$ because $\partial_\epsilon f$ is locally bounded.

Let $k \in K$ at Step 6. Since $f_y^{k+1} > f_x^k - \kappa v_k$ and $y^{k+1} = x^k + d^k$, using (2.2) gives

$$f_x^k - f_{k+1}(x^k) = f_x^k - f_y^{k+1} - \langle g^{k+1}, x^k - y^{k+1}\rangle \le \kappa v_k + |g^{k+1}||d^k|. \tag{3.8}$$

Now, (2.23), (3.8), and the fact $v_k = |d^k||p^k| + \alpha_k$ (cf. (2.18)) imply

$$\begin{aligned} V_k := \max\left\{ |p^k|, \alpha_k \right\} &\le f_x^k - f_{k+1}(x^k) \le \kappa\left(|d^k||p^k| + \alpha_k\right) + |g^{k+1}||d^k| \\ &\le \kappa(1 + |d^k|)\max\left\{ |p^k|, \alpha_k \right\} + |g^{k+1}||d^k| = \kappa(1 + |d^k|)V_k + |g^{k+1}||d^k|. \end{aligned} \tag{3.9}$$

Therefore, since $\kappa < 1$, $d^k \xrightarrow{K} 0$, and $\{g^{k+1}\}_{k\in K}$ is bounded, for large $k \in K$,

$$0 \le V_k \le |g^{k+1}||d^k| / \left[1 - \kappa(1 + |d^k|)\right] \xrightarrow{K} 0.$$

Thus $\lim_{k\in K} V_k = 0$. □

We may now finish the case of infinitely many consecutive null steps.

LEMMA 3.6. *Suppose there exists* $\bar{k}$ *such that only null steps occur for all* $k \ge \bar{k}$. *Then either* $T_\infty = \infty$ *and* $\underline{\lim}_k V_k' = 0$, *or* $T_\infty < \infty$ *and* $\underline{\lim}_k V_k = 0$ *at Step* 4.

Proof. If $\underline{\lim}_k t_k = 0$ at Step 6, then $\underline{\lim}_k V_k = 0$ by Lemma 3.5, so assume $\underline{\lim}_k t_k > 0$. Next, if $T_\infty = \infty$, then $\underline{\lim}_k V_k' = 0$ by Lemma 3.2, so assume $T_\infty < \infty$.

If Step 3 increases t_k for some $k = k' \ge \bar{k}$, then $t_k \ge 10t_{k-1}$ and $i_t^k \ne 0$, whereas for $k \ge k'$ Step 4 keeps $i_t^{k+1} = i_t^k \ne 0$ and Step 6 sets $t_{k+1} = t_k$, so the number of such increases must be finite (otherwise $t_k \to \infty$ and $T_\infty = \infty$, a contradiction). Hence we may assume that Step 3 doesn't increase t_k for $k \ge \bar{k}$. Then Lemma 3.3 gives $v_k \to 0$. Since (cf. (2.20)) $V_k \le \max\{(2v_k/t_k)^{1/2}, v_k\}$ and $\underline{\lim}_k t_k > 0$, we get $V_k \to 0$. □

For analyzing the remaining case of infinitely many descent steps, we shall use the *descent indicator* i_k defined by $i_k := 1$ if (2.22) holds and use $i_k := 0$ otherwise.

LEMMA 3.7.

(i) *If* $f_x^\infty > -\infty$, *then* $i_k v_k \to 0$ *at Step* 4.

(ii) *If* $f_x^\infty > f_* + \epsilon_g$, *then* $\{x^k\}$ *is bounded.*

Proof. (i) At Step 4, $0 \le \kappa i_k v_k \le f_x^k - f_x^{k+1}$, so $\sum_k i_k v_k \le (f_x^1 - f_x^\infty)/\kappa < \infty$.

(ii) Pick $x \in S$ and $\gamma > 0$ such that $f_x^k > f(x) + \epsilon_g + \gamma$ for all k. Since $\langle p^k, x - x^k\rangle \le \alpha_k - \gamma$ by (2.13), $x^{k+1} - x^k = -i_k t_k p^k$ and $v_k = t_k|p^k|^2 + \alpha_k$ by (2.18), we deduce that

$$\begin{aligned}|x^{k+1}-x|^2 &= |x^k-x|^2+2\langle x^{k+1}-x^k, x^k-x\rangle+|x^{k+1}-x^k|^2\\ &\le |x^k-x|^2+2i_kt_k(\alpha_k-\gamma)+2i_kt_k^2|p^k|^2\\ &= |x^k-x|^2+2i_kt_k(v_k-\gamma).\end{aligned}$$

Since $i_kv_k\to 0$ by (i), there is k_γ such that for all $k\ge k_\gamma$, $i_k(v_k-\gamma)\le 0$ above, and hence $|x^{k+1}-x|\le|x^k-x|$. Thus $\{x^k\}$ is bounded. □

LEMMA 3.8. *If infinitely many descent steps occur, then $f_x^\infty\le f_*+\epsilon_g$.*

Proof. Suppose for contradiction $f_x^\infty>f_*+\epsilon_g$. By Lemma 3.7(ii), $\{x^k\}$ is bounded. Further, $T_\infty<\infty$, since otherwise Lemmas 3.1 and 3.2 would yield $f_x^\infty\le f_*+\epsilon_g$, a contradiction. Similarly, $\underline{\lim}_k\, t_k>0$, since otherwise Lemmas 3.1 and 3.5 would yield a contradiction. Let $K:=\{k:i_k=1\}$. Using $\underline{\lim}_k\, t_k>0$ and $v_k\xrightarrow{K}0$ (cf. Lemma 3.7(i)) in the bound $V_k\le\max\{(2v_k/t_k)^{1/2},v_k\}$ (cf. (2.20)) yields $V_k\xrightarrow{K}0$. Hence $\underline{\lim}_k\, V_k=0$, and Lemma 3.1 again gives a contradiction. □

We may now prove our principal result. Note that $f_x^k\downarrow f_x^\infty\ge f_*-\epsilon_f$ by (2.5).

THEOREM 3.9. *We have $f_x^k\downarrow f_x^\infty\le f_*+\epsilon_g$. Moreover, $\overline{\lim}_k\, f(x^k)\le f_*+\epsilon$ for $\epsilon:=\epsilon_f+\epsilon_g$ so that each cluster point x^* of $\{x^k\}$ (if any) satisfies $x^*\in S$ and $f(x^*)\le f_*+\epsilon$.*

Proof. To get $f_x^\infty\le f_*+\epsilon_g$, invoke Lemmas 3.1 and 3.6 in the case of finitely many descent steps, and invoke Lemma 3.8 otherwise. By (2.5), $\overline{\lim}_k\, f(x^k)\le\lim_k f_x^k+\epsilon_f\le f_*+\epsilon_f+\epsilon_g$. The final assertion follows from the fact $\{x^k\}\subset S$ and the closedness of S and f. □

It is instructive to examine the assumptions of the preceding results.

Remark 3.10.

(i) Inspection of the proofs of Lemmas 3.3 and 3.5 reveals that Lemmas 3.3–3.8 and Theorem 3.9 require only convexity, finiteness, and closedness of f on S and *local boundedness* of the approximate subgradient mapping $g.$ on S. In particular, it suffices to assume that f is finite convex on a neighborhood of S, since $g.\in\partial_\epsilon f(\cdot)$.

(ii) For Lemma 3.5, it suffices to assume boundedness of $\{g^k\}$ instead of local boundedness of $g.$ and boundedness of $\{x^k\}$. Note that $\{x^k\}$ is bounded if f_S is coercive, since then the level set $\{x\in S: f(x)\le f_x^1+\epsilon_f\}$ is bounded and contains $\{x^k\}$ by (2.5).

The next result will justify the stopping criteria of section 4.2.

LEMMA 3.11. *Suppose $f_*>-\infty$, and either $\{g^k\}$ is bounded, or $g.$ is locally bounded and $\{x^k\}$ is bounded (e.g., f_S is coercive). Then $\underline{\lim}_k\, V_k'=0$.*

Proof. If only finitely many descent steps occur, then the proofs of Lemma 3.6 and Remark 3.10 yield $\underline{\lim}_k\, V_k'=0$. Hence suppose for contradiction that $\underline{\lim}_k\, V_k'>0$ for infinitely many descent steps.

We have $T_\infty<\infty$, since otherwise Lemma 3.2 would yield $\underline{\lim}_k\, V_k'=0$. Similarly, $\underline{\lim}_k\, t_k>0$, since otherwise Lemma 3.5 and Remark 3.10(ii) would imply $\underline{\lim}_k\, V_k=0$. Next, $f_x^k\ge f(x^k)-\epsilon_f\ge f_*-\epsilon_f>-\infty$ (cf. (2.5)) gives $f_x^\infty>-\infty$. Let $K:=\{k:i_k=1\}$. Using $\underline{\lim}_k\, t_k>0$ and $v_k\xrightarrow{K}0$ (cf. Lemma 3.7(i)) in the bound $V_k\le\max\{(2v_k/t_k)^{1/2},v_k\}$ (cf. (2.20)) yields $V_k\xrightarrow{K}0$ and hence $\underline{\lim}_k\, V_k'=0$, a contradiction. □

4. Modifications.

4.1. Subgradient aggregation.

To trade off storage and work per iteration for speed of convergence, one may replace subgradient selection with aggregation as in

[Kiw90] so that only $M \geq 2$ subgradients are stored. To this end, we note that the preceding results remain valid if, for each k, $\check{f}_{k+1}$ is a closed convex function such that $\partial(\check{f}_{k+1} + i_S) = \partial\check{f}_{k+1} + \partial i_S$ (cf. (2.7)) and

$$\max\left\{\, \bar{f}_k(x), f_{k+1}(x) \,\right\} \leq \check{f}_{k+1}(x) \leq f(x) + \epsilon_g \quad \text{for all } x \in S. \tag{4.1}$$

Examples include $\check{f}_{k+1} = \max\{\bar{f}_k, f_{k+1}\}$, or $\check{f}_{k+1} = \max\{\bar{f}_k, f_j : j \in J^{k+1}\}$ with $k+1 \in J^{k+1} \subset \{1\colon k+1\}$, and possibly some f_j replaced by $\bar{f}_j$ for $j \leq k$. In fact the aggregate linearization $\bar{f}_k$ may be omitted in (4.1) after a descent step.

4.2. Optimality measures and stopping criteria. In practice Step 2 may use the stopping criterion $V_k \leq \epsilon_{\mathrm{opt}}$, where $\epsilon_{\mathrm{opt}} > 0$ is an *optimality tolerance*. Then any loop between Steps 1 and 3 is finite (cf. the proof of Lemma 2.3(iii)), whereas Lemma 3.11 gives conditions that ensure finite termination.

It may be more appropriate to replace V_k by the *modified optimality measure*

$$\hat{V}_k := R|p^k| + \alpha_k^+ \quad \text{with} \quad \alpha_k^+ := \max\{\alpha_k, 0\}, \tag{4.2}$$

where $R > 0$ is the "radius of the picture" [HUL93, Note XIV.3.4.3^6], because the optimality estimate (2.15) combined with $f(x^k) \leq f_x^k + \epsilon_f$ (cf. (2.5)) gives the bounds

$$f(x^k) - \min_{|x-x^k|\leq R} f_S(x) - \epsilon \leq f_x^k - \min_{|x-x^k|\leq R} f_S(x) - \epsilon_g \leq R|p^k| + \alpha_k. \tag{4.3}$$

Since $\min\{R, 1\}V_k \leq \hat{V}_k \leq (R+1)V_k$ by (2.16) and (4.2), the preceding results hold with V_k replaced by $\hat{V}_k$, also in the safeguard (2.23) of Step 6, since (3.9) may be replaced by

$$\begin{aligned} &\hat{V}_k := R|p^k| + \alpha_k^+ \leq f_x^k - f_{k+1}(x^k) \leq \kappa\left(|d^k||p^k| + \alpha_k\right) + |g^{k+1}||d^k| \\ &\leq \kappa(1 + |d^k|/R)(R|p^k| + \alpha_k^+) + |g^{k+1}||d^k| = \kappa(1 + |d^k|/R)\hat{V}_k + |g^{k+1}||d^k|. \end{aligned} \tag{4.4}$$

In view of (4.3), another optimality measure $\bar{V}_k := R|p^k| + \alpha_k$ may replace V_k both in the stopping criterion (since $\bar{V}_k \leq \hat{V}_k \leq (R+1)V_k$) and in the safeguard (2.23), which becomes

$$f_x^k - f_{k+1}(x^k) \geq \bar{V}_k := R|p^k| + \alpha_k. \tag{4.5}$$

LEMMA 4.1. *Suppose Step* 6 *employs the safeguard* (4.5) *instead of* (2.23)*. Then Lemma* 3.5*, Remark* 3.10*, and Lemma* 3.11 *remain true.*

Proof. We give only two replacements for (3.9). First, for $k \in K_+ := \{k \in K : \alpha_k \geq 0\}$, we have $\bar{V}_k = \hat{V}_k$ in (4.5), so (4.4) holds. Hence if K_+ is infinite, then $\hat{V}_k \xrightarrow{K_+} 0$ by the previous argument, and thus $V_k \xrightarrow{K_+} 0$ because $V_k \leq \hat{V}_k / \min\{R, 1\}$. Otherwise $K_- := \{k \in K : \alpha_k < 0\}$ is infinite. Let $k \in K_-$. Then $V_k := \max\{|p^k|, \alpha_k\} = |p^k|$, whereas $v_k \geq -\alpha_k$ and (2.18) yield $\alpha_k \geq -\frac{1}{2}t_k|p^k|^2 = -\frac{1}{2}|d^k||p^k|$, so $\bar{V}_k := R|p^k| + \alpha_k \geq (R - \frac{1}{2}|d^k|)V_k$. Hence using (4.5) we may replace (3.9) by

$$(R - \tfrac{1}{2}|d^k|)V_k \leq f_x^k - f_{k+1}(x^k) \leq \kappa|d^k||p^k| + |g^{k+1}||d^k| = \kappa|d^k|V_k + |g^{k+1}||d^k|$$

to get $V_k \xrightarrow{K_-} 0$ as before. □

4.3. Tests for stepsize expansion and descent. Consider replacing the test $v_k \geq -\alpha_k$ of Step 3 by the stronger test $\kappa_v v_k \geq -\alpha_k$ with a fixed coefficient $\kappa_v \in (0,1)$. The preceding results are not impaired, since (2.20)–(2.21) are replaced by

$$V_k \leq \max\left\{ [(1+\kappa_v)v_k/t_k]^{1/2}, v_k \right\} \qquad \text{if} \quad \kappa_v v_k \geq -\alpha_k,$$

$$V_k < [-(1+\kappa_v^{-1})\alpha_k/t_k]^{1/2} \leq [(1+\kappa_v^{-1})\epsilon/t_k]^{1/2} \quad \text{if} \quad \kappa_v v_k < -\alpha_k.$$

Further, the facts that $v_k = t_k|p^k|^2 + \alpha_k$ (cf. (2.18)), $w_k = \frac{1}{2}t_k|p^k|^2 + \alpha_k$ (cf. (3.7b)), and $\kappa_v v_k \geq -\alpha_k$ at Step 4 yield the bounds

$$w_k \leq v_k \leq \tfrac{2}{1-\kappa_v} w_k. \tag{4.6}$$

These bounds allow us to replace v_k by w_k in the descent test (2.22), thus bringing it closer to those of [HUL93, Alg. XV.3.1.4] and [Kiw90, section 5]. Again the preceding results extend easily (in the proof of Lemma 3.3, $f_y^{k+1} > f_x^k - \kappa w_k$ implies $f_y^{k+1} > f_x^k - \kappa v_k$, whereas in the proof of Lemma 3.7(i), $\sum_k i_k v_k \leq \frac{2}{1-\kappa_v}\sum_k i_k w_k < \infty$).

For $\kappa_v = \frac{1}{3}$, we have $w_k \leq v_k \leq 3w_k$ by (4.6), whereas the test $\kappa_v v_k \geq -\alpha_k$ is equivalent to $w_k \geq -\alpha_k$. Note that $w_k \geq 0$ is equivalent to the original test $v_k \geq -\alpha_k$.

4.4. Zigzag searches. Our analysis may accommodate zigzag searches (cf. [HUL93, section XV.3.3], [Hin01], [Kiw96], [ScZ92]), which amount to trying possibly more than one value of t_k at each iteration.

We first consider stepsize expansion at descent steps. Suppose that the descent test (2.22) holds, but $t_k < T_k$ and some other tests, e.g., $f_y^{k+1} \leq f_x^k - \bar{\kappa} v_k$ or $\langle g^{k+1}, d^k\rangle < -\bar{\kappa} v_k$ with $\bar{\kappa} \in (\kappa, 1)$, indicate that larger descent might occur if t_k were increased. Letting $\underline{t}_k := t_k$, we may choose a larger $t_k \in (\underline{t}_k, T_k]$ and go back to Step 1. If (2.22) fails when Step 4 is reentered, then a descent step must be made with t_k reset to $\underline{t}_k$. Otherwise, either a descent step with the current t_k is accepted, or a larger stepsize may be tested as above.

One may use simple safeguards, such as $1.1\underline{t}_k \leq T_k$ and $t_k \geq 1.1\underline{t}_k$, to ensure finiteness of the loop between Steps 1 and 4. (If Step 3 drove t_k and T_k to ∞, the conclusions of Lemma 2.3(iii) would hold by its proof, so a cycle between Steps 1 and 3 would occur by Lemma 2.3(iv).) In effect, the preceding results are not affected by such modifications.

To enable zigzag searches at null steps, it suffices to redefine $\check{f}_{k+1}$ after Step 6 as

$$\check{f}_{k+1} := \check{f}_k \quad \text{if} \quad t_{k+1} \leq 0.9t_k. \tag{4.7}$$

Then "$t_{k+1} \leq t_k$" in Lemma 3.3 must be replaced by "$0.9t_k < t_{k+1} \leq t_k$," but this is enough for the proof of Lemma 3.6, since if $\underline{\lim}_k t_k > 0$ and $t_{k+1} \leq t_k$ for $k \geq \bar{k}$, then $t_{k+1} > 0.9t_k$ for all large k. The remaining results are not affected.

4.5. Ad hoc modification. Our analysis also sheds light on the behavior of the original proximal bundle method [Kiw90], [HUL93, section XV.3] in the inexact case.

Consider the following crippled version of Algorithm 2.1 with the safeguard (2.23) replaced by (4.5). Suppose Step 2 employs any of the stopping criteria of section 4.2 with a positive optimality tolerance ϵ_{opt}, whereas Step 3 is replaced by

Step 3′ (inaccuracy detection). If $w_k < 0$, then stop; else set $T_{k+1} := T_k$.

This version is an ad hoc modification of the method of [Kiw90] that employs only the additional stopping criterion $w_k < 0$; in fact most existing implementations use this criterion anyway (to detect QP inaccuracy or erroneous subgradients).

As for convergence of this modification, there are three cases. First, if no termination occurs, then the results of section 3 apply (with $T_\infty = T_1$); in view of Lemma 3.11, this case is quite unlikely. Second, termination at Step 2 means a satisfactory solution has been found. Third, termination at Step 3′ implies $V_k < (2\epsilon/t_k)^{1/2}$ (cf. (2.21)); thus x^k is a satisfactory solution if t_k is "large enough"; otherwise a failure occurs.

The above analysis suggests that the existing bundle codes may behave reasonably well in the inexact case, provided large enough stepsizes are used (most codes allow the user to choose the initial stepsize and its updating strategies). Of course, in case of failure, the user may choose a larger stepsize, disallow stepsize decreases, and restart the algorithm at Step 1; such a "natural" strategy reinvents Algorithm 2.1! Finally, note that the existing codes won't face any trouble until the predicted descent v_k falls below the oracle's error ϵ (since $w_k < 0$ implies $v_k < -\alpha_k \le \epsilon$ by (3.7b), (2.18), and (2.17)).

5. Lagrangian relaxation. In this section we consider the special case where problem (1.1) with $S := \mathbb{R}^n_+$ is the Lagrangian dual problem of the *primal* convex optimization problem

$$(5.1) \qquad \psi_0^{\max} := \max\ \psi_0(z) \quad \text{s.t.} \quad \psi_j(z) \ge 0,\ j = 1\colon n, \qquad z \in Z,$$

where $\emptyset \ne Z \subset \mathbb{R}^{\bar m}$ is compact and convex, and each ψ_j is concave and closed (upper semicontinuous) with $\operatorname{dom}\psi_j \supset Z$. The Lagrangian of (5.1) has the form $\psi_0(z) + \langle y, \psi(z)\rangle$, where $\psi := (\psi_1, \dots, \psi_n)$ and y is a multiplier. Suppose that, at each $y \in S$, the *dual function*

$$(5.2) \qquad f(y) := \max\{\, \psi_0(z) + \langle y, \psi(z)\rangle : z \in Z \,\}$$

can be evaluated with *accuracy* $\epsilon \ge 0$ by finding a *partial Lagrangian* ϵ*-solution*

$$(5.3) \qquad z(y) \in Z \quad \text{such that} \quad f_y := \psi_0(z(y)) + \langle y, \psi(z(y))\rangle \ge f(y) - \epsilon.$$

Thus f is finite convex and has an ϵ-subgradient mapping $g_\cdot := \psi(z(\cdot))$ on S. In view of Remark 3.10(i), we suppose that $\psi(z(\cdot))$ is locally bounded on S (e.g., $\psi(z(S))$ is bounded if $\inf_Z \min_{j=1}^n \psi_j > -\infty$, or ψ is continuous on Z). Finally, we assume that f_S is coercive, i.e., $\operatorname{Arg\,min}_S f$ is nonempty and bounded (e.g., Slater's condition holds: $\psi(\check z) > 0$ for some $\check z \in Z$).

In effect, assuming $k \to \infty$, the results of section 3 hold with $\epsilon_f := \epsilon$ and $\epsilon_g := 0$, $f_* > -\infty$, $\{x^k\}$ is bounded (cf. Remark 3.10(ii)), and Lemma 3.11 yields $\underline{\lim}_k V'_k = 0$. In particular, the partial Lagrangian solutions $z^k := z(y^k)$ (cf. (5.3)) and their constraint values $g^k := \psi(z^k)$ determine the linearizations (2.2) as Lagrangian pieces of f in (5.2):

$$(5.4) \qquad f_k(\cdot) = \psi_0(z^k) + \langle \cdot, \psi(z^k)\rangle.$$

Using their weights $\{\nu_j^k\}_{j\in J^k}$ (cf. (2.8)), we may estimate solutions to (5.1) via *aggregate primal solutions*

$$(5.5) \qquad \tilde z^k := \sum_{j \in J^k} \nu_j^k z^j.$$

We now derive useful bounds on $\psi_0(\tilde{z}^k)$ and $\psi(\tilde{z}^k)$ as in [Kiw95a, Lem. 4.1].

Lemma 5.1. *$\tilde{z}^k \in Z$, $\psi_0(\tilde{z}^k) \geq f_x^k - \alpha_k - \langle p^k, x^k \rangle$, $\psi(\tilde{z}^k) \geq p_f^k \geq p^k$.*

Proof. We have (cf. (2.8)) $\sum_{j\in J^k} \nu_j^k = 1$ with $\nu_j^k \geq 0$. Hence $\tilde{z}^k \in \mathrm{co}\{z^j\}_{j\in J^k} \subset Z$, $\psi_0(\tilde{z}^k) \geq \sum_j \nu_j^k \psi_0(z^j)$, $\psi(\tilde{z}^k) \geq \sum_j \nu_j^k \psi(z^j)$ by convexity of Z and concavity of ψ_0, ψ. Since (cf. (2.7)) $p_S^k \in \partial i_S(y^{k+1})$ with $S := \mathbb{R}_+^n$, we have $p_S^k \leq 0$ and $\langle p_S^k, y^{k+1} \rangle = 0$, so (cf. (2.14)) $p_f^k = p^k - p_S^k \geq p^k$. Next, using (2.8) and (5.4) with $\psi(z^j) =: g^j$, we get $\sum_j \nu_j^k \psi(z^j) = \sum_j \nu_j^k g^j = p_f^k$ and

$$\begin{aligned}\check{f}_k(y^{k+1}) &= \sum\nolimits_j \nu_j^k f_j(y^{k+1}) = \sum\nolimits_j \nu_j^k \left[\psi_0(z^j) + \langle y^{k+1}, \psi(z^j)\rangle\right] \\ &= \sum\nolimits_j \nu_j^k \psi_0(z^j) + \langle y^{k+1}, p_f^k \rangle.\end{aligned}$$

Rearranging and using $\langle p_S^k, y^{k+1} \rangle = 0$, $p^k := p_f^k + p_S^k$ (cf. (2.14)), (2.12), and (2.13) gives

$$\sum\nolimits_j \nu_j^k \psi_0(z^j) = \check{f}_k(y^{k+1}) - \langle p_f^k + p_S^k, y^{k+1} \rangle = \bar{f}_S^k(0) = f_x^k - \alpha_k - \langle p^k, x^k \rangle.$$

Combining the preceding relations yields the conclusion. □

The bounds of Lemma 5.1 are expressed in terms of the *primal-dual* optimality measure

$$\check{V}_k := \max\left\{ \max_{j=1:\, n} [-p_f^k]_j, \alpha_k + \langle p^k, x^k \rangle \right\} \tag{5.6}$$

as $\psi_0(\tilde{z}^k) \geq f_x^k - \check{V}_k$, $\min_{j=1}^n \psi_j(\tilde{z}^k) \geq -\check{V}_k$. Hence we may generate *record* measures $\check{V}_k^*$ and primal solutions $\tilde{z}_*^k$ as follows. At Step 0, set $\check{V}_1^* := \infty$. At Step 1, if $\check{V}_k < \check{V}_k^*$, set $\check{V}_k^* := \check{V}_k$, $\tilde{z}_*^k := \tilde{z}^k$. At Step 4 set $\check{V}_{k+1}^* := \check{V}_k^*$, $\tilde{z}_*^{k+1} := \tilde{z}_*^k$. In effect, $\check{V}_k^*$ (the current minimum of $\check{V}_j$ for $j \leq k$) measures the quality of the primal iterate

$$\tilde{z}_*^k \in Z \quad \text{with} \quad \psi_0(\tilde{z}_*^k) \geq f_x^k - \check{V}_k^*, \quad \psi_j(\tilde{z}_*^k) \geq -\check{V}_k^*, \quad j = 1\colon n. \tag{5.7}$$

We now show that $\{\tilde{z}_*^k\}$ converges to the set of *ϵ-optimal primal solutions* of (5.1)

$$Z_\epsilon := \{ z \in Z : \psi_0(z) \geq \psi_0^{\max} - \epsilon, \psi(z) \geq 0 \}. \tag{5.8}$$

Theorem 5.2.

(i) *$\{\tilde{z}_*^k\}$ is bounded and all its cluster points lie in Z.*

(ii) *$\lim_k f_x^k =: f_x^\infty \geq f_* - \epsilon$ and $\lim_k \check{V}_k^* \leq 0$.*

(iii) *Let $\tilde{z}_*^\infty$ be a cluster point of $\{\tilde{z}_*^k\}$. Then $\tilde{z}_*^\infty \in Z_\epsilon$.*

(iv) *$d_{Z_\epsilon}(\tilde{z}_*^k) := \inf_{z\in Z_\epsilon} |\tilde{z}_*^k - z| \to 0$ as $k \to \infty$.*

Proof. (i) By (5.7), $\{\tilde{z}_*^k\}$ lies in the set Z, which is compact by our assumption.

(ii) By (2.5), $f_x^k \geq f(x^k) - \epsilon_f$ with $\epsilon_f := \epsilon$ gives $f_x^\infty \geq f_* - \epsilon$. Next, since $p_f^k \geq p^k$ (cf. Lemma 5.1) implies $\max_j [-p_f^k]_j \leq |p^k|$, using (5.6) and (2.16) yields

$$\check{V}_k \leq \max\{ |p^k|, \alpha_k + \langle p^k, x^k \rangle \} \leq \max\{ |p^k|, \alpha_k \} + |p^k||x^k| \leq V_k \left(1 + |x^k|\right); \tag{5.9}$$

hence by construction $\check{V}_k^* \leq \min_{j=1}^k V_j'(1 + |x^j|)$. Recall that under our assumptions on (5.1), $\underline{\lim}_k V_k' = 0$ and $\{x^k\}$ is bounded. Therefore, $\lim_k \check{V}_k^* \leq 0$ by monotonicity.

(iii) By (i), $\tilde{z}_*^\infty \in Z$. Using (ii) in (5.7) gives $\psi_0(\tilde{z}_*^\infty) \geq f_x^\infty$, $\psi(\tilde{z}_*^\infty) \geq 0$ by closedness of ψ_0, ψ. Since $f_x^\infty \geq f_* - \epsilon$ by (ii), where $f_* \geq \psi_0^{\max}$ by weak duality (cf. (1.1), (5.1), (5.2)), we have $\psi_0(\tilde{z}_*^\infty) \geq \psi_0^{\max} - \epsilon$. Thus $\tilde{z}_*^\infty \in Z_\epsilon$ by the definition (5.8).

(iv) This follows from (i), (iii), and the continuity of the distance function d_{Z_ϵ}. □

Remark 5.3.

(i) By the proofs of Lemma 2.3(iii) and Theorem 5.2, if an infinite loop between Steps 1 and 3 occurs, then $V_k \to 0$ yields $\max\{\breve{V}_k, 0\} \to 0$ and $d_{Z_\epsilon}(\tilde{z}^k) \to 0$. Similarly, if Step 2 terminates with $V_k = 0$, then $\breve{V}_k \le 0$ and $\tilde{z}^k \in Z_\epsilon$.

(ii) Theorem 5.2 holds for $\{\tilde{z}_*^k\}$ replaced by $\{\tilde{z}^k\}_{k\in K}$ for any $K \subset \{1, 2, \dots\}$ such that $\lim_{k\in K} \max\{\breve{V}_k, 0\} = 0$; i.e., other selections could be considered.

(iii) Given a tolerance $\epsilon_{\mathrm{tol}} > 0$, the method may stop if

$$\psi_0(\tilde{z}^k) \ge f_x^k - \epsilon_{\mathrm{tol}} \quad \text{and} \quad \psi_j(\tilde{z}^k) \ge -\epsilon_{\mathrm{tol}}, \quad j = 1\colon n.$$

Then $\psi_0(\tilde{z}^k) \ge \psi_0^{\max} - \epsilon - \epsilon_{\mathrm{tol}}$ from $f_x^k \ge f_* - \epsilon$ (cf. (2.5)) and $f_* \ge \psi_0^{\max}$ (weak duality), so $\tilde{z}^k \in Z$ is an approximate solution of (5.1). This stopping criterion will be satisfied for some k (cf. (5.7) and Theorem 5.2(ii)).

No longer assuming coercivity of f_S, we still have the following.

THEOREM 5.4. *Theorem* 5.2 *holds if* $f_* > -\infty$ *and* $t_k \ge t_{\min} > 0$ *for all* k.

Proof. In view of the proof of Theorem 5.2, we need only to show that $\lim_k \breve{V}_k^* \le 0$ when infinitely many descent steps occur (since otherwise $\{x^k\}$ is bounded, whereas $\underline{\lim}_k V_k' = 0$ by Lemma 3.11).

Let $K := \{k : i_k = 1\}$. Since $v_k \xrightarrow{K} 0$ (cf. Lemma 3.7(i)) with $v_k = t_k|p^k|^2 + \alpha_k$ (cf. (2.18)) and $v_k \ge |\alpha_k|$ at Step 4, we have $\alpha_k \xrightarrow{K} 0$ and $t_k|p^k|^2 \xrightarrow{K} 0$. By (2.18), $x^{k+1} - x^k = -i_k t_k p^k$, so

$$|x^{k+1}|^2 - |x^k|^2 = i_k t_k \left\{t_k|p^k|^2 - 2\langle p^k, x^k\rangle\right\}.$$

Sum up and use the fact $\sum_k i_k t_k \ge \sum_{k\in K} t_{\min} = \infty$ to get

$$\overline{\lim_{k\in K}} \left\{t_k|p^k|^2 - 2\langle p^k, x^k\rangle\right\} \ge 0$$

(since otherwise $|x^{k+1}|^2 \to -\infty$, which is impossible). Combining this with $t_k|p^k|^2 \xrightarrow{K} 0$ yields $\underline{\lim}_{k\in K}\langle p^k, x^k\rangle \le 0$, as well as $|p^k|^2 \xrightarrow{K} 0$ by using the fact $t_k \ge t_{\min}$. Since also $\alpha_k \xrightarrow{K} 0$, we have $\underline{\lim}_{k\in K} \breve{V}_k \le 0$ by (5.9). Then the fact $\breve{V}_k^* \le \breve{V}_k$ implies $\lim_k \breve{V}_k^* \le 0$. □

Remark 5.5.

(i) For Theorem 5.4, we may impose a lower bound $t_{\min} > 0$ on t_{k+1} at Step 6, whereas $f_* > -\infty$ if problem (5.1) is feasible (by weak duality). Thus, in contrast with [FeK00], [Kiw95a], our primal recovery works even if (5.1) has no Lagrange multipliers.

(ii) Remark 5.3 remains valid under the assumptions of Theorem 5.4.

In the remainder of this section we allow the primal problem (5.1) to be nonconvex. As before, our standing assumptions are that $\{\psi_j\}_{j=0}^n$ are finite and upper semicontinuous on the compact set Z, $\psi(z(\cdot))$ is locally bounded on S, and either f_S is coercive or $f_* > -\infty$ and $t_k \ge t_{\min} > 0$ as in Theorem 5.4 (cf. Remark 5.5(i)).

Since problem (5.1) may be nonconvex, consider its *relaxed convexified version*

(5.10)

$$\psi_0^{\mathrm{rel}} := \max_{(\nu_j, z^j)_{j=1}^M} \sum_{j=1}^M \nu_j \psi_0(z^j) \quad \text{s.t.} \quad \sum_{j=1}^M \nu_j \psi(z^j) \ge 0,\ \sum_{j=1}^M \nu_j = 1,\ z^j \in Z,\ \nu_j \ge 0,$$

where $M := n+1$. Both (5.1) and (5.10) have the same dual (1.1) with $f_* = \psi_0^{\rm rel} \geq \psi_0^{\max}$; see [FeK00], [LeR01], [MSW76]. Similarly to (5.8), let $\tilde{Z}_\epsilon$ denote the set of ϵ-optimal solutions of (5.10). Such solutions may be estimated by $(\nu_j^k, z^j)_{j\in\hat{J}^k}$ with $\hat{J}^k := \{j \in J^k : \nu_j^k \neq 0\}$ as follows. Since the QP routine of [Kiw94] delivers $|\hat{J}^k| \leq M$, whereas any (ν_j^k, z^j) can be split into two elements $(\nu_j^k/2, z^j)$, we may assume $|\hat{J}^k| = M$. Denoting $(\nu_j^k, z^j)_{j\in\hat{J}^k}$ as $(\hat{\nu}_j^k, \hat{z}^{jk})_{j=1}^M$, the proof of Lemma 5.1 yields

$$\sum_{j=1}^M \hat{\nu}_j^k \psi_0(\hat{z}^{jk}) = f_x^k - \alpha_k - \langle p^k, x^k\rangle \quad \text{and} \quad \sum_{j=1}^M \hat{\nu}_j^k \psi(\hat{z}^{jk}) = p_f^k \geq p^k. \tag{5.11}$$

The record solutions $(\tilde{\nu}_j^k, \tilde{z}^{jk})_{j=1}^M$ are generated just like $\tilde{z}_*^k$ by setting $(\tilde{\nu}_j^k, \tilde{z}^{jk})_{j=1}^M := (\hat{\nu}_j^k, \hat{z}^{jk})_{j=1}^M$ at Step 1 if $\check{V}_k < \check{V}_k^*$, and $(\tilde{\nu}_j^{k+1}, \tilde{z}^{j,k+1})_{j=1}^M := (\tilde{\nu}_j^k, \tilde{z}^{jk})_{j=1}^M$ at Step 4. We now show that $(\tilde{\nu}_j^k, \tilde{z}^{jk})_{j=1}^M$ converges to $\tilde{Z}_\epsilon$, thus extending [FeK00, Thm. 6.2].

THEOREM 5.6.

(i) $\{(\tilde{\nu}_j^k, \tilde{z}^{jk})_{j=1}^M\}$ *lies in a compact set.*

(ii) $\lim_k f_x^k =: f_x^\infty \geq f_* - \epsilon$ *and* $\lim_k \check{V}_k^* \leq 0$.

(iii) *Let* $(\tilde{\nu}_j, \tilde{z}^j)_{j=1}^M$ *be a cluster point of* $\{(\tilde{\nu}_j^k, \tilde{z}^{jk})_{j=1}^M\}$. *Then* $(\tilde{\nu}_j, \tilde{z}^j)_{j=1}^M \in \tilde{Z}_\epsilon$.

(iv) $d_{\tilde{Z}_\epsilon}((\tilde{\nu}_j^k, \tilde{z}^{jk})_{j=1}^M) \to 0$ *as* $k \to \infty$.

Proof. (i) By construction (cf. (2.8)), $\sum_j \tilde{\nu}_j^k = 1$, $\tilde{\nu}_j^k > 0$, $\tilde{z}^{jk} \in Z$, a compact set.

(ii) The proofs of Theorems 5.2(ii) and 5.4 remain valid.

(iii) By (i), $\sum_j \tilde{\nu}_j = 1$, $\tilde{\nu}_j \geq 0$, $\tilde{z}^j \in Z$, $j = 1\colon M$. Next, using (ii) with $\check{V}_k^* = \check{V}_k$ (cf. (5.6)) for k such that $(\hat{\nu}_j^k, \hat{z}^{jk}) = (\tilde{\nu}_j^k, \tilde{z}^{jk})$ in (5.11) and the upper semicontinuity of ψ_0, ψ gives

$$\sum_{j=1}^M \tilde{\nu}_j \psi_0(\tilde{z}^j) \geq f_x^\infty \geq f_* - \epsilon \quad \text{and} \quad \sum_{j=1}^M \tilde{\nu}_j \psi(\tilde{z}^j) \geq 0.$$

Since $(\tilde{\nu}_j, \tilde{z}^j)_{j=1}^M$ is feasible in (5.10) and $f_* \geq \psi_0^{\rm rel}$ by weak duality (cf. (1.1), (5.2), (5.10)), we have $\sum_{j=1}^M \tilde{\nu}_j \psi_0(\tilde{z}^j) \geq \psi_0^{\rm rel} - \epsilon$; i.e., $(\tilde{\nu}_j, \tilde{z}^j)_{j=1}^M$ is an ϵ-optimal solution of (5.10).

(iv) This follows from (i), (iii), and the continuity of $d_{\tilde{Z}_\epsilon}$. □

Extensions to separable problems are easily developed as in [FeK00, section 6].

Acknowledgments. I would like to thank the associate editor and the two anonymous referees for helpful comments.

REFERENCES

[Ber99] D. P. BERTSEKAS, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, MA, 1999.

[FeK00] S. FELTENMARK AND K. C. KIWIEL, *Dual applications of proximal bundle methods, including Lagrangian relaxation of nonconvex problems*, SIAM J. Optim., 10 (2000), pp. 697–721.

[HeK02] C. HELMBERG AND K. C. KIWIEL, *A spectral bundle method with bounds*, Math. Programming, 93 (2002), pp. 173–194.

[HeR00] C. HELMBERG AND F. RENDL, *A spectral bundle method for semidefinite programming*, SIAM J. Optim., 10 (2000), pp. 673–696.

[Hin01] M. HINTERMÜLLER, *A proximal bundle method based on approximate subgradients*, Comput. Optim. Appl., 20 (2001), pp. 245–266.

[HUL93] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Springer, Berlin, 1993.

[Kiw85] K. C. Kiwiel, *An algorithm for nonsmooth convex minimization with errors*, Math. Comp., 45 (1985), pp. 173–180.

[Kiw90] K. C. Kiwiel, *Proximity control in bundle methods for convex nondifferentiable minimization*, Math. Programming, 46 (1990), pp. 105–122.

[Kiw94] K. C. Kiwiel, *A Cholesky dual method for proximal piecewise linear programming*, Numer. Math., 68 (1994), pp. 325–340.

[Kiw95a] K. C. Kiwiel, *Approximations in proximal bundle methods and decomposition of convex programs*, J. Optim. Theory Appl., 84 (1995), pp. 529–548.

[Kiw95b] K. C. Kiwiel, *Finding normal solutions in piecewise linear programming*, Appl. Math. Optim., 32 (1995), pp. 235–254.

[Kiw96] K. C. Kiwiel, *Restricted step and Levenberg–Marquardt techniques in proximal bundle methods for nonconvex nondifferentiable optimization*, SIAM J. Optim., 6 (1996), pp. 227–249.

[LeR01] C. Lemaréchal and A. Renaud, *A geometric study of duality gaps, with applications*, Math. Programming, 90 (2001), pp. 399–427.

[Mil01] S. A. Miller, *An Inexact Bundle Method for Solving Large Structured Linear Matrix Inequalities*, Ph.D. thesis, Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA, 2001.

[MSW76] T. L. Magnanti, J. F. Shapiro, and M. H. Wagner, *Generalized linear programming solves the dual*, Management Sci., 22 (1976), pp. 1195–1203.

[ScZ92] H. Schramm and J. Zowe, *A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results*, SIAM J. Optim., 2 (1992), pp. 121–152.

[Sol03] M. V. Solodov, *On approximations with finite precision in bundle methods for nonsmooth optimization*, J. Optim. Theory Appl., 119 (2003), pp. 151–165.

[Tod01] M. J. Todd, *Semidefinite optimization*, Acta Numer., 10 (2001), pp. 515–560.

SIAM J. OPTIM.
Vol. 16, No. 4, pp. 1024–1043

STOCHASTIC ORDER RELATIONS AND LATTICES OF PROBABILITY MEASURES[*]

ALFRED MÜLLER[†] AND MARCO SCARSINI[‡]

Abstract. We study various partially ordered spaces of probability measures and we determine which of them are lattices. This has important consequences for optimization problems with stochastic dominance constraints. In particular we show that the space of probability measures on $\mathbb{R}$ is a lattice under most of the known partial orders, whereas the space of probability measures on $\mathbb{R}^d$ typically is not. Nevertheless, some subsets of this space, defined by imposing strong conditions on the dependence structure of the measures, are lattices.

Key words. copula, zonoid, lattice, univariate stochastic orders, multivariate stochastic orders, optimization with stochastic dominance constraints

AMS subject classifications. 60E15, 06B23

DOI. 10.1137/040611021

1. Introduction.

A partially ordered set is called a lattice if every pair of elements has a supremum and an infimum. A great deal of literature has appeared in recent decades about ordered sets of probability measures (see, for instance, [34] and [25] for the state of the art), but surprisingly little attention has been given to the lattice structure of these sets. To the best of our knowledge the only exceptions are [19], [17], [7], [11], and [20], [21].

Lattice structures of ordered sets of probability measures have important implications for optimization problems in which ordered sets of probability measures occur as constraint sets. As a concrete example of an application in which a lattice structure is very helpful we consider optimization problems with stochastic dominance constraints as considered, e.g., in [5]. There, optimization problems of the form

$$\max \ f(X) \tag{1.1}$$
$$\text{subject to } X \leq_* Y_i, \quad i = 1, \dots, n, \tag{1.2}$$
$$X \in C \tag{1.3}$$

are considered, where f is some real valued functional, C is a set of random variables, and $\leq_*$ is some stochastic order relation. If the stochastic order relation $\leq_*$ leads to a lattice, then the problem with multiple stochastic dominance constraints is equivalent to the problem

$$\begin{aligned} &\max \ f(X) \\ &\text{subject to } X \leq_* Y_1 \wedge_* \cdots \wedge_* Y_n, \\ &X \in C \end{aligned}$$

[*] Received by the editors July 5, 2004; accepted for publication (in revised form) June 24, 2005; published electronically February 3, 2006. This research was partially supported by the Vigoni Program and MIUR-COFIN.

http://www.siam.org/journals/siopt/16-4/61102.html

[†] Institut für Wirtschaftstheorie und Operations Research, Universität Karlsruhe, Geb. 20.21, D-76128 Karlsruhe, Germany (mueller@wior.uni-karlsruhe.de).

[‡] Dipartimento di Statistica e Matematica Applicata, Università di Torino, Piazza Arbarello 8, I–10122 Torino, Italy (marco.scarsini@unito.it).

with only one constraint, which is much easier to solve. Similar optimization problems can be found in [8] and [16].

In several other fields of research it is also often useful to resort to classes of distributions rather than one single distribution. This happens, for instance, in robust Bayesian statistics, where one considers families of prior distributions such as the so-called ε-contamination classes (see, e.g., [2], [3], and [28]).

Whereas in decision theory à la Savage the decision maker maximizes her expected utility with respect to her subjective probability measure, more recent developments in the field have led to paradigms of choice that involve a whole class of probability measures rather than a single probability. This allows us to incorporate in the model the idea of ambiguity (see, e.g., [12] and the subsequent literature).

An interesting area in which the concepts of robustness and ambiguity coexist is robust control under economic model uncertainty (see, for instance, [14] and [13]).

In mathematical finance, classes of equivalent martingale measures occur when dealing with incomplete markets (see, e.g., [18]).

In all these situations it may be useful to compute bounds with respect to some order. That is, given a class C of distributions, it may be interesting to find, among all the distributions that are larger than all distributions in C, the smallest one, where of course larger and smaller refer to some prespecified partial order. For the above problem to be well defined it is necessary to have a lattice structure on the space of distributions.

In this paper we will try to study this issue in a more systematic way. It will turn out that the space of probability measures on $\mathbb{R}$ (or some suitable subsets of it) is a lattice when endowed with most of the well-known stochastic orders such as usual stochastic order, convex order, dispersion order, hazard rate order, etc., whereas the space of probability measures on $\mathbb{R}^d$ is in general not a lattice. In order to obtain a lattice structure for sets of probability measures on $\mathbb{R}^d$ we need to put severe restrictions on their dependence structure.

The paper is organized as follows. Section 2 contains some preliminary definitions and results, section 3 is devoted to the space of probability measures on $\mathbb{R}$, and section 4 deals with the space of probability measures on $\mathbb{R}^d$. Section 5 studies in detail the properties of a special order.

2. Preliminaries.

In this section we will introduce notation and the majority of the definitions used in the rest of the paper.

2.1. Orders and lattices.

We first recall the basic definitions of the theory of lattices.

DEFINITION 2.1. *Let $(\mathcal{X}, \leq_*)$ be an ordered set. For $x, y \in \mathcal{X}$ let $U(x,y) = \{z \in \mathcal{X} : x \leq_* z, y \leq_* z\}$. If $U(x,y)$ has a smallest element $\tilde{z}$ such that $\tilde{z} \leq_* z$ for all $z \in U(x,y)$, then $\tilde{z}$ is called the supremum of x and y, denoted by $\tilde{z} = x \vee_* y = \sup\{x,y\}$. Similarly, if there is a unique largest element z' smaller than x and y, then this is called the infimum, denoted by $z' = x \wedge_* y = \inf\{x,y\}$.*

If $x \vee_* y$ and $x \wedge_* y$ exist for all $x, y \in \mathcal{X}$, then $(\mathcal{X}, \leq_*)$ is called a lattice.

A subset $\mathcal{Z} \subset \mathcal{X}$ of a lattice is called a sublattice *if $x, y \in \mathcal{Z}$ implies $x \vee_* y \in \mathcal{Z}$ and $x \wedge_* y \in \mathcal{Z}$. Notice that $(\mathcal{Z}, \leq_*)$ can be a lattice in its own right without being a sublattice.*

For properties of lattices the reader is referred to [4] and [1].

The following lattice and some of its sublattices will be used quite frequently. Let S be an arbitrary set and let $\mathcal{X}$ be the set of all functions $f : S \to \mathbb{R}$, endowed with the pointwise order

$$f \le g \quad \text{if} \quad f(s) \le g(s) \quad \text{for all } s \in S.$$

This is obviously a lattice with

$$f \vee g(\cdot) = \max\{f(\cdot), g(\cdot)\} \quad \text{and} \quad f \wedge g(\cdot) = \min\{f(\cdot), g(\cdot)\}. \tag{2.1}$$

Remark 2.2. Consider the lattices induced by the two orders $\preceq_1, \preceq_2$ on the same space $\mathcal{X}$. Let $\preceq_1$ be stronger than $\preceq_2$, namely, for all $x, y \in \mathcal{X}$, let $x \preceq_1 y$ imply $x \preceq_2 y$. Then $x \wedge_1 y \preceq_2 x \wedge_2 y$ and $x \vee_2 y \preceq_2 x \vee_1 y$. Therefore comparability of the orders induces comparability of the suprema and infima (with respect to the weaker order).

The following order $\le_\uparrow$ will be of interest in what follows. Let $S = [a, b] \subset \mathbb{R}$ be a finite interval and let $\mathrm{BV}(S)$ be the set of functions $F : S \to \mathbb{R}$, which are of bounded variation. Endow $\mathrm{BV}(S)$ with the following relation $\le_\uparrow$:

$$\text{For} \quad F, G \in \mathrm{BV}(S), \quad F \le_\uparrow G \quad \text{if} \quad s \mapsto G(s) - F(s) \quad \text{is increasing.}$$

Properties of $\le_\uparrow$ will be studied in section 5.

2.2. Probability. Next, we will collect some basic notation from probability that is needed in what follows.

Definition 2.3. *Given a topological space $\mathcal{Y}$, $\mathcal{B}(\mathcal{Y})$ will indicate its Borel σ-field, $\mathcal{M}(\mathcal{Y})$ will be the set of σ-additive probability measures on $(\mathcal{Y}, \mathcal{B}(\mathcal{Y}))$, and for some measure μ on $(\mathcal{Y}, \mathcal{B}(\mathcal{Y}))$, $\mathcal{M}^\mu(\mathcal{Y}) \subset \mathcal{M}(\mathcal{Y})$ will be the set of probability measures dominated by the measure μ.*

When $\mathcal{Y}$ is a linear space, then $\mathcal{M}^*(\mathcal{Y}) \subset \mathcal{M}(\mathcal{Y})$ will be the set of probability measures with finite expectation, and $\mathcal{M}_{\mathbf{a}}(\mathcal{Y}) \subset \mathcal{M}^*(\mathcal{Y})$ will be the set of probability measures with finite expectation equal to $\mathbf{a}$.

We denote by $\delta_{\mathbf{x}}$ the degenerate probability measure at $\mathbf{x}$.

Definition 2.4. *Given a probability measure $P \in \mathcal{M}(\mathbb{R})$, define F_P as the associated distribution function and $\overline{F}_P$ as the associated survival function, i.e.,*

$$F_P(x) = P((-\infty, x]),$$
$$\overline{F}_P(x) = P((x, \infty)) = 1 - F_P(x).$$

Define the quantile function as

$$F_P^{-1}(u) = \sup\{x : F_P(x) \le u\}, \quad 0 < u < 1.$$

When $P \in \mathcal{M}^\mu(\mathbb{R})$, define f_P as the associated density function and r_P as the associated hazard rate function, i.e.,

$$f_P(x) = \frac{\mathrm{d}P}{\mathrm{d}\mu}(x), \quad \mu\text{-a.s.},$$
$$r_P(x) = \frac{f_P(x)}{\overline{F}_P(x)}, \quad \mu\text{-a.s.}$$

Given $P \in \mathcal{M}^*(\mathbb{R})$, let $\overline{F}_P^{(2)}$ be the associated integrated survival function,

$$\overline{F}_P^{(2)}(x) = \int_x^\infty \overline{F}_P(t)\ \mathrm{d}t.$$

For $P \in \mathcal{M}^*(\mathbb{R}_+)$ we call m_P the associated mean residual life function, i.e.,

$$m_P(x) = \frac{\overline{F}_P^{(2)}(x)}{\overline{F}_P(x)}.$$

The name is due to the fact that, if the nonnegative random variable X has law P, then $m_P(x) = E[X - x | X > x]$.

2.3. Univariate stochastic orders. The following definitions of stochastic orders can be found, e.g., in [34] and [25].

Definition 2.5. *Given $P, Q \in \mathcal{M}(\mathbb{R})$ we define*

$$\begin{aligned} &P \leq_{\mathrm{st}} Q \quad \textit{if} \quad \int \phi \,\mathrm{d}P \leq \int \phi \,\mathrm{d}Q \quad \textit{for all increasing } \phi, \\ &P \leq_{\mathrm{disp}} Q \quad \textit{if} \quad F_P^{-1}(t) - F_P^{-1}(s) \leq F_Q^{-1}(t) - F_Q^{-1}(s) \quad \textit{for all} \;\; 0 < s < t < 1, \\ &P \leq_{\mathrm{hr}} Q \quad \textit{if} \quad t \mapsto \frac{\overline{F}_Q(t)}{\overline{F}_P(t)} \quad \textit{is increasing.} \end{aligned}$$

Given $P, Q \in \mathcal{M}^*(\mathbb{R})$ we define

$$\begin{aligned} &P \leq_{\mathrm{cx}} Q \quad \textit{if} \quad \int \phi \,\mathrm{d}P \leq \int \phi \,\mathrm{d}Q \quad \textit{for all convex } \phi, \\ &P \leq_{\mathrm{icx}} Q \quad \textit{if} \quad \int \phi \,\mathrm{d}P \leq \int \phi \,\mathrm{d}Q \quad \textit{for all increasing convex } \phi. \end{aligned}$$

Given $P, Q \in \mathcal{M}^*(\mathbb{R}_+)$ we define

$$P \leq_{\mathrm{mrl}} Q \quad \textit{if} \quad m_P(t) \leq m_Q(t) \quad \textit{for all } t \in \mathbb{R}_+.$$

Given $P, Q \in \mathcal{M}^\mu(\mathbb{R})$ we define

$$P \leq_{\mathrm{lr}} Q \quad \textit{if} \quad f_P(t) f_Q(s) \leq f_P(s) f_Q(t) \quad \textit{for all } s \leq t.$$

Remark 2.6. In all the integral orders ($\leq_{\mathrm{st}}$, $\leq_{\mathrm{cx}}$, $\leq_{\mathrm{icx}}$) the defining inequality is assumed to hold whenever the expectations exist.

Remark 2.7. If $P, Q \in \mathcal{M}^\mu(\mathbb{R})$, then $P \leq_{\mathrm{hr}} Q$ iff $r_P(x) \geq r_Q(x)$ for all $x \in \mathbb{R}$. If, furthermore, $f_P > 0$, then $P \leq_{\mathrm{lr}} Q$ iff f_Q/f_P is increasing.

Remark 2.8. Given a random variable X with law P, and a number $a \in \mathbb{R}$, we denote by P_a the law of $X + a$. Then we have

$$P \leq_{\mathrm{disp}} Q \quad \text{iff} \quad P_a \leq_{\mathrm{disp}} Q \quad \text{for all} \quad a \in \mathbb{R}.$$

Therefore the relation $\leq_{\mathrm{disp}}$ is not antisymmetric, which implies that it is not a partial order on $\mathcal{M}(\mathbb{R})$. It is a partial order on the quotient space $(\mathcal{M}(\mathbb{R})_{/\sim})$ with respect to the relation

$$P \sim Q \quad \text{iff} \quad Q = P_a \quad \text{for some} \quad a \in \mathbb{R}.$$

2.4. Multivariate stochastic orders.

Definition 2.9. *Any function $C : [0,1]^d \to [0,1]$ which is (the restriction of) a d-variate distribution function with uniform marginals on $[0,1]$ is called a* copula.

Lemma 2.10 (see [35]). *Let $P \in \mathcal{M}(\mathbb{R}^d)$. For $i \in \{1, \dots, d\}$ let $P_i \in \mathcal{M}(\mathbb{R})$ be the ith unidimensional marginal of P (i.e., $P_i(A) = P(\mathbb{R} \times \cdots \times \mathbb{R} \times A \times \mathbb{R} \cdots \times \mathbb{R})$).*

Then there exists a copula C_P such that

$$P(\times_{i=1}^d(-\infty,x_i]) = C_P(P_1((-\infty,x_1]),\dots,P_d((-\infty,x_d])).$$

For the properties of copulae the reader is referred to [32], [15], and [26]. For the properties of copulae of probability measures on more general product spaces, see [31].

Definition 2.11. *Let $\mathcal{M}^{(C)}(\mathbb{R}^d)$ be the set of probability measures with a common copula C, and let $\mathcal{M}_{\mathbf{a}}(\mathbb{R}^d)$ be the set of probability measure with expectation equal to $\mathbf{a}$.*

Definition 2.12. *A random variable X is stochastically increasing in the random vector $\mathbf{Y}$ (denoted by $X \uparrow_{\mathrm{st}} \mathbf{Y}$) if for all $\mathbf{s} \le \mathbf{t}$ we have $\mathcal{L}(X|\mathbf{Y}=\mathbf{s}) \le_{\mathrm{st}} \mathcal{L}(X|\mathbf{Y}=\mathbf{t})$, where $\mathcal{L}(X|A)$ is the conditional law of X given A.*

Definition 2.13 (see [24]). *A random vector $\mathbf{X} = (X_1,\dots,X_d)$ is said to be* conditionally increasing *(CI) if*

$$X_i \uparrow_{\mathrm{st}} (X_j, \quad j \in J) \quad \textit{for all } J \subset \{1,\dots,d\} \textit{ and } i \notin J.$$

A copula is called CI if it is the copula of the distribution of a CI random vector.

Definition 2.14. *A function $\phi : \mathbb{R}^d \to \mathbb{R}$ is called* supermodular *if*

$$\phi(\mathbf{x}) + \phi(\mathbf{y}) \le \phi(\mathbf{x} \vee \mathbf{y}) + \phi(\mathbf{x} \wedge \mathbf{y}) \quad \textit{for all } \mathbf{x},\mathbf{y},$$

where $\mathbb{R}^d$ is endowed with the usual componentwise order and the corresponding lattice structure.

A function $\phi : \mathbb{R}^d \to \mathbb{R}$ is called directionally convex *if for all $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1,2,3,4$, such that $\mathbf{x}_1 \le \mathbf{x}_2 \le \mathbf{x}_4$, $\mathbf{x}_1 \le \mathbf{x}_3 \le \mathbf{x}_4$, and $\mathbf{x}_1 + \mathbf{x}_4 = \mathbf{x}_2 + \mathbf{x}_3$,*

$$\phi(\mathbf{x}_2) + \phi(\mathbf{x}_3) \le \phi(\mathbf{x}_1) + \phi(\mathbf{x}_4).$$

Notice that a function is directionally convex if it is supermodular and convex in each variable when the others are held fixed.

For the following definitions of stochastic orders the reader is referred again to [34] and [25].

Definition 2.15. *Given $P, Q \in \mathcal{M}(\mathbb{R}^d)$ we define*

$$P \le_{\mathrm{st}} Q \quad \textit{if} \quad \int \phi \,\mathrm{d}P \le \int \phi \,\mathrm{d}Q \quad \textit{for all increasing } \phi,$$

$$P \le_{\mathrm{cx}} Q \quad \textit{if} \quad \int \phi \,\mathrm{d}P \le \int \phi \,\mathrm{d}Q \quad \textit{for all convex } \phi,$$

$$P \le_{\mathrm{sm}} Q \quad \textit{if} \quad \int \phi \,\mathrm{d}P \le \int \phi \,\mathrm{d}Q \quad \textit{for all supermodular } \phi,$$

$$P \le_{\mathrm{dcx}} Q \quad \textit{if} \quad \int \phi \,\mathrm{d}P \le \int \phi \,\mathrm{d}Q \quad \textit{for all directionally convex } \phi,$$

$$P \le_{\mathrm{lcx}} Q \quad \textit{if} \quad \int \phi \,\mathrm{d}P \le \int \phi \,\mathrm{d}Q \quad \textit{for all } \phi \textit{ such that}$$

$$\phi(\mathbf{x}) = \psi(\ell(\mathbf{x})) \textit{ with } \psi : \mathbb{R} \to \mathbb{R} \textit{ convex and } \ell : \mathbb{R}^d \to \mathbb{R} \textit{ linear},$$

$$P \le_{\mathrm{plcx}} Q \quad \textit{if} \quad \int \phi \,\mathrm{d}P \le \int \phi \,\mathrm{d}Q \quad \textit{for all } \phi \textit{ such that}$$

$$\phi(\mathbf{x}) = \psi(\ell(\mathbf{x})) \textit{ with } \psi : \mathbb{R} \to \mathbb{R} \textit{ convex and } \ell : \mathbb{R}^d \to \mathbb{R} \textit{ linear and increasing},$$

$$P \le_{\mathrm{lo}} Q \quad \textit{if} \quad P\left(\times_{i=1}^d(-\infty,x_i]\right) \le Q\left(\times_{i=1}^d(-\infty,x_i]\right) \quad \textit{for all } (x_1,\dots,x_d) \in \mathbb{R}^d,$$

$$P \le_{\mathrm{uo}} Q \quad \textit{if} \quad P\left(\times_{i=1}^d(x_i,\infty)\right) \le Q\left(\times_{i=1}^d(x_i,\infty)\right) \quad \textit{for all } (x_1,\dots,x_d) \in \mathbb{R}^d.$$

3. Lattices of measures on $\mathbb{R}$. Now we will investigate whether the orders defined in subsection 2.3 lead to a lattice structure. In case of $\leq_{\rm st}$ this is easy. The well-known fact that $P \leq_{\rm st} Q$ iff $\overline{F}_P(x) \leq \overline{F}_Q(x)$ for all $x \in \mathbb{R}$ immediately implies the following result.

Theorem 3.1. *The ordered set* $(\mathcal{M}(\mathbb{R}), \leq_{\rm st})$ *is a lattice with*

$$\overline{F}_{P\wedge_{\rm st}Q} = \min\{\overline{F}_P, \overline{F}_Q\} \quad \textit{and} \quad \overline{F}_{P\vee_{\rm st}Q} = \max\{\overline{F}_P, \overline{F}_Q\}.$$

In the next result we use the notation

$$\operatorname{vex}(f)(x) = \sup\{g(x) : g \text{ is convex and } g(y) \leq f(y) \text{ for all } y \in \mathbb{R}\}$$

for the *convex hull operator*, yielding the largest convex function smaller than a given one.

Theorem 3.2. *The ordered set* $(\mathcal{M}^*(\mathbb{R}), \leq_{\rm icx})$ *is a lattice with*

$$\begin{aligned}\overline{F}^{(2)}_{P\wedge_{\rm icx}Q} &= \operatorname{vex}\left(\min\left\{\overline{F}^{(2)}_P(x), \overline{F}^{(2)}_Q(x)\right\}\right),\\ \overline{F}^{(2)}_{P\vee_{\rm icx}Q} &= \max\left\{\overline{F}^{(2)}_P(x), \overline{F}^{(2)}_Q(x)\right\}.\end{aligned}$$

Proof. It is well known that increasing convex order can be characterized by pointwise comparison of the integrated survival functions; i.e., $P \leq_{\rm icx} Q$ holds iff

$$\overline{F}^{(2)}_P(x) = \int_x^\infty \overline{F}_P(t)\ {\rm d}t \quad \leq \quad \int_x^\infty \overline{F}_Q(t)\ {\rm d}t = \overline{F}^{(2)}_Q(x)$$

for all real x. Denote by $\mathcal{F}^{(2)}(\mathbb{R})$ the class of all integrated survival functions. $\mathcal{F}^{(2)}(\mathbb{R})$ contains all functions f that are continuous, decreasing, convex, and satisfy $\lim_{x\to-\infty} f(x) - x = a$ for some $a \in \mathbb{R}$ and $\lim_{x\to\infty} f(x) = 0$; see, e.g., Theorem 1.5.10 in [25]. Therefore the pointwise maximum of two such functions f and g again is such a function, and clearly the smallest integrated survival function is larger than f and g. The pointwise minimum of two such functions is not necessarily convex, but there is always a largest integrated survival function $h \in \mathcal{F}^{(2)}(\mathbb{R})$ smaller than f and g, namely $h = \operatorname{vex}(\min\{f, g\})$. □

The ordered set $(\mathcal{M}^*(\mathbb{R}), \leq_{\rm cx})$ is not a lattice, as $P \leq_{\rm cx} Q$ can hold only for distributions with the same mean. Therefore only the set $(\mathcal{M}^a(\mathbb{R}), \leq_{\rm cx})$ containing all distributions with some fixed mean $a \in \mathbb{R}$ can be a lattice. Since in this case $\lim_{x\to-\infty} \overline{F}^{(2)}_P(x) - x = a$, the following result can be proved exactly as Theorem 3.2.

Theorem 3.3. *For all* $a \in \mathbb{R}$ *the ordered set* $(\mathcal{M}_a(\mathbb{R}), \leq_{\rm cx})$ *is a lattice with*

$$\begin{aligned}\overline{F}^{(2)}_{P\wedge_{\rm cx}Q} &= \operatorname{vex}\left(\min\left\{\overline{F}^{(2)}_P(x), \overline{F}^{(2)}_Q(x)\right\}\right),\\ \overline{F}^{(2)}_{P\vee_{\rm cx}Q} &= \max\left\{\overline{F}^{(2)}_P(x), \overline{F}^{(2)}_Q(x)\right\}.\end{aligned}$$

The problem examined in Theorems 3.2 and 3.3 has been studied extensively in [20], [21]. The reader is also referred to [19], [7], and [11].

In the next result we investigate the lattice structure of the mean residual life order $\leq_{\rm mrl}$ for distributions on $\mathbb{R}_+$ with a finite mean.

Theorem 3.4. *The ordered set* $(\mathcal{M}^*(\mathbb{R}_+), \leq_{\rm mrl})$ *is a lattice with*

$$\overline{F}_{P\wedge_{\rm mrl}Q}(x) = \exp\left(-\int_0^x \frac{1}{\min\{m_P(t), m_Q(t)\}}\ {\rm d}t\right) \cdot \frac{\min\{m_P(0), m_Q(0)\}}{\min\{m_P(x), m_Q(x)\}},$$

$$\overline{F}_{P\vee_{\rm mrl}Q}(x) = \exp\left(-\int_0^x \frac{1}{\max\{m_P(t), m_Q(t)\}}\ {\rm d}t\right) \cdot \frac{\max\{m_P(0), m_Q(0)\}}{\max\{m_P(x), m_Q(x)\}}.$$

Proof. A function $m : \mathbb{R}_+ \to \mathbb{R}$ is a mean residual life function of some probability measure P on $\mathbb{R}_+$ iff it has the following properties: It is nonnegative, right continuous, and such that $t \mapsto m(t) + t$ is increasing, and if t_0 exists such that $m(t_0) = 0$, then $m(t) = 0$ for all $t > t_0$. If such a t_0 does not exist, then

$$\int_0^\infty \frac{1}{m(t)} \, \mathrm{d}t = \infty$$

(see [36], [33], [34]). The class of these functions is closed under pointwise minimum and maximum. As the survival function $\overline{F}_P$ is uniquely determined by the mean residual life function m_P via

$$\overline{F}_P(x) = \exp\left(-\int_0^x \frac{1}{m_P(t)} \, \mathrm{d}t\right) \cdot \frac{m_P(0)}{m_P(x)},$$

the given representation for the survival functions of $P \vee_{\mathrm{mrl}} Q$ and $P \wedge_{\mathrm{mrl}} Q$ follows. □

The following theorems use properties of the order $\leq_\uparrow$ that are proved is section 5.

THEOREM 3.5. *The ordered set* $((\mathcal{M}(\mathbb{R})_{/\sim}), \leq_{\mathrm{disp}})$ *is a lattice with*

$$F_{P \wedge_{\mathrm{disp}} Q} = \left(F_P^{-1} \wedge_\uparrow F_Q^{-1}\right)^{-1},$$
$$F_{P \vee_{\mathrm{disp}} Q} = \left(F_P^{-1} \vee_\uparrow F_Q^{-1}\right)^{-1}.$$

Proof. Notice that $P \leq_{\mathrm{disp}} Q$ iff $F_P^{-1} \leq_\uparrow F_Q^{-1}$. Therefore the assertion follows from Lemma 5.2. □

Remark 3.6. The set $(\mathcal{M}(\mathbb{R}), \leq_{\mathrm{hr}})$ is not a lattice. Notice that $P \leq_{\mathrm{hr}} Q$ holds iff $\log(\bar{F}_P) \leq_\uparrow \log(\bar{F}_Q)$. Therefore $(\mathcal{M}(\mathbb{R}), \leq_{\mathrm{hr}})$ would be a lattice if the set of logarithms of survival functions endowed with $\leq_\uparrow$ were a lattice. However, whereas $\log(\bar{F}_P) \wedge_\uparrow \log(\bar{F}_Q)$ is always a logarithm of a survival function, this is not necessarily the case for $\log(\bar{F}_P) \vee_\uparrow \log(\bar{F}_Q)$. In this case it may happen that the limit for $x \to \infty$ is finite. If, for example, P has as support all even numbers and Q has as support the odd numbers, then $\log(\bar{F}_P) \vee_\uparrow \log(\bar{F}_Q) \equiv 0$, and this obviously is not a logarithm of a survival function of a distribution on $\mathbb{R}$.

However, the order relation $\leq_{\mathrm{hr}}$ defines a lattice for distributions on the extended real line $\mathbb{R} \cup \{+\infty\}$, allowing explicit mass on $\{+\infty\}$. Thus the logarithm of the survival function is allowed to have a finite limit as $x \to \infty$. The function $f \equiv 0$, for instance, then is the logarithm of the distribution with $P(\{+\infty\}) = 1$.

THEOREM 3.7. *The set* $(\mathcal{M}(\mathbb{R} \cup \{+\infty\}), \leq_{\mathrm{hr}})$ *is a lattice with*

$$\overline{F}_{P \wedge_{\mathrm{hr}} Q} = \exp(\log(\overline{F}_P) \wedge_\uparrow \log(\overline{F}_Q)),$$
$$\overline{F}_{P \vee_{\mathrm{hr}} Q} = \exp(\log(\overline{F}_P) \vee_\uparrow \log(\overline{F}_Q)).$$

Proof. As $P \leq_{\mathrm{hr}} Q$ holds iff $\log(\bar{F}_P) \leq_\uparrow \log(\bar{F}_Q)$, this is an immediate consequence of Lemma 5.1. □

Example 3.8. We will illustrate how the suprema and infima vary with respect to the various orders by comparing two simple discrete distributions, which have the same mean and variance, and therefore are not comparable with respect to any of the mentioned orders.

Let

$$P = \frac{1}{2}\delta_1 + \frac{1}{2}\delta_3, \qquad Q = \frac{1}{8}\delta_0 + \frac{3}{4}\delta_2 + \frac{1}{8}\delta_4.$$

Then

$$\begin{aligned} P \wedge_{\mathrm{st}} Q &= \frac{1}{8}\delta_0 + \frac{3}{8}\delta_1 + \frac{3}{8}\delta_2 + \frac{1}{8}\delta_3, \\ P \vee_{\mathrm{st}} Q &= \frac{1}{8}\delta_1 + \frac{3}{8}\delta_2 + \frac{3}{8}\delta_3 + \frac{1}{8}\delta_4; \end{aligned}$$

$$\begin{aligned} P \wedge_{\mathrm{cx}} Q = P \wedge_{\mathrm{icx}} Q &= \frac{1}{4}\delta_1 + \frac{1}{2}\delta_2 + \frac{1}{4}\delta_3, \\ P \vee_{\mathrm{cx}} Q = P \vee_{\mathrm{icx}} Q &= \frac{1}{8}\delta_0 + \frac{3}{8}\delta_{4/3} + \frac{3}{8}\delta_{8/3} + \frac{1}{8}\delta_4; \end{aligned}$$

$$\begin{aligned} P \wedge_{\mathrm{disp}} Q &= \delta_1, \\ P \vee_{\mathrm{disp}} Q &= \frac{1}{8}\delta_1 + \frac{3}{8}\delta_3 + \frac{3}{8}\delta_5 + \frac{1}{8}\delta_7; \end{aligned}$$

$$\begin{aligned} P \wedge_{\mathrm{hr}} Q &= \frac{1}{8}\delta_0 + \frac{7}{16}\delta_1 + \frac{3}{8}\delta_2 + \frac{1}{16}\delta_3, \\ P \vee_{\mathrm{hr}} Q &= \delta_4; \end{aligned}$$

$$\begin{aligned} P \wedge_{\mathrm{mrl}} Q &= \frac{2}{9}\delta_1 + \frac{5}{9}\delta_2 + \frac{2}{9}\delta_3, \\ P \vee_{\mathrm{mrl}} Q &= \frac{1}{8}\delta_0 + \frac{10}{32}\delta_1 + \frac{9}{32}\delta_2 + \frac{9}{32}\delta_4. \end{aligned}$$

Notice that in most cases the support is contained in the union of the supports of P and Q, whereas

$$\mathrm{supp}(P \vee_{\mathrm{cx}} Q) \not\subseteq \mathrm{supp}(P) \cup \mathrm{supp}(Q) =: S = \{0, 1, 2, 3, 4\}.$$

Therefore $(\mathcal{M}(S), \leq_{\mathrm{cx}})$ is not a sublattice of $(\mathcal{M}(\mathbb{R}), \leq_{\mathrm{cx}})$. Notice, however, that $(\mathcal{M}(S), \leq_{\mathrm{cx}})$ is still a lattice in its own right. In the lattice $(\mathcal{M}(S), \leq_{\mathrm{cx}})$ the supremum of P and Q from the above example is given by

$$P \vee_{\mathrm{cx}} Q = \frac{1}{8}\delta_0 + \frac{1}{4}\delta_1 + \frac{1}{4}\delta_2 + \frac{1}{4}\delta_3 + \frac{1}{8}\delta_4.$$

Theorem 3.9.

(a) *For any* $P, Q \in \mathcal{M}(\mathbb{R})$ *we have*

$$\begin{aligned} \mathrm{supp}(P \vee_{\mathrm{st}} Q) &\subseteq \mathrm{supp}(P) \cup \mathrm{supp}(Q), \\ \mathrm{supp}(P \wedge_{\mathrm{st}} Q) &\subseteq \mathrm{supp}(P) \cup \mathrm{supp}(Q). \end{aligned}$$

(b) *For any* $P, Q \in \mathcal{M}_a(\mathbb{R})$ *we have*

$$\begin{aligned} \mathrm{supp}(P \vee_{\mathrm{cx}} Q) &\subseteq \mathrm{conv}(\mathrm{supp}(P) \cup \mathrm{supp}(Q)), \\ \mathrm{supp}(P \wedge_{\mathrm{cx}} Q) &\subseteq \mathrm{supp}(P) \cup \mathrm{supp}(Q). \end{aligned}$$

(c) *For any* $P, Q \in \mathcal{M}^*(\mathbb{R}_+)$ *we have*

$$\begin{aligned} \mathrm{supp}(P \vee_{\mathrm{mrl}} Q) &\subseteq \mathrm{supp}(P) \cup \mathrm{supp}(Q), \\ \mathrm{supp}(P \wedge_{\mathrm{mrl}} Q) &\subseteq \mathrm{supp}(P) \cup \mathrm{supp}(Q). \end{aligned}$$

Proof. We show the case $\wedge_{\mathrm{cx}}$. The other cases are similar. Let $K = F_{P \wedge_{\mathrm{cx}} Q}$ and fix $x \notin \operatorname{supp}(P) \cup \operatorname{supp}(Q)$. Then there is a neighborhood U_x of x, where $\bar{F}_P^{(2)}$ and $\bar{F}_Q^{(2)}$ are affine. Hence $\overline{K}^{(2)} = \operatorname{vex}(\min(\overline{F}_P^{(2)}, \overline{F}_Q^{(2)}))$ is also affine on U_x, and thus $x \notin \operatorname{supp}(P \wedge_{\mathrm{cx}} Q)$. □

As a consequence of Theorem 3.9 we get the following result. The proof is omitted.

THEOREM 3.10.

(a) *For any measurable subset $S \subset \mathbb{R}$ the partially ordered set $(\mathcal{M}(S), \leq_{\mathrm{st}})$ is a sublattice of $(\mathcal{M}(\mathbb{R}), \leq_{\mathrm{st}})$.*

(b) *For any convex subset $S \subset \mathbb{R}$ and any $a \in S$ the partially ordered set $(\mathcal{M}_a(S), \leq_{\mathrm{cx}})$ is a sublattice of $(\mathcal{M}_a(\mathbb{R}), \leq_{\mathrm{cx}})$.*

(c) *For any measurable subset $S \subset \mathbb{R}_+$ the partially ordered set $(\mathcal{M}^*(S), \leq_{\mathrm{mrl}})$ is a sublattice of $(\mathcal{M}^*(\mathbb{R}_+), \leq_{\mathrm{mrl}})$.*

Remark 3.11. It is well known that on $\mathcal{M}(\mathbb{R})$ we have

$$\leq_{\mathrm{lr}} \subset \leq_{\mathrm{hr}} \subset \leq_{\mathrm{st}} \subset \leq_{\mathrm{icx}},$$

and on $\mathcal{M}_{\mathbf{a}}(\mathbb{R})$ we have

$$\leq_{\mathrm{disp}} \subset \leq_{\mathrm{cx}} .$$

For some of the orders examined in this section we have a stronger comparability result than the one stated in Remark 2.2; that is, the infima and suprema are comparable with respect to the stronger order, as the following proposition shows.

PROPOSITION 3.12.

(a) $P \vee_{\mathrm{icx}} Q \leq_{\mathrm{st}} P \vee_{\mathrm{st}} Q$,

(b) $P \wedge_{\mathrm{st}} Q \leq_{\mathrm{st}} P \wedge_{\mathrm{icx}} Q$,

(c) $P \vee_{\mathrm{st}} Q \leq_{\mathrm{hr}} P \vee_{\mathrm{hr}} Q$,

(d) $P \wedge_{\mathrm{hr}} Q \leq_{\mathrm{hr}} P \wedge_{\mathrm{st}} Q$.

Proof.

(a) Define $H = F_{P \vee_{\mathrm{icx}} Q}$. Then $\overline{H}^{(2)}(x) = \max(\overline{F_P}^{(2)}(x), \overline{F_Q}^{(2)}(x))$. As $\overline{H}(x) = -\, \mathrm{d}^+ \overline{H}^{(2)}(x)/\mathrm{d}x$, we have that $\overline{H}(x)$ equals either $\overline{F_P}(x)$ or $\overline{F_Q}(x)$, and therefore

$$\overline{H}(x) \leq \max(\overline{F_P}(x), \overline{F_Q}(x)) = \overline{F}_{P \vee_{\mathrm{st}} Q}(x),$$

which implies the desired result.

(b) Define $K = F_{P \wedge_{\mathrm{icx}} Q}$. Then $\overline{K}^{(2)}(x) = \operatorname{vex}(\min(\overline{F_P}^{(2)}, \overline{F_Q}^{(2)}))(x)$. For fixed x we have that $\overline{K}(x)$ equals either $\overline{F}_P(x)$ or $\overline{F}_Q(x)$, or that there exists a largest interval $[a, b)$ containing x on which $\overline{K}^{(2)}$ is affine, and hence $\overline{K}$ is constant. Since in the latter case $\overline{K}^{(2)}$ equals either $\overline{F}_P^{(2)}$ or $\overline{F}_Q^{(2)}$ in the point a and it is smaller than both these functions between a and b, we then have

$$\overline{K}(x) = \overline{K}(a) \geq \min(\overline{F}_P(a), \overline{F}_Q(a)) \geq \min\{\overline{F}_P(x), \overline{F}_Q(x)\} = \overline{F}_{P \wedge_{\mathrm{st}} Q}(x),$$

which implies the desired result.

(c) Define $R = P \vee_{\mathrm{st}} Q$. Then $\overline{F_R}(x) = \max(\overline{F_P}(x), \overline{F_Q}(x))$. Let μ be a dominating measure of P and Q and let r_P and r_Q be the corresponding hazard rate functions. Then $r_R(x)$ equals either $r_P(x)$ or $r_Q(x)$ μ-a.s. Therefore

$$r_R(x) \leq \max\{r_P(x), r_Q(x)\} = r_{P \vee_{\mathrm{hr}} Q}(x) \ \mu\text{-a.s.}$$

and therefore $R \leq_{\mathrm{hr}} P \vee_{\mathrm{hr}} Q$. The proof of (d) is similar. □

THEOREM 3.13. *The following sets of probability measures on $\mathbb{R}$ are lattices if they are endowed with the order $\leq_{\mathrm{lr}}$:*

(i) *the set of all probability measures with a common finite support,*

(ii) *the set of all probability measures on a bounded interval (a, b) having a strictly positive Lebesgue density f such that $\log f$ is of locally bounded variation.*

Proof.

(i) In general, $P \leq_{\mathrm{lr}} Q$ holds if they both have densities f_P, f_Q with respect to some dominating measure such that $\log f_P \leq_{\uparrow} \log f_Q$. Therefore, if both probability measures have a common finite support $\{x_1, \dots, x_n\}$ with $x_1 < \dots < x_n$ then we get, by (5.2) and (5.3),

$$\frac{(f_{P\vee_{\mathrm{lr}}Q})(x_{i+1})}{(f_{P\vee_{\mathrm{lr}}Q})(x_i)} = \max\left\{\frac{f_Q(x_{i+1})}{f_Q(x_i)}, \frac{f_P(x_{i+1})}{f_P(x_i)}\right\} \tag{3.1}$$

and

$$\frac{(f_{P\wedge_{\mathrm{lr}}Q})(x_{i+1})}{(f_{P\wedge_{\mathrm{lr}}Q})(x_i)} = \min\left\{\frac{f_Q(x_{i+1})}{f_Q(x_i)}, \frac{f_P(x_{i+1})}{f_P(x_i)}\right\}. \tag{3.2}$$

(ii) We have

$$\log(f_{P\vee_{\mathrm{lr}}Q}) = c(\log f_P \vee_{\uparrow} \log f_Q),$$

where c is such that $\int_a^b (f_{P\vee_{\mathrm{lr}}Q})(x)\ \mathrm{d}x = 1$. Similarly,

$$\log(f_{P\wedge_{\mathrm{lr}}Q}) = c'(\log f_P \wedge_{\uparrow} \log f_Q),$$

where c' is such that $\int_a^b (f_{P\wedge_{\mathrm{lr}}Q})(x)\ \mathrm{d}x = 1$. □

Remark 3.14. For a probability measure P with values in $\{0, 1, \dots, N\}$ the so-called equilibrium rate function e_P is defined as

$$e_P(n) := \frac{P(\{n-1\})}{P(\{n\})}, \quad n = 1, \dots, N.$$

It is well known that e_P uniquely determines P and that $P \leq_{\mathrm{lr}} Q$ holds iff $e_P(n) \geq e_Q(n)$ for all $n = 1, \dots, N$; see, e.g., [34, p. 435ff]. Thus the proof of part (i) of Theorem 3.13 is not surprising. It just states that

$$e_{P\vee_{\mathrm{lr}}Q}(n) = \min\{e_P(n), e_Q(n)\} \quad \text{and} \quad e_{P\wedge_{\mathrm{lr}}Q}(n) = \max\{e_P(n), e_Q(n)\}.$$

Example 3.15. (a) Let $S = \{1, 2, 3\}$, let

$$f_P(1) = f_P(3) = \frac{1}{4}, \quad f_P(2) = \frac{1}{2},$$

and let

$$f_Q(1) = f_Q(2) = f_Q(3) = \frac{1}{3}.$$

It follows from (3.1) that

$$f_{P\vee_{\mathrm{lr}}Q}(3) = f_{P\vee_{\mathrm{lr}}Q}(2) = 2f_{P\vee_{\mathrm{lr}}Q}(1).$$

Normalization yields

$$f_{P\vee_{\rm lr}Q}(3)=f_{P\vee_{\rm lr}Q}(2)=\frac{2}{5}\quad\text{and}\quad f_{P\vee_{\rm lr}Q}(1)=\frac{1}{5}.$$

Similarly, one obtains

$$f_{P\wedge_{\rm lr}Q}(1)=f_{P\wedge_{\rm lr}Q}(2)=\frac{2}{5}\quad\text{and}\quad f_{P\wedge_{\rm lr}Q}(3)=\frac{1}{5}.$$

(b) Let $S=[0,1]$ and consider the following example for Lebesgue densities:

$$f_P(x)=\begin{cases}4x, & x\le 1/2,\\ 4-4x, & x>1/2,\end{cases}\quad\text{and}\quad f_Q(x)=1,\ 0\le x\le 1.$$

From (5.1) it follows that

$$\frac{\mathrm{d}}{\mathrm{dx}}\log f_{P\vee_{\rm lr}Q}(x)=\begin{cases}1/x, & 0<x\le 1/2,\\ 0, & x>1/2.\end{cases}$$

Normalization yields

$$f_{P\vee_{\rm lr}Q}(x)=\begin{cases}(8/3)x, & x\le 1/2,\\ 4/3, & x>1/2.\end{cases}$$

Similarly, one obtains

$$f_{P\wedge_{\rm lr}Q}(x)=\begin{cases}4/3, & x\le 1/2,\\ (8/3)(1-x), & x>1/2.\end{cases}$$

(c) The set of all probability measures with support $\mathbb{N}_0$, endowed with the order $\le_{\rm lr}$, is not a lattice. To see this, let

$$f_P(n)=(1/2)^{k+1},\quad n=2k,2k+1,\quad k=1,2,\dots,$$

and

$$f_Q(n)=(1/2)^{k+1},\quad n=2k-1,2k,\quad k=1,2,\dots.$$

A density h with h/f_P increasing and h/f_Q increasing would have to be increasing on the whole of $\mathbb{N}_0$, which is impossible. Therefore the set $\{P,Q\}$ has no upper bound with respect to $\le_{\rm lr}$. A very similar argument can be used to show that the set of all probability measures on $\mathbb{R}$ having Lebesgue densities, endowed with the order $\le_{\rm lr}$, is not a lattice. Let

$$f_P(x)=(1/2)^{k+3},\quad 2k\le|x|<2k+2,\quad k=0,1,2,\dots,$$

and

$$f_Q(x)=\begin{cases}1/4, & |x|<1,\\ (1/2)^{k+3}, & 2k-1\le|x|<2k+1,\ k=1,2,\dots.\end{cases}$$

Then a density h with h/f_P increasing and h/f_Q increasing would have to be increasing on the whole of $\mathbb{R}$ which is impossible.

4. Lattices of measures on $\mathbb{R}^d$.

In this section we will study the lattice structure of the orders defined in subsection 2.4.

THEOREM 4.1. *The following ordered sets of probability measures are lattices:*

(a) *for any copula C, the set $(\mathcal{M}^{(C)}(\mathbb{R}^d), \leq_{\mathrm{st}})$;*

(b) *for any CI copula C, and for all $\mathbf{a} \in \mathbb{R}^d$, the set $(\mathcal{M}_{\mathbf{a}}^{(C)}(\mathbb{R}^d), \leq_{\mathrm{dcx}})$;*

(c) *for any CI copula C, and for all $\mathbf{a} \in \mathbb{R}^d$, the set $(\mathcal{M}_{\mathbf{a}}^{(C)}(\mathbb{R}^d), \leq_{\mathrm{plcx}})$.*

The following lemmas will be needed.

LEMMA 4.2 (see [29], [30]). *Let $P, Q \in \mathcal{M}^{(C)}(\mathbb{R}^d)$. Then $P \leq_{\mathrm{st}} Q$ iff, for $i \in \{1, \ldots, d\}$, $P_i \leq_{\mathrm{st}} Q_i$.*

LEMMA 4.3 (see [24]). *Let C be a CI copula, and let $P, Q \in \mathcal{M}^{(C)}(\mathbb{R}^d)$. Then $P \leq_{\mathrm{dcx}} Q$ iff, for $i \in \{1, \ldots, d\}$, $P_i \leq_{\mathrm{cx}} Q_i$.*

Proof of Theorem 4.1.

(a) By Lemma 4.2 and Theorem 3.1, we obtain that the ordered set $(\mathcal{M}^{(C)}(\mathbb{R}^d), \leq_{\mathrm{st}})$ is a product of lattices, and hence is a lattice.

(b) By Lemma 4.3 and Theorem 3.3, we obtain that, if C is CI, then the set $(\mathcal{M}_{\mathbf{a}}^{(C)}(\mathbb{R}^d), \leq_{\mathrm{dcx}})$ is a product of lattices, and hence is a lattice,

(c) As $P \leq_{\mathrm{dcx}} Q$ implies $P \leq_{\mathrm{plcx}} Q$, which in turn implies $P_i \leq_{\mathrm{cx}} Q_i$ for all marginals, it follows from Lemma 4.3 that the orderings $\leq_{\mathrm{dcx}}$ and $\leq_{\mathrm{plcx}}$ are equivalent on the set of all probability measures with a fixed CI copula C. Thus the result follows immediately from part (b). □

Example 4.4. Theorem 4.1 can be helpful for solving some multivariate versions of optimization problems with stochastic ordering constraints of the type described in (1.1). Let $\mathbf{X} = (X_1, \ldots, X_d)$ describe a portfolio of d risks, corresponding, e.g., to different lines of business. Consider a portfolio optimization problem of the following type:

$$\begin{aligned} &\max \;\; f(\mathbf{X}) \\ &\text{subject to } \mathbf{X} \leq_{\mathrm{plcx}} \mathbf{Y}^{(i)}, \quad i = 1, \ldots, n, \\ &\qquad\qquad C_{\mathbf{X}} = C^+. \end{aligned} \tag{4.1}$$

Notice that $\mathbf{X} \leq_{\mathrm{plcx}} \mathbf{Y}$ is equivalent to

$$\sum_{j=1}^{d} w_j X_j \leq_{\mathrm{cx}} \sum_{j=1}^{d} w_j Y_j \qquad \text{for all } w_1, \ldots, w_d \geq 0,$$

which can be interpreted as follows: A portfolio consisting of the risks $(X_1, \ldots, X_d)$ is less risky than a portfolio consisting of the benchmark risks $(Y_1, \ldots, Y_d)$ for all possible portfolio weights $w_1, \ldots, w_d$. The assumption that the copula of $\mathbf{X}$ is given by the upper Fréchet bound C^+ (or, in other words, that the risks are comonotonic) is a quite common assumption in the calculation of risk measures for portfolios; see, e.g., [6]. The two main reasons for this are first that comonotonicity typically yields a worst case bound, and second that many risk measures are easy to evaluate in the case of comonotonicity.

It follows from Theorem 4.1 that problem (4.1) is equivalent to

$$\begin{aligned} &\max \;\; f(\mathbf{X}) \\ &\text{subject to } \mathbf{X} \leq_{\mathrm{plcx}} \mathbf{Y}^{(1)} \wedge_{\mathrm{plcx}} \cdots \wedge_{\mathrm{plcx}} \mathbf{Y}^{(n)}, \\ &\qquad\qquad C_{\mathbf{X}} = C^+, \end{aligned} \tag{4.2}$$

which in turn is equivalent to

$$\begin{aligned} &\max \quad f(\mathbf{X}) \\ &\text{subject to } X_j \leq_{\mathrm{cx}} Y_j^{(1)} \wedge_{\mathrm{cx}} \cdots \wedge_{\mathrm{cx}} Y_j^{(n)}, \quad j = 1, \ldots, d, \\ &C_{\mathbf{X}} = C^+. \end{aligned} \tag{4.3}$$

Kamae, Krengel, and O'Brien [17] were the first to recognize that, for $d > 1$, the stochastic order on $\mathbb{R}^d$ does not generate a lattice.

PROPOSITION 4.5 (see [17]). *The ordered set* $(\mathcal{M}(\mathbb{R}^d), \leq_{\mathrm{st}})$ *is not a lattice.*

The counterexample that Kamae, Krengel, and O'Brien use in their proof is the following. Consider for $d = 2$ the probability measures

$$P = \frac{1}{2}\delta_{(0,0)} + \frac{1}{2}\delta_{(1,1)}, \quad Q = \frac{1}{2}\delta_{(0,1)} + \frac{1}{2}\delta_{(1,0)}.$$

Given the upper sets

$$A = \{(x_1, x_2) : x_1 \geq 1, x_2 \geq 1\}, \quad B = \{(x_1, x_2) : x_1 \geq 0, x_2 \geq 0, \max\{x_1, x_2\} \geq 1\},$$

any upper bound (with respect to $\leq_{\mathrm{st}}$) R for P and Q has to satisfy

$$R(A) \geq \frac{1}{2}, \quad R(B) = 1.$$

The measures

$$R_1 = \frac{1}{2}\delta_{(0,1)} + \frac{1}{2}\delta_{(1,1)} \quad \text{and} \quad R_2 = \frac{1}{2}\delta_{(1,0)} + \frac{1}{2}\delta_{(1,1)}$$

are upper bounds for $\{P, Q\}$ with respect to $\leq_{\mathrm{st}}$. But $\tilde{R} \leq_{\mathrm{st}} R_1$, $\tilde{R}(A) \geq 1/2$, and $\tilde{R}(B) = 1$ imply $\tilde{R} = R_1$; therefore R_1 is a smallest upper bound. By symmetry it can be shown that R_2 is a smallest upper bound with respect to $\leq_{\mathrm{st}}$. Therefore no supremum exists for $\{P, Q\}$.

Remark 4.6. Even if Theorem 4.1(a) holds, in general two distributions in the set of all distributions on $\mathbb{R}^d$ do not have a supremum with respect to $\leq_{\mathrm{st}}$, even if they have a common copula. Consider, for instance, $\mathcal{N}(0,1) \times \delta_0$ and $\delta_0 \times \mathcal{N}(0,1)$, where $\mathcal{N}(0,1)$ is the standard normal distribution. Notice that any distribution with marginals $1/2 \cdot (\mathcal{N}^+(0,1) + \delta_0)$ is an upper bound with respect to $\leq_{\mathrm{st}}$, where we denote by $\mathcal{N}^+(0,1)$ a standard normal distribution conditioned to be positive.

PROPOSITION 4.7. *For any* $\mathbf{a} \in \mathbb{R}^d$ *the ordered set* $(\mathcal{M}_{\mathbf{a}}(\mathbb{R}^d), \leq_{\mathrm{cx}})$ *is not a lattice.*

Proof. Without loss of generality we will consider the case $\mathbf{a} = \mathbf{0}$. Any other case can be obtained by translation. Let

$$\begin{aligned} P &= \frac{1}{2}\delta_{(-1,-1)} + \frac{1}{2}\delta_{(1,1)}, \\ Q &= \frac{1}{2}\delta_{(-1,1)} + \frac{1}{2}\delta_{(1,-1)}. \end{aligned}$$

The measure

$$R = \frac{1}{4}\delta_{(-2,0)} + \frac{1}{4}\delta_{(0,-2)} + \frac{1}{4}\delta_{(2,0)} + \frac{1}{4}\delta_{(0,2)} \tag{4.4}$$

dominates both P and Q in $(\mathcal{M}_{\mathbf{0}}(\mathbb{R}^d), \leq_{\mathrm{cx}})$. This can be seen by using the idea of fusion studied in [9], [10].

In fact P can be obtained from R by fusing $\frac{1}{4}\delta_{(-2,0)}+\frac{1}{4}\delta_{(0,-2)}$ into $\frac{1}{2}\delta_{(-1,-1)}$ and $\frac{1}{4}\delta_{(2,0)}+\frac{1}{4}\delta_{(0,2)}$ into $\frac{1}{2}\delta_{(1,1)}$.

Similarly Q can be obtained from R by fusing $\frac{1}{4}\delta_{(-2,0)}+\frac{1}{4}\delta_{(0,2)}$ into $\frac{1}{2}\delta_{(-1,1)}$ and $\frac{1}{4}\delta_{(2,0)}+\frac{1}{4}\delta_{(0,-2)}$ into $\frac{1}{2}\delta_{(1,-1)}$.

Consider now a measure

$$S=\frac{5}{11}\delta_{(\frac{3}{2},\frac{3}{2})}+\frac{3}{11}\delta_{(\frac{3}{2},-4)}+\frac{3}{11}\delta_{(-4,\frac{3}{2})}.$$

Since any measure with support in the convex hull of

$$\left(\frac{3}{2},\frac{3}{2}\right),\left(\frac{3}{2},-4\right),\left(-4,\frac{3}{2}\right)$$

and expectation $(0,0)$ is convexly dominated by S (see [10]), we have that S is an upper bound for $\{P,Q\}$.

On the other hand R and S are not comparable on $(\mathcal{M}_{\mathbf{0}}(\mathbb{R}^d),\leq_{\mathrm{cx}})$, since the convex hulls of their supports are not ordered by inclusion (again see [10]).

If $P\vee_{\mathrm{cx}}Q$ existed, then it would have to be dominated by both R and S, and hence its support would have to be contained in the intersection of the convex hulls of the supports of R and S (indicated in grey in Figure 1). Assume that this is possible. Then in order to dominate P, the measure $P\vee_{\mathrm{cx}}Q$ would have to deposit mass $\frac{1}{2}$ on the segment $B=\overline{(\frac{1}{2},\frac{3}{2}),(\frac{3}{2},\frac{1}{2})}$. In order to dominate Q it would have to deposit mass $\frac{1}{2}$ on the segment $A=\overline{(-2,0),(-\frac{1}{2},\frac{3}{2})}$ and mass $\frac{1}{2}$ on the segment $C=\overline{(0,-2),(\frac{3}{2},-\frac{1}{2})}$; see Figure 1. Since the three segments are disjoint, this leads to a contradiction.

Hence $\{P,Q\}$ have no supremum in $(\mathcal{M}_{\mathbf{0}}(\mathbb{R}^d),\leq_{\mathrm{cx}})$. ◻

PROPOSITION 4.8. *The ordered set $(\mathcal{M}^*(\mathbb{R}^d),\leq_{\mathrm{lcx}})$ is not a lattice.*

In order to prove the above proposition we need the following definition and result, for which the reader is referred to [23].

DEFINITION 4.9. *Given a probability measure $P\in\mathcal{M}^*(\mathbb{R}^d)$, we define $\ell(P)$ its* lift-zonoid

$$\ell(P)=\operatorname{conv}\left\{\left(P(B),\int_B \mathbf{x}\;P(\mathrm{d}\mathbf{x})\right):B\in\mathcal{B}(\mathbb{R}^d)\right\}.$$

LEMMA 4.10 (see [22]). *For $\mathbf{a}\in\mathbb{R}^d$ and $P,Q\in\mathcal{M}_{\mathbf{a}}(\mathbb{R}^d)$, the following two conditions are equivalent:*

(a) $P\leq_{\mathrm{lcx}}Q$,

(b) $\ell(P)\subseteq\ell(Q)$.

LEMMA 4.11. *The class of zonoids in $\mathbb{R}_+^{d+1}$ having one common vertex in $\mathbf{0}$ and another in $(1,\boldsymbol{\mu})$, ordered by inclusion, is not a lattice.*

Proof. Given two sets $A,B\in\mathbb{R}^{d+1}$, let $A\oplus B=\{\mathbf{s}+\mathbf{t}:\mathbf{s}\in A,\mathbf{t}\in B\}$ be their Minkowski sum. Consider the following zonotopes in $\mathbb{R}^4$:

$$\begin{aligned}
Z_1&=\overline{\mathbf{0},\mathbf{a_1}}\oplus\overline{\mathbf{0},\mathbf{a_2}}\oplus\overline{\mathbf{0},\mathbf{a_3}},\\
Z_2&=\overline{\mathbf{0},\mathbf{b_1}}\oplus\overline{\mathbf{0},\mathbf{b_2}}\oplus\overline{\mathbf{0},\mathbf{b_3}},\\
Z_3&=\overline{\mathbf{0},\mathbf{c_1}}\oplus\overline{\mathbf{0},\mathbf{c_2}}\oplus\overline{\mathbf{0},\mathbf{c_3}},\\
Z_4&=\overline{\mathbf{0},\mathbf{d_1}}\oplus\overline{\mathbf{0},\mathbf{d_2}}\oplus\overline{\mathbf{0},\mathbf{d_3}},
\end{aligned}$$

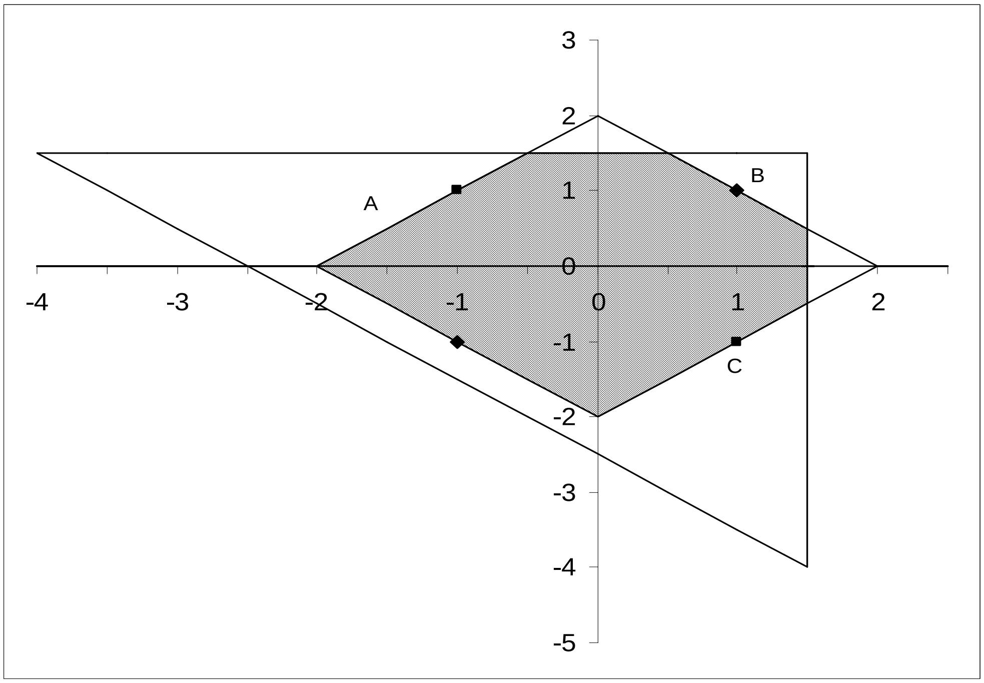


FIG. 1. *Graphical illustration of the proof of Proposition* 4.7.

where

$$\mathbf{a_1} = \left(\frac{1}{3}, \frac{2}{9}, \frac{2}{9}, \frac{5}{9}\right), \qquad \mathbf{a_2} = \left(\frac{1}{3}, \frac{2}{9}, \frac{5}{9}, \frac{2}{9}\right), \qquad \mathbf{a_3} = \left(\frac{1}{3}, \frac{5}{9}, \frac{2}{9}, \frac{2}{9}\right),$$
$$\mathbf{b_1} = \left(\frac{1}{3}, \frac{4}{9}, \frac{4}{9}, \frac{1}{9}\right), \qquad \mathbf{b_2} = \left(\frac{1}{3}, \frac{4}{9}, \frac{1}{9}, \frac{4}{9}\right), \qquad \mathbf{b_3} = \left(\frac{1}{3}, \frac{1}{9}, \frac{4}{9}, \frac{4}{9}\right),$$
$$\mathbf{c_1} = \left(\frac{1}{3}, \frac{3}{9}, \frac{4}{9}, \frac{2}{9}\right), \qquad \mathbf{c_2} = \left(\frac{1}{3}, \frac{2}{9}, \frac{3}{9}, \frac{4}{9}\right), \qquad \mathbf{c_3} = \left(\frac{1}{3}, \frac{4}{9}, \frac{2}{9}, \frac{3}{9}\right),$$
$$\mathbf{d_1} = \left(\frac{1}{3}, \frac{3}{9}, \frac{2}{9}, \frac{4}{9}\right), \qquad \mathbf{d_2} = \left(\frac{1}{3}, \frac{4}{9}, \frac{3}{9}, \frac{2}{9}\right), \qquad \mathbf{d_3} = \left(\frac{1}{3}, \frac{2}{9}, \frac{4}{9}, \frac{3}{9}\right).$$

It is not difficult to verify that the zonotopes Z_1, Z_2, Z_3, Z_4 have one vertex in $\mathbf{0}$ and the other in $\mathbf{1} := (1,1,1,1)$. Let $\mathcal{S}_2$ be the simplex

$$\mathcal{S}_2 = \left\{ (x_1, x_2, x_3) : x_1, x_2, x_3 \geq 0,\ \sum_{j=1}^{3} x_j = 1 \right\}.$$

Then each of the above zonoids is generated by segments of the type $\overline{\mathbf{0}, \left(\frac{1}{3}, x_1, x_2, x_3\right)}$, with $(x_1, x_2, x_3) \in \mathcal{S}_2$.

It is enough to look at the simplex $\mathcal{S}_2$ to notice that both Z_3 and Z_4 are included in $Z_1 \cap Z_2$, so they are lower bounds for $\{Z_1, Z_2\}$. Therefore an infimum of $\{Z_1, Z_2\}$ would have to contain Z_3 and Z_4. We observe that each of the six segments that generate Z_3 and Z_4 is extreme for the convex hull of Z_3 and Z_4, which coincides with

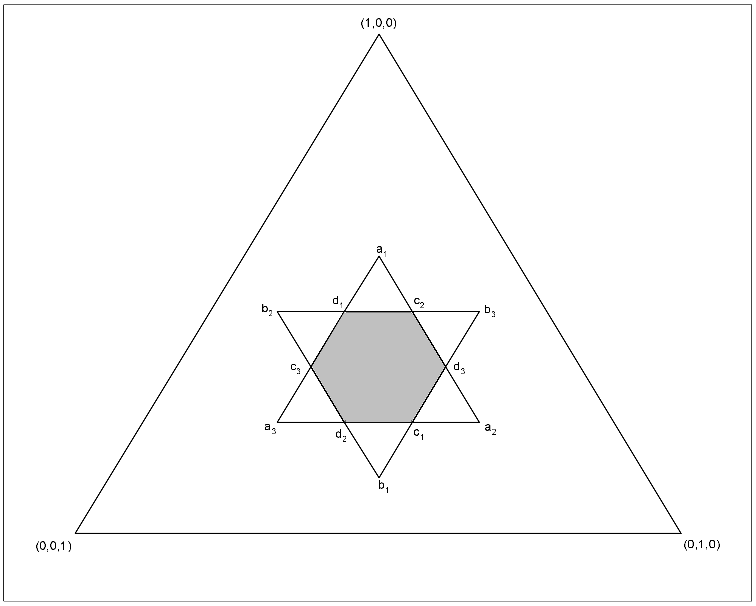


FIG. 2. *Graphical illustration of the proof of Lemma* 4.11.

the intersection of Z_1 and Z_2; see Figure 2. Therefore any zonoid that includes both Z_3 and Z_4 would have to contain all six generating segments among its generators, but then it would not have a vertex in $\mathbf{1}$. This proves that the set $\{Z_1, Z_2\}$ has no infimum. □

Proof of Proposition 4.8. It is enough to combine Lemmas 4.10 and 4.11. □

PROPOSITION 4.12. *The ordered set* $(\mathcal{M}(\mathbb{R}^d), \leq_{\mathrm{sm}})$ *is not a lattice.*

Proof. Let

$$P = \frac{1}{3}(\delta_{(2,1)} + \delta_{(1,3)} + \delta_{(3,2)}),$$
$$Q = \frac{1}{3}(\delta_{(1,2)} + \delta_{(3,1)} + \delta_{(2,3)}).$$

The measures

$$R = \frac{1}{3}(\delta_{(1,1)} + \delta_{(3,2)} + \delta_{(2,3)}),$$
$$S = \frac{1}{3}(\delta_{(1,2)} + \delta_{(2,1)} + \delta_{(3,3)})$$

are upper bounds for $\{P, Q\}$. To prove this, notice that, for instance,

$$\int f \, \mathrm{d}R - \int f \, \mathrm{d}P = \frac{1}{3}\left(f(\mathbf{x} \vee \mathbf{y}) + f(\mathbf{x} \wedge \mathbf{y}) - f(\mathbf{x}) - f(\mathbf{y})\right)$$

for $\mathbf{x} = (2,1)$, $\mathbf{y} = (1,3)$. Similar results hold for the other cases. The definition of supermodularity implies that R, S are upper bounds for $\{P, Q\}$.

The distribution function $F_{P\vee_{\rm sm}Q}$ would have to satisfy

$$F_P, F_Q \leq F_{P\vee_{\rm sm}Q} \leq F_R, F_S,$$

which implies

$$F_{P\vee_{\rm sm}Q}(1,1) = 0, \quad F_{P\vee_{\rm sm}Q}(1,2) = \frac{1}{3}, \quad F_{P\vee_{\rm sm}Q}(2,1) = \frac{1}{3}, \quad F_{P\vee_{\rm sm}Q}(2,2) = \frac{1}{3}.$$

This is not possible. □

The argument in the above proof can be used to prove also the following proposition.

PROPOSITION 4.13. *The ordered sets* $(\mathcal{M}(\mathbb{R}^d), \leq_{\rm dcx})$, $(\mathcal{M}(\mathbb{R}^d), \leq_{\rm lo})$, *and* $(\mathcal{M}(\mathbb{R}^d), \leq_{\rm uo})$ *are not lattices.*

A different argument showing that $(\mathcal{M}(\mathbb{R}^d), \leq_{\rm lo})$ is not a lattice can be derived by Example 2.1 in [27].

5. Properties of the order $\leq_\uparrow$. The relation $\leq_\uparrow$ on $\mathrm{BV}(S)$ induces a partial order $\leq_\uparrow$ on $\mathrm{BV}_{/\sim}(S)$, the set of all equivalence classes $F_{/\sim}$ defined by the equivalence relation

$$F \sim G \quad \text{if} \quad G - F \quad \text{is constant.}$$

Then $(\mathrm{BV}_{/\sim}(S), \leq_\uparrow)$ is a lattice (see, e.g., section 8.6 in [1]). It is easy to see that this can be extended to arbitrary measurable subsets $S \subset \mathbb{R}$, denoting by $\mathrm{BV}_{loc}(S)$ the set of functions $F : S \to \mathbb{R}$ that are of bounded variation on closed bounded subsets.

The subset $(\mathrm{BV}^+_{loc/\sim}(S), \leq_\uparrow)$ of all right-continuous functions of local bounded variation forms a sublattice, which is strongly related to the set of all signed measures of local bounded variation endowed with the natural partial order. Indeed, let $\mathcal{S}(S)$ be the set of all signed measures on $(S, \mathcal{B}(S))$ of local bounded variation. Define on it the order relation $\preceq$ as follows:

$$\mu \preceq \nu \quad \text{if} \quad \mu(B) \leq \nu(B) \quad \text{for all } B \in \mathcal{B}(S).$$

Then $(\mathcal{S}(S), \preceq)$ is a lattice, where $\mu \vee \nu$ and $\mu \wedge \nu$ are given as follows. Let ρ be a dominating measure of μ and ν (e.g., take $\rho = |\mu| + |\nu|$), and denote by

$$f_\mu = \frac{\mathrm{d}\mu}{\mathrm{d}\rho} \quad \text{and} \quad f_\nu = \frac{\mathrm{d}\nu}{\mathrm{d}\rho}$$

the corresponding Radon–Nikodym derivatives. Then $\mu \vee \nu$ and $\mu \wedge \nu$ are the signed measures with the Radon–Nikodym derivatives

$$\frac{\mathrm{d}(\mu \vee \nu)}{\mathrm{d}\rho} = \max\{f_\mu, f_\nu\} \quad \text{and} \quad \frac{\mathrm{d}(\mu \wedge \nu)}{\mathrm{d}\rho} = \min\{f_\mu, f_\nu\}.$$

The mapping from $(\mathrm{BV}^+_{loc/\sim}(S), \leq_\uparrow)$ to $(\mathcal{S}(S), \preceq)$, assigning to (the equivalence class of) a distribution function F the corresponding signed measure μ_F with

$$\mu_F((a, b]) = F(b) - F(a) \quad \text{for all } a, b \in S,\ a < b,$$

is a lattice isomorphism; see [1, Theorem 9.61]. There are two important special cases, where this lattice isomorphism can be used to derive explicit formulas for $F \vee_\uparrow G$ and

$F \wedge_\uparrow G$. If S is an open set, and F and G are differentiable, then

$$(F \vee_\uparrow G)'(s) = \max\{F'(s), G'(s)\} \quad \text{and} \quad (F \wedge_\uparrow G)'(s) = \min\{F'(s), G'(s)\}, \quad s \in S. \tag{5.1}$$

If $S = \mathbb{N}_0$, then

$$(F \vee_\uparrow G)(s+1) - (F \vee_\uparrow G)(s) = \max\{F(s+1) - F(s), G(s+1) - G(s)\}, \quad s \in \mathbb{N}_0, \tag{5.2}$$

and

$$(F \wedge_\uparrow G)(s+1) - (F \wedge_\uparrow G)(s) = \min\{F(s+1) - F(s), G(s+1) - G(s)\}, \quad s \in \mathbb{N}_0. \tag{5.3}$$

The following special cases of spaces of functions endowed with the order $\leq_\uparrow$ are needed in section 3. Let $\mathcal{F}^{\log}(\mathbb{R})$ be the set of all functions $f : \mathbb{R} \to \mathbb{R} \cup \{-\infty\}$, which are decreasing and right-continuous with $\lim_{x\to-\infty} f(x) = 0$; in other words $\mathcal{F}^{\log}(\mathbb{R})$ is the set of logarithms of survival functions of distributions with support $\mathbb{R} \cup \{+\infty\}$, where we define $\log(0) = -\infty$; see Remark 3.6.

LEMMA 5.1. *The partially ordered set* $(\mathcal{F}^{\log}(\mathbb{R}), \leq_\uparrow)$ *is a lattice.*

Proof. For $f, g \in \mathcal{F}^{\log}(\mathbb{R})$ define

$$\begin{aligned} S_{f,g} &= \{x \in \mathbb{R} : f(x) > -\infty, g(x) > -\infty\} \\ &=: (-\infty, \alpha_{f,g}). \end{aligned}$$

Then $(\mathrm{BV}^+_{loc/\sim}(S_{f,g}), \leq_\uparrow)$ is a lattice as described above, and therefore $f \wedge_\uparrow g(x)$ and $f \vee_\uparrow g(x)$ are well defined for $x \in S_{f,g}$. This can be extended to the whole of $\mathbb{R}$ by defining

$$f \wedge_\uparrow g(x) := -\infty \quad \text{for } x \geq \alpha_{f,g}$$

and

$$f \vee_\uparrow g(x) := \begin{cases} \lim_{t\uparrow\alpha_{f,g}} f \vee_\uparrow g(t) + g(x) - g(\alpha_{f,g}) & \text{for } x \geq \alpha_{f,g},\ g(x) > -\infty, \\ \lim_{t\uparrow\alpha_{f,g}} f \vee_\uparrow g(t) + f(x) - f(\alpha_{f,g}) & \text{for } x \geq \alpha_{f,g},\ f(x) > -\infty, \\ -\infty & \text{for } f(x) = g(x) = -\infty. \end{cases}$$

As monotonicity and right continuity are preserved under the lattice operations, it is straightforward to see that $(\mathcal{F}^{\log}(\mathbb{R}), \leq_\uparrow)$ becomes a lattice under these operations. ☐

Now let $\mathcal{Q}$ be the set of all quantile functions, i.e., the set of all right-continuous increasing functions $f : (0, 1) \to \mathbb{R}$. The proof of the following result is immediate.

LEMMA 5.2. *The partially ordered set* $(\mathcal{Q}, \leq_\uparrow)$ *is a lattice.*

Acknowledgments. The idea for this paper originated from a question posed by Nicolas Vieille during a talk of one of the authors. The authors thank Dirk Potthast for drawing the figures, Wolfgang Weil and Markus Kiderlen for insightful discussions on zonoids, and Gerd Herzog and Moshe Shaked for some useful references. Helpful comments of two referees are gratefully acknowledged.

REFERENCES

[1] C. D. Aliprantis and K. C. Border, *Infinite-Dimensional Analysis*, Springer-Verlag, Berlin, 1999.

[2] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis*, Springer-Verlag, New York, 1985.

[3] J. O. Berger, B. Betrò, E. Moreno, L. R. Pericchi, F. Ruggeri, G. Salinetti, and L. Wasserman, eds., *Bayesian Robustness*, IMS Lecture Notes Monogr. Ser. 29, Institute of Mathematical Statistics, Hayward, CA, 1996.

[4] B. A. Davey and H. A. Priestley, *Introduction to Lattices and Order*, Cambridge University Press, New York, 2002.

[5] D. Dentcheva and A. Ruszczyński, *Optimization with stochastic dominance constraints*, SIAM J. Optim., 14 (2003), pp. 548–566.

[6] J. Dhaene, S. Vanduffel, Q. Tang, M. J. Goovaerts, R. Kaas, and D. Vyncke, *Solvency Capital, Risk Measures and Comonotonicity: A Review*, DTEW Research Report 0416, Katholieke Universiteit Leuven, Leuven, The Netherlands, 2004. Available online at http://www.econ.kuleuven.be/eng/fetew/medewerker/pers_publication.aspx?PID=986

[7] L. E. Dubins and D. Gilat, *On the distribution of maxima of martingales*, Proc. Amer. Math. Soc., 68 (1978), pp. 337–338.

[8] P. Dybvig, *Distributional analysis of portfolio choice*, J. Business, 61 (1988), pp. 369–393.

[9] J. Elton and T. P. Hill, *Fusions of a probability distribution*, Ann. Probab., 20 (1992), pp. 421–454.

[10] J. Elton and T. P. Hill, *On the basic representation theorem for convex domination of measures*, J. Math. Anal. Appl., 228 (1998), pp. 449–466.

[11] D. Gilat and I. Meilijson, *A simple proof of a theorem of Blackwell & Dubins on the maximum of a uniformly integrable martingale*, in Séminaire de Probabilités, XXII, Lecture Notes in Math. 1321, Springer, Berlin, 1988, pp. 214–216.

[12] I. Gilboa and D. Schmeidler, *Maxmin expected utility with nonunique prior*, J. Math. Econom., 18 (1989), pp. 141–153.

[13] L. P. Hansen and T. J. Sargent, *Robust Control and Economic Model Uncertainty*, working paper, 2004. Available online at http://homepages.nyu.edu/˜ts43/

[14] L. P. Hansen, T. J. Sargent, and T. Tallarini, *Robust permanent income and pricing*, Rev. Econom. Stud., 66 (1999), pp. 873–907.

[15] H. Joe, *Multivariate Models and Dependence Concepts*, Chapman & Hall, London, 1997.

[16] E. Jouini and H. Kallal, *Efficient trading strategies in the presence of market frictions*, Rev. Financ. Stud., 14 (2001), pp. 343–369.

[17] T. Kamae, U. Krengel, and G. L. O'Brien, *Stochastic inequalities on partially ordered spaces*, Ann. Probab., 5 (1977), pp. 899–912.

[18] I. Karatzas and S. E. Shreve, *Methods of Mathematical Finance*, Springer-Verlag, New York, 1998.

[19] H. G. Kellerer, *Markov-Komposition und eine Anwendung auf Martingale*, Math. Ann., 198 (1972), pp. 99–122.

[20] R. P. Kertz and U. Rösler, *Stochastic and convex orders and lattices of probability measures, with a martingale interpretation*, Israel J. Math., 77 (1992), pp. 129–164.

[21] R. P. Kertz and U. Rösler, *Complete lattices of probability measures with applications to martingale theory*, in Game Theory, Optimal Stopping, Probability and Statistics, IMS Lecture Notes Monogr. Ser. 35, Institute of Mathematical Statistics, Beachwood, OH, 2000, pp. 153–177.

[22] G. Koshevoy and K. Mosler, *Lift zonoids, random convex hulls and the variability of random vectors*, Bernoulli, 4 (1998), pp. 377–399.

[23] K. Mosler, *Multivariate Dispersion, Central Regions and Depth: The Lift Zonoid Approach*, Springer-Verlag, Berlin, 2002.

[24] A. Müller and M. Scarsini, *Stochastic comparison of random vectors with a common copula*, Math. Oper. Res., 26 (2001), pp. 723–740.

[25] A. Müller and D. Stoyan, *Comparison Methods for Stochastic Models and Risks*, John Wiley & Sons Ltd., Chichester, UK, 2002.

[26] R. B. Nelsen, *An Introduction to Copulas*, Springer-Verlag, New York, 1999.

[27] R. B. Nelsen, J. J. Quesada Molina, J. A. Rodriguez Lallena, and M. Ubeda Flores, *Best-possible bounds on sets of bivariate distribution functions*, J. Multivariate Anal., 90 (2004), pp. 348–358.

[28] D. Ríos Insua and F. Ruggeri, eds., *Robust Bayesian Analysis*, Lecture Notes in Statistics 152, Springer-Verlag, New York, 2000.

[29] L. Rüschendorf, *Stochastically ordered distributions and monotonicity of the OC-function of sequential probability ratio tests*, Math. Operationsforsch. Statist. Ser. Statist., 12 (1981), pp. 327–338.

[30] M. Scarsini, *Multivariate stochastic dominance with fixed dependence structure*, Oper. Res. Lett., 7 (1988), pp. 237–240.

[31] M. Scarsini, *Copulae of probability measures on product spaces*, J. Multivariate Anal., 31 (1989), pp. 201–219.

[32] B. Schweizer and A. Sklar, *Probabilistic Metric Spaces*, North–Holland, New York, 1983.

[33] M. Shaked and J. G. Shanthikumar, *Dynamic multivariate mean residual life functions*, J. Appl. Probab., 28 (1991), pp. 613–629.

[34] M. Shaked and J. G. Shanthikumar, *Stochastic Orders and Their Applications*, Academic Press, Boston, MA, 1994.

[35] M. Sklar, *Fonctions de répartition à n dimensions et leurs marges*, Publ. Inst. Statist. Univ. Paris, 8 (1959), pp. 229–231.

[36] G. L. Yang, *Estimation of a biometric function*, Ann. Statist., 6 (1978), pp. 112–116.

SIAM J. OPTIM.
Vol. 16, No. 4, pp. 1044–1053

CRITICAL VALUE FUNCTIONS HAVE FINITE MODULUS OF CONCAVITY*

HARALD GÜNZEL[†], FRANCISCO GUERRA VAZQUEZ[‡], AND HUBERTUS TH. JONGEN[†]

Abstract. We consider a smooth finite dimensional parametric optimization problem $\mathcal{P}(y)$ with objective function $f(x, y)$. Here, x and y denote the state variable and the parameter, respectively. In the case that $\overline{x}$ is a strongly stable Karush–Kuhn–Tucker point for $\mathcal{P}(\overline{y})$, a neighborhood of $\overline{x}$ contains a unique Karush–Kuhn–Tucker point $x(y)$ for $\mathcal{P}(y)$, provided that y is sufficiently close to $\overline{y}$. This gives rise to the critical value function $y \mapsto \varphi(y) := f(x(y), y)$. Under the additional assumption that the Mangasarian–Fromovitz constraint qualification is satisfied at $\overline{x}$, we show that φ has finite modulus of concavity. That means φ becomes convex in a neighborhood of $\overline{y}$ by adding to it the function $y \mapsto (\alpha/2) \cdot \|y - \overline{y}\|^2$ for some $\alpha > 0$. Moreover, we present an explicit upper bound for the α to be used. The latter bound turns out to be sharp for problem data in general position.

Key words. parametric optimization, marginal function, critical value function, modulus of concavity

AMS subject classifications. 90C26, 90C30, 90C31

DOI. 10.1137/S1052623403434735

1. Introduction and main result.

In this paper we consider parametric optimization problems of the form

$$\mathcal{P}(y) \qquad \text{Minimize } f(x, y) \text{ on the feasible set } M(y),$$

$$M(y) := \{x \in \mathbb{R}^n \mid g_i(x, y) \leq 0,\ i \in I,\ h_j(x, y) = 0,\ j \in J\},$$

where $y \in \mathbb{R}^p$ is a parameter vector, I and J are finite index sets, and the real valued functions f, g_i, h_j are in $C^k(\mathbb{R}^n \times \mathbb{R}^p)$, $k \geq 2$. If $\overline{x} \in M(\overline{y})$ is a strongly stable Karush–Kuhn–Tucker (KKT) point for $\mathcal{P}(\overline{y})$, then there exist neighborhoods $U \ni \overline{x}$ and $V \ni \overline{y}$ and a continuous mapping $x : V \to U$ such that $x(y)$ is the unique KKT point in U for $\mathcal{P}(y)$. This gives rise to the *critical value function* $\varphi : V \to \mathbb{R}$, $\varphi(y) := f(x(y), y)$. In the case that $\overline{x}$ is a (local) minimizer for $\mathcal{P}(\overline{y})$, the function φ is a *(local) marginal function.*

DEFINITION 1. *Let ψ be a continuous real valued function defined on an open subset $V \subset \mathbb{R}^p$ and let $\overline{y} \in V$. For $\alpha \in [0, \infty)$ put $\psi^\alpha(y) := \psi(y) + \frac{\alpha}{2}\|y - \overline{y}\|^2$, where $\|.\|$ stands for the Euclidean norm. The* modulus of concavity *$\alpha_\psi(\overline{y})$ of ψ at $\overline{y}$ is defined to be the infimum over all $\alpha \in [0, \infty)$ such that ψ^α is convex in a convex neighborhood of $\overline{y}$, where* $\inf(\emptyset) := \infty$.

EXAMPLE 2. *For a C^2-function ψ, let $D^2\psi(y)$ denote the Hessian at y. Then*

$$\alpha_\psi(\overline{y}) = \max\left(\{0\} \cup \{-\lambda \mid \lambda \textit{ is eigenvalue of } D^2\psi(\overline{y})\}\right).$$

Note that the modulus of concavity might be infinite. In fact, for the function $\psi(y) := -|y|$ we have $\alpha_\psi(0) = \infty$. For later use also note that the modulus of concavity is an upper semicontinuous function, i.e., $\limsup_{y \to \overline{y}} \alpha_\psi(y) \leq \alpha_\psi(\overline{y})$.

[*] Received by the editors September 18, 2003; accepted for publication (in revised form) August 10, 2005; published electronically February 3, 2006.

http://www.siam.org/journals/siopt/16-4/43473.html

[†] RWTH, Aachen University, Department of Mathematics (C), 52056 Aachen, Germany (guenzel@RWTH-Aachen.de, jongen@RWTH-Aachen.de).

[‡] Universidad de las Americas, Department of Physics and Mathematics, Puebla 72820, Mexico, CONACYT Grand 44003 (francisco.guerra@udlap.mx).

Theorem 3 (main result). *Let $\overline{x}$ be a strongly stable KKT point for $\mathcal{P}(\overline{y})$ and let φ denote the corresponding critical value function defined in a neighborhood of $\overline{y}$. Moreover, we assume the Mangasarian–Fromovitz constraint qualification (MFCQ) for $\mathcal{P}(\overline{y})$ to be satisfied at $\overline{x}$ (for a definition see section* 2*). Then the modulus of concavity $\alpha_\varphi(\overline{y})$ is finite.*

Remark 4. The assumption in Theorem 3, that MFCQ is satisfied, can be omitted, since recently it was shown that strong stability of the KKT point $\overline{x}$ already implies MFCQ; see [3].

Section 2 contains preliminaries and some examples. In section 3 we prove Theorem 3 and explicitly present (a tight upper bound of) the modulus of concavity $\alpha_\varphi(\overline{y})$.

2. Preliminaries and examples. Given fixed finite index sets I and J, the general optimization problem $\mathcal{P}(f,g,h)$ is defined for any triple of problem data (f,g,h) with $f, g_i, h_j \in C^k(\mathbb{R}^n)$, $i \in I$, $j \in J$, by setting

$$\mathcal{P}(f,g,h) \quad \min\{f(x) \mid x \in M\}, \text{ where} \tag{1}$$

$$M := \{x \in \mathbb{R}^n \mid g_i(x) \le 0, i \in I,\ h_j(x) = 0, j \in J\}. \tag{2}$$

A point $x \in M$ is called *feasible*, and for a feasible point x let $I_0(x) := \{i \in I \mid g_i(x) = 0\}$ denote the index set of *active* inequality constraints.

At a feasible point $x \in M$ the *linear independence constraint qualification* (LICQ) is said to hold if the set of gradients of the equality constraints and the active inequality constraints are linearly independent. We say that the MFCQ holds at $x \in M$ if the gradients of the equality constraints are linearly independent and, moreover, there exists a vector $\xi \in \mathbb{R}^n$ satisfying $Dh_j(x)\xi = 0$, $j \in J$, $Dg_i(x)\xi < 0$, $i \in I_0(x)$. Here, $Dh_j(x)$ stands for the row vector of first partial derivatives of h_j evaluated at x. Besides strong stability, the MFCQ will be the main assumption of this paper.

The *Lagrange polyhedron* $\Delta(x)$, $x \in M$, is defined to be the set of all *Lagrange multiplier* vectors $(\mu, \lambda) \in \mathbb{R}^{|I|+|J|}$ with the property that $D_x L(x,\mu,\lambda) = 0$ and, moreover, $\mu_i \ge 0$ and $\mu_i g_i(x) = 0$, $i \in I$. Here $|I|$ stands for the cardinality of I and L denotes the *Lagrange function*:

$$L(x,\mu,\lambda) := f(x) + \sum_{i \in I} \mu_i g_i(x) + \sum_{j \in J} \lambda_j h_j(x).$$

A feasible point x is called a KKT point for $\mathcal{P}(f,g,h)$ if the Lagrange polyhedron $\Delta(x)$ is nonempty.

Remark 5 (see [1]). At a KKT point x it holds MFCQ if and only if $\Delta(x)$ is compact.

We topologize the space of possible problem data (f,g,h) by means of a nonempty compact set K and an associated seminorm $\|(f,g,h)\|_K$ which is defined to be the maximum modulus of any partial derivative up to order two of f, g_i, h_j at any point of K. (Here the function value itself is considered as the partial derivative of order zero.) Note that the following definition does not depend on the particular choice of K.

Definition 6 (strong stability, cf. [6]). *Let $\overline{x}$ be a KKT point for $\mathcal{P}(\overline{f},\overline{g},\overline{h})$ and let K be a compact neighborhood of $\overline{x}$. Then $\overline{x}$ is called* strongly stable *if there exist a neighborhood V of $(\overline{f},\overline{g},\overline{h})$ (with respect to (w.r.t.) the seminorm associated with K) and a neighborhood $U \subset K$ of $\overline{x}$ such that the following conditions are met:*

(i) *For any $(f, g, h) \in V$ the neighborhood U of $\overline{x}$ contains exactly one KKT point, say, $x(f, g, h)$, of $\mathcal{P}(f, g, h)$.*

(ii) *The KKT mapping $x : V \to U$, defined in* (i)*, is continuous at $(\overline{f}, \overline{g}, \overline{h})$.*

For $\mu \in \mathbb{R}^{|I|}$ let $I_+(\mu) := \{i \in I \mid \mu_i > 0\}$. With this notation we are ready to state Kojima's characterization of strong stability by means of first and second order data of the problem.

Theorem 7. *Let x be a KKT point and let MFCQ at x be satisfied for $\mathcal{P}(f, g, h)$. Then x is strongly stable if and only if the following conditions are satisfied:*

(i) *For all $(\mu, \lambda) \in \Delta(x)$ and all index sets I' with $I_+(\mu) \subset I' \subset I_0(x)$ the restriction of the Hessian matrix $D^2_{xx}L(x, \mu, \lambda)$ to the tangent space $T^{I'}$ is nonsingular and exhibits for all possible choices of (μ, λ) and I' the same number of negative eigenvalues, say,* $\text{ind}_-(x)$*, called the* (negative) index*. The tangent space $T^{I'}$ is defined by*

$$T^{I'} := \{\xi \in \mathbb{R}^n \mid Dg_i \cdot \xi = 0,\ i \in I',\ Dh_j \cdot \xi = 0,\ j \in J\}. \tag{3}$$

(ii) *If LICQ fails to hold (at x), then we have* $\text{ind}_-(x) = 0$*. (Note that x has index zero if and only if $D^2_{xx}L(x, \mu, \lambda)$ is positive definite on $T^{I_+(\mu)}$ for all $(\mu, \lambda) \in \Delta(x)$, i.e., the smaller tangent spaces considered in* (i) *are of no interest.)*

Note that a strongly stable KKT point with index zero is necessarily a local minimizer. The following is an immediate consequence of Theorem 7.

Remark 8. Strong stability is a stable property in the following sense. Let $\overline{x}$ be a strongly stable KKT point for $\mathcal{P}(\overline{f}, \overline{g}, \overline{h})$ with index i such that MFCQ holds. Moreover, let K be a compact neighborhood of $\overline{x}$. Then there exists a neighborhood V of $(\overline{f}, \overline{g}, \overline{h})$ w.r.t. $\|.\|_K$ and a neighborhood $U \subset K$ of $\overline{x}$ such that the corresponding KKT point $x(f, g, h) \in U$ is strongly stable again and also has index i. (MFCQ is a stable property in the aforementioned sense, too.)

Next we discuss some typical examples of critical value functions (in particular marginal functions). In all examples strong stability and MFCQ are satisfied. Thus, in view of Theorem 3, the marginal functions appearing in these examples have finite modulus of concavity.

Example 9. *For $x, y \in \mathbb{R}$ consider $\mathcal{P}(y)$: $\min\{x \mid -x-y-y^2 \leq 0,\ -x+y-y^2 \leq 0\}$. The critical value function φ associated with the minimizer $\overline{x} = 0$ of the problem $\mathcal{P}(0)$ becomes $\varphi(y) = |y| - y^2$. In this case, $\varphi(y) = \max\{y - y^2, -y - y^2\}$, and, in particular, φ is a finite continuous selection of maximum type. Note that φ has modulus of concavity $\alpha_\varphi(0) = 2$.*

Example 10. *For $x, y \in \mathbb{R}$ consider $\mathcal{P}(y)$: $\min\{x^2 \mid -x + y \leq 0\}$. The critical value function φ associated with the minimizer $\overline{x} = 0$ of the problem $\mathcal{P}(0)$ becomes $\varphi(y) = y^2$ $(y \geq 0)$ and $\varphi(y) = 0$ $(y \leq 0)$. Again, φ is a continuous selection, but in contrast to Example* 9*, it is not of maximum type.*

Example 11. *In contrast to Examples* 9 *and* 10*, this example shows that the critical value function φ need not to be a continuous selection of a finite number of C^1-functions. Let $x, y \in \mathbb{R}^2$ and consider*

$$\mathcal{P}(y) : \min\{x_2 \mid g_1(x, y) \leq 0,\ g_2(x, y) \leq 0\},$$

where $g_1(x, y) := x_1^2 - x_2$, $g_2(x, y) := \frac{1}{2}(x_1 - y_1)^2 + y_2 - x_2$. The problems $\mathcal{P}(y)$ are convex and the point $\overline{x} = 0$ is a strongly stable KKT point for $\mathcal{P}(0)$. Put $Y := \{y \in \mathbb{R}^2 \mid y_1 > 0,\ \frac{1}{2}y_1^2 + y_2 > 0\}$. For $y \in Y$ both constraints g_1 and g_2 are active *at the minimizer $x(y)$ of $\mathcal{P}(y)$. In that case we have $x_1(y) = -y_1 + \sqrt{2y_1^2 + 2y_2}$, $x_2(y) = x_1(y)^2$, and, consequently, $\varphi(y) = (-y_1 + \sqrt{2y_1^2 + 2y_2})^2$. Now suppose that in a neighborhood of $\overline{y} = 0$, the function φ is a continuous selection of C^1-functions, say, $\varphi_1, \ldots, \varphi_k$. Then, at least one of the functions φ_i would have to coincide with φ on an open subset $\tilde{Y} \subset Y$ containing the point $\overline{y} = 0$ in its closure. This, however, cannot be true, since the restricted function $\varphi|_{\tilde{Y}}$ (computed above) cannot be extended as a C^1-function through the origin.*

As a final preliminary we need a result on the Hessian of a critical value function and Schur complements; see [4, p. 37]. For a real symmetric matrix M let the *inertia* $\mathrm{in}(M)$ denote the triple consisting of the numbers $\mathrm{ind}_+(M)$, $\mathrm{ind}_-(M)$, $\mathrm{ind}_0(M)$ of its positive, negative, and zero eigenvalues, respectively. Suppose that M has the form

$$\begin{pmatrix} A & B \\ B^\top & C \end{pmatrix}. \tag{4}$$

If A is nonsingular, the matrix $S_A := C - B^\top A^{-1}\mathrm{B}$ is called the *Schur complement* of A in M. It is well known that the matrices M and $\mathrm{diag}(A, S_A)$ are conjugated and, hence, they have the same inertia, i.e.,

$$\mathrm{in}(M) = \mathrm{in}(A) + \mathrm{in}(S_A). \tag{5}$$

Now, suppose that the matrix C in (4) vanishes. Moreover, let A be of dimension n and let B be an (n, m)-matrix of rank k. Then we have (cf. [5])

$$\mathrm{in}(M) = \mathrm{in}(A|_{\ker B^\top}) + (k, k, m - k). \tag{6}$$

In (6) the symbol $A|_{\ker B^\top}$ denotes the restriction of the bilinear form induced by A, to the subspace $\ker B^\top := \{\xi \in \mathbb{R}^n \mid B^\top \xi = 0\}$.

Now, let $\overline{x}$ be a strongly stable KKT point for $\mathcal{P}(\overline{y})$ and suppose that $I_0(\overline{x}, \overline{y}) = \emptyset$, i.e., no inequality constraints are active. In this case MFCQ and LICQ are equivalent; suppose them to be satisfied.

Then the Lagrange function $L(x, y, \lambda) = f(x, y) + \sum_{j \in J} \lambda_j h_j(x, y)$ can be used for an implicit definition of the KKT mapping $x(y)$,

$$\begin{aligned} D_x^\top L(x, y, \lambda) &= 0, \\ D_\lambda^\top L(x, y, \lambda) &= 0, \end{aligned} \tag{7}$$

where the second equation just denotes feasibility, i.e., $h_j(x, y) = 0$, $j \in J$. The functions in (7) have the following derivative w.r.t. (x, λ), where $D_{x\lambda}^2 L = (D_x^\top h_1 | \ldots | D_x^\top h_{|J|})$:

$$\begin{pmatrix} D_{xx}^2 L & D_{x\lambda}^2 L \\ D_{\lambda x}^2 L & 0 \end{pmatrix}. \tag{8}$$

In view of Theorem 7 the restriction of the bilinear form defined by the matrix $D_{xx}^2 L$ to $\ker D_{\lambda x}^2$ is regular, hence (6) implies that the matrix (8) is regular, i.e., the implicit function theorem applies to the system (7). Thus the KKT mapping $x(y)$ and the corresponding $\lambda(y)$ are of class C^1. For the critical value function φ we have $\varphi(y) = f(x(y), y) = L(x(y), y, \lambda(y))$ (since $h_j(x(y), y) \equiv 0$), and therefore

$$D\varphi(y) = D_y L(x(y), y, \lambda(y)). \tag{9}$$

The implicit function theorem also yields explicit formulas for the derivatives. By computing the derivative of (9), substituting the derivatives of $x(y)$, $\lambda(y)$ by the explicit expressions obtained from (7), we obtain the following formula for the Hessian of φ:

$$D^2\varphi = D^2_{yy}L - \left(D^2_{yx}L, D^2_{y\lambda}L\right)\begin{pmatrix} D^2_{xx}L & D^2_{x\lambda}L \\ D^2_{\lambda x}L & 0 \end{pmatrix}^{-1}\begin{pmatrix} D^2_{xy}L \\ D^2_{\lambda y}L \end{pmatrix}, \tag{10}$$

where all partial derivatives are evaluated at $\overline{x}$, $\overline{y}$ and $\overline{\lambda} := \lambda(\overline{y})$. Note that the latter formula for $D^2\varphi$ can be interpreted as the Schur complement of $D^2_{(x,\lambda)(x,\lambda)}L$ in D^2L.

3. Proof of the main result. In this section we prove a quantitative result on the modulus of concavity. This will prove Theorem 3 and, moreover, it provides an upper bound for the modulus of concavity. The latter upper bound is sharp for problem data (f, g, h) in general position.

For $\alpha \in \mathbb{R}$ and $w \in \mathbb{R}^p$ put $f^\alpha_w(x,y) := f(x,y) - w^\top y + \frac{\alpha}{2}\|y-\overline{y}\|^2$ and consider the family of optimization problems $\mathcal{P}^\alpha_w$ in the variable (x,y), also referred to as *unfolded* problems:

$$\mathcal{P}^\alpha_w \quad \min\left\{ \begin{matrix} f^\alpha_w(x,y) \mid (x,y) \in \mathbb{R}^{n+p}, & g_i(x,y) \le 0,\ i \in I, \\ & h_j(x,y) = 0,\ j \in J \end{matrix} \right\}. \tag{11}$$

Put

$$W := \{D_y^\top L(\overline{x}, \overline{y}, \mu, \lambda) \mid (\mu, \lambda) \in \Delta(\overline{x}, \overline{y})\}, \tag{12}$$

where $\Delta(\overline{x}, \overline{y})$ stands for the set of Lagrange multipliers at the KKT point $\overline{x}$ corresponding to the problem $\mathcal{P}(\overline{y})$.

Remark 12. The pair $(\overline{x}, \overline{y})$ is a KKT point for $\mathcal{P}^\alpha_w$ if and only if $\overline{x}$ is a KKT point for $\mathcal{P}(\overline{y})$ and $w \in W$.

DEFINITION 13. *Let $\overline{x}$ be a strongly stable KKT point for $\mathcal{P}(\overline{y})$. Then let $\alpha(\overline{y})$ denote the infimum over all $\alpha \ge 0$ such that for any $w \in W$ the point $(\overline{x}, \overline{y})$ is a strongly stable KKT point for $\mathcal{P}^\alpha_w$ with the additional property:* $\mathrm{ind}_-(\overline{x}, \overline{y}) = \mathrm{ind}_-(\overline{x})$, *again with the convention that* $\inf(\emptyset) = \infty$. *Note that* $\mathrm{ind}_-(\overline{x})$ *is the index of $\overline{x}$ as a KKT point of $\mathcal{P}(\overline{y})$ whereas* $\mathrm{ind}_-(\overline{x}, \overline{y})$ *is the index of the KKT point $(\overline{x}, \overline{y})$ of $\mathcal{P}^\alpha_w$.*

THEOREM 14 (quantitative modulus of concavity result).

(i) *Let $\overline{x}$ be a strongly stable KKT point and let MFCQ at $\overline{x}$ be satisfied for $\mathcal{P}(\overline{y})$. Then the critical value function φ has finite modulus of concavity $\alpha_\varphi(\overline{y})$. The modulus of concavity can be estimated from above by $\alpha_\varphi(\overline{y}) \le \alpha(\overline{y})$.*

(ii) *For a set of problem data (f, g, h), which is open and dense with respect to the Whitney C^2-topology, it even holds $\alpha_\varphi(\overline{y}) = \alpha(\overline{y})$ for any pair $(\overline{x}, \overline{y})$ satisfying the assumptions of (i).*

In the Whitney C^2-topology, a typical base neighborhood of a function $\overline{F}$ consists of all functions F with the property that at any $x \in \mathbb{R}^n$ the function value $F(x)$ and all its partial derivatives up to order 2 at x are $\varepsilon(x)$-close to $\overline{F}(x)$ and its corresponding derivatives. Here, ε is a continuous positive function on $\mathbb{R}^n$. Note that the Whitney C^2-topology on the space of problem data $C^2(\mathbb{R}^n, \mathbb{R}^{1+|I|+|J|})$, considered in Definition 6, is much finer than the topology defined by the seminorm in that definition. This is because the seminorm only compares function values on a (small) compact set. To prevent ambiguities, recall that in Definition 6 we have topologized the space of problem data (f, g, h) of nonparametric problems. Therefore its domain is $\mathbb{R}^n$. In Theorem 14 we topologize the space of problem data of a p-parametric problem, hence its domain is $\mathbb{R}^n \times \mathbb{R}^p$, in contrast.

In view of Definition 13 and Theorem 7, the upper estimate $\alpha(\overline{y})$ for the modulus of concavity $\alpha_\varphi(\overline{y})$ can be computed explicitly by means of first and second order problem data, evaluated at the point $(\overline{x}, \overline{y})$. We emphasize that the sharpness result (ii) therefore implies that for problem data in general position the modulus of concavity can be computed explicitly as well.

In sections 3.2 and 3.3 we will prove assertion (i) of Theorem 14, where section 3.2 deals with the LICQ case and section 3.3 with the local minimizer case. (In particular, in the local minimizer case, LICQ does not need to be satisfied.) Finally, section 3.4 contains the proof of assertion (ii) of the theorem, here dealing with both the LICQ case and the local minimizer case at once. As a preliminary we need the following lemma dealing with the case where no inequality constraints are present. In this case the critical value function φ is a C^2-function, and we may compute its Hessian.

LEMMA 15. *Let $\overline{x}$ be a strongly stable KKT point for $\mathcal{P}(\overline{y})$ with $I = \emptyset$ and let $\operatorname{ind}_-(\overline{x}) = i$. Suppose that MFCQ (thus LICQ) is satisfied at $\overline{x}$. Then the Hessian $D^2\varphi(\overline{y})$ of the critical value function φ at $\overline{y}$ is positive definite if and only if the KKT point $(\overline{x}, \overline{y})$ of $\mathcal{P}^0_w$ is strongly stable with $\operatorname{ind}_-(\overline{x}, \overline{y}) = i$. Here, $w := D_y^\top L(\overline{x}, \overline{y}, \overline{\lambda})$, where $\overline{\lambda}$ is the unique element from $\Delta(\overline{x}, \overline{y})$.*

Proof. From formulas (5), (6), and (10) we obtain

$$\operatorname{in}(D^2L) = \operatorname{in}(D^2\varphi) + \operatorname{in}(D^2_{(x,\lambda)(x,\lambda)}L), \tag{13}$$

$$\operatorname{in}(D^2_{(x,\lambda)(x,\lambda)}L) = \operatorname{in}(D^2_{xx}L|_{\ker D^2_{\lambda x}L}) + (|J|, |J|, 0). \tag{14}$$

In fact, we obtain formula (13) from (5) for $M = D^2L$ and $A = D^2_{(x,\lambda)(x,\lambda)}L$, whereas we obtain formula (14) from (6) for $M = D^2_{(x,\lambda)(x,\lambda)}L$ and $A = D^2_{xx}L$. Since the rows of the matrix $D^2_{\lambda x}L$ are precisely the vectors $D_x h_j$, $j \in J$, it follows from (14) that

$$\operatorname{ind}_-(D^2_{(x,\lambda)(x,\lambda)}L) = i + |J|, \qquad \operatorname{ind}_0(D^2_{(x,\lambda)(x,\lambda)}L) = 0. \tag{15}$$

From (13) and (14) we see that $D^2\varphi$ is positive definite if and only if

$$\operatorname{ind}_-(D^2L) = i + |J|, \qquad \operatorname{ind}_0(D^2L) = 0. \tag{16}$$

Let us write D^2L in the form

$$D^2L = \begin{pmatrix} D^2_{(x,y)(x,y)}L & D^2_{(x,y)\lambda}L \\ D^2_{\lambda(x,y)}L & 0 \end{pmatrix}. \tag{17}$$

Note that the rows of the matrix $D^2_{\lambda(x,y)}L$ are precisely the vectors Dh_j, $j \in J$; hence they are linearly independent, since already $D_x h_j$, $j \in J$, are linearly independent. From (6) and (17) it follows that

$$\operatorname{ind}_-(D^2L) = i^* + |J|, \qquad \operatorname{ind}_0(D^2L) = i_0^*, \tag{18}$$

where i^* and i_0^* stand for the numbers of negative and zero eigenvalues of the restriction of $D^2_{(x,y)(x,y)}L$ to the subspace $\ker(D^2_{\lambda(x,y)}L)$. From (16) and (18) and the characterization of strong stability (Theorem 7) it follows that $D^2\varphi$ is positive definite if and only if $(\overline{x}, \overline{y})$ is a strongly stable KKT point for $\mathcal{P}_w^0$ ($i_0^* = 0$) with $\text{ind}_-(\overline{x}, \overline{y}) = i^* = i$. $\square$

Corollary 16. *Let $\overline{x}$ be a strongly stable KKT point for $\mathcal{P}(\overline{y})$ with $I = \emptyset$. Then $\alpha(\overline{y})$ is finite and we have $\alpha_\varphi(\overline{y}) = \alpha(\overline{y})$.*

Proof. It suffices to show the following equality of intervals: $(\alpha_\varphi(\overline{y}), \infty) = (\alpha(\overline{y}), \infty)$. We prove only the inclusion ($\subset$); the other inclusion can be shown along the same lines. Let $\alpha > \alpha_\varphi(\overline{y})$. Then, by Definition 1, φ^α is convex (in a neighborhood of $\overline{y}$); thus its Hessian $D^2\varphi^\alpha(\overline{y})$ must be positive semidefinite. For the same reason this also holds for smaller values of α still bigger than $\alpha_\varphi(\overline{y})$, i.e., the Hessian of φ^α must in fact be positive definite. By Lemma 15 this implies $\alpha > \alpha(\overline{y})$. $\square$

3.1. Intermezzo. Besides showing that the estimating quantity $\alpha(\overline{y})$ is finite, the major part of the proof is devoted to the verification of the inequality $\alpha_\varphi(\overline{y}) \leq \alpha(\overline{y})$. The latter holds if and only if for any $\alpha > \alpha(\overline{y})$ the function φ^α is convex on a neighborhood of $\overline{y}$. The convexity of φ^α restricted to an open convex neighborhood of $\overline{y}$ will be established by means of the nonemptiness of its subdifferential. Here we say that the vector w belongs to the subdifferential of φ^α at $\overline{y}$ if we have $\varphi^\alpha(y) - \varphi^\alpha(\overline{y}) \geq w^\top(y - \overline{y})$ for all y from the latter neighborhood. Evidently, the latter is equivalent with $\overline{y}$ being a minimizer (on that neighborhood) of the function $y \mapsto \varphi^\alpha(y) - w^\top(y - \overline{y}) =: \varphi_w^\alpha$. In case that $\overline{x}$ is a strongly stable minimizer of $\mathcal{P}(\overline{y})$, then $\overline{y}$ is a local minimizer of φ_w^α if and only if $(\overline{x}, \overline{y})$ is a local minimizer of the blown up problem $\mathcal{P}_w^\alpha$. For this reason the blown-up problem comes into play in the case that $\overline{x}$ is a local minimizer of $\mathcal{P}(\overline{y})$; see section 3.3. If $\overline{x}$ is not a local minimizer of $\mathcal{P}(\overline{y})$, the aforementioned argument cannot be used. In this case, however, LICQ must be fulfilled. Then the function φ^α is a continuous selection of smooth functions $\tilde{\varphi}^\alpha$. The vector w belongs to the subdifferential of φ^α if and only if it belongs to the subdifferential of all the functions $\tilde{\varphi}^\alpha$. Thus here, the argumentation is different from the minimizer case. The LICQ case will be handled in section 3.2.

3.2. The LICQ case. In this section we will prove assertion (i) of Theorem 14 under the additional assumption that LICQ holds for $\mathcal{P}(\overline{y})$ at the point of interest $\overline{x}$. The proof consists of two parts. First we show that $\overline{\alpha} := \alpha(\overline{y})$ is finite, and then we verify the estimating inequality $\alpha_\varphi(\overline{y}) \leq \overline{\alpha}$.

Note that under LICQ the Lagrange polyhedron $\Delta(\overline{x}, \overline{y})$ is a singleton. Let $(\overline{\mu}, \overline{\lambda})$ denote its unique element. Put $I_0 := I_0(\overline{x}, \overline{y})$ and $I_+ := I_+(\overline{\mu})$. For y close to $\overline{y}$ let $x(y)$ denote the corresponding KKT point of $\mathcal{P}(y)$.

Then we have $I_+ \subset I_0(x(y), y) \subset I_0$. In fact, by continuity, any inequality constraint being active at $x(y)$ for $\mathcal{P}(y)$ must be active at $\overline{x}$ for $\mathcal{P}(\overline{y})$. This provides the second inclusion. For the first inclusion let $(\mu(y), \lambda(y))$ denote the (unique) Lagrange multiplier corresponding to the KKT point $x(y)$ for $\mathcal{P}(y)$. Then $(\mu(y), \lambda(y))$ converges for $y \to \overline{y}$, say, to $(\tilde{\mu}, \tilde{\lambda})$. Continuity arguments show that $(\tilde{\mu}, \tilde{\lambda})$ is a Lagrange multiplier corresponding to $\overline{x}$ as a KKT point for $\mathcal{P}(\overline{y})$, hence we have $(\tilde{\mu}, \tilde{\lambda}) = (\overline{\mu}, \overline{\lambda})$. Consequently, we have $I_+ \subset I_+(\tilde{\mu}) \subset I_0(x(y), y)$, where the last inclusion is due to the definition of a KKT point.

For $\widetilde{I}$ such that $I_+ \subset \widetilde{I} \subset I_0$ we consider the parametric optimization problem $\widetilde{\mathcal{P}}(y)$ obtained from $\mathcal{P}(y)$ by deleting the inequality constraints g_i, $i \notin \widetilde{I}$, and turning the remaining inequality constraints g_i, $i \in \widetilde{I}$, into additional equality constraints. To be precise, the constraint set of $\widetilde{\mathcal{P}}(y)$ is given by the constraints $g_i(x, y) = 0$, $i \in \widetilde{I}$, $h_j(x, y) = 0$, $j \in J$. For y close to $\overline{y}$ the point $x(y)$ is a KKT point of the problem $\widetilde{\mathcal{P}}(y)$ for $\widetilde{I} := I_0(x(y), y)$. Obviously, $\overline{x}$ is a KKT point such that LICQ holds for $\widetilde{\mathcal{P}}(\overline{y})$

as well. The strong stability of $\overline{x}$ for $\widetilde{\mathcal{P}}(\overline{y})$ follows from Theorem 7. This gives rise to the (local) critical value function $\widetilde{\varphi}$ associated with the parametric family of problems $\widetilde{\mathcal{P}}(y)$. Since $\varphi(y)$ coincides with at least one of the function values $\widetilde{\varphi}(y)$, $I_+ \subset \widetilde{I} \subset I_0$, we say that φ is a *selection* of the functions $\widetilde{\varphi}$. (Since φ is continuous by Definition 6 and Remark 8, we shall rather call it a "continuous selection" thereof.)

Let $\widetilde{I}$ be as above and consider the problem $\widetilde{\mathcal{P}}$. Let $\widetilde{\alpha}(\overline{y})$ denote the upper bound for $\alpha_{\widetilde{\varphi}}(\overline{y})$ delivered by Corollary 16. Let $\widetilde{\mathcal{P}}_w^\alpha$ denote the problem $\mathcal{P}_w^\alpha$ with $\mathcal{P}$ replaced by $\widetilde{\mathcal{P}}$. Then for $w := D_y L(\overline{x}, \overline{y}, \overline{\mu}, \overline{\lambda})$ the pair $(\overline{x}, \overline{y})$ is a KKT point for $\widetilde{\mathcal{P}}_w^\alpha$. By definition of $\widetilde{\alpha}(\overline{y})$ the index of $(\overline{x}, \overline{y})$ as a KKT point of the blown-up problem $\widetilde{\mathcal{P}}_w^\alpha$ coincides with the index of $\overline{x}$ as a KKT point of $\widetilde{\mathcal{P}}(\overline{y})$ if and only if $\alpha > \widetilde{\alpha}(\overline{y})$. By definition of $\alpha(\overline{y})$ and Theorem 7 this implies that $\alpha > \alpha(\overline{y})$ if and only if $\alpha > \widetilde{\alpha}(\overline{y})$ for all the problems $\widetilde{\mathcal{P}}$, i.e., $\alpha(\overline{y})$ is the *maximum* of all the upper bounds $\widetilde{\alpha}(\overline{y})$ considered above. Hence, $\alpha(\overline{y}) = \overline{\alpha}$ is finite.

It remains to show that $\alpha_\varphi(\overline{y}) \le \overline{\alpha}$. We have to show that (locally) φ^α is a convex function, provided that $\alpha > \overline{\alpha}$. To this end, let $\alpha > \alpha(\overline{y})$. Then we have $\alpha > \widetilde{\alpha}(\overline{y})$ for any of the problems $\widetilde{\mathcal{P}}$. In virtue of Corollary 16, each of the functions $\widetilde{\varphi}^\alpha$ is convex on a neighborhood of $\overline{y}$. The fact that $\Delta(\overline{x}, \overline{y})$ is a singleton implies that all the functions $\widetilde{\varphi}^\alpha$ have the same gradient at $\overline{y}$, and thus the latter gradient belongs to the subdifferential of their selection φ^α, evaluated at $\overline{y}$.

For $\alpha > \alpha(\overline{y})$, continuity arguments ensure that the above argumentation can also be followed if $\overline{y}$ is slightly perturbed, say, to y. This yields a nonempty subdifferential of φ^α at any y sufficiently close to $\overline{y}$, finally yielding its convexity.

3.3. The local minimizer case. In this section we will prove assertion (i) of Theorem 14 under the additional assumption, that $\overline{x}$ is a local minimizer for $\mathcal{P}(\overline{y})$. As in section 3.2, we first show that $\overline{\alpha} := \alpha(\overline{y})$ is finite, and afterward we verify the estimating inequality $\alpha_\varphi(\overline{y}) \le \overline{\alpha}$.

Let $I_0 := I_0(\overline{x}, \overline{y})$, $\Delta := \Delta(\overline{x}, \overline{y})$, and consider the following collection of index sets:

$$\mathcal{I} := \{\widetilde{I} \subset I_0 \mid \exists (\mu, \lambda) \in \Delta : \ I_+(\mu) \subset \widetilde{I}\}. \tag{19}$$

Continuity arguments, as used in section 3.2, show that $\mathcal{I}$ contains all possible active index sets $I_0(x(y), y)$ provided that the parameter y is sufficiently close to $\overline{y}$. The minimal elements of $\mathcal{I}$ (w.r.t. inclusion) are the sets of the form $I_+(\mu)$, such that (μ, λ) is a vertex of the polytope Δ. For $\widetilde{I}$ being a minimal element of $\mathcal{I}$ let $\widetilde{\mathcal{P}}(y)$ denote the parametric optimization problem obtained from $\mathcal{P}(y)$ by deleting all the inequality constraints g_i with i not belonging to $\widetilde{I}$ and turning the remaining inequality constraints g_i, $i \in \widetilde{I}$, into additional equality constraints, in just the same way as in section 3.2. By Theorem 7 the point $\overline{x}$ is a strongly stable local minimizer also for $\widetilde{\mathcal{P}}(\overline{y})$. Considering $\widetilde{\mathcal{P}}$, we are in the situation of Corollary 16. The corresponding local marginal function $\widetilde{\varphi}$ has finite modulus of concavity $\alpha_{\widetilde{\varphi}}$, and let $\widetilde{\alpha} := \widetilde{\alpha}(\overline{y})$ denote the corresponding upper estimate. We claim that $\overline{\alpha}$ is the *maximum* of all the bounds $\widetilde{\alpha}$, where $\widetilde{I}$ ranges over all minimal elements of $\mathcal{I}$.

Let us now prove this claim. By the very construction of $\widetilde{\mathcal{P}}$, it follows from Theorem 7 that $\overline{\alpha} \ge \widetilde{\alpha}$ for any $\widetilde{I}$. To complete the proof of the claim, it suffices to verify that each number α, exceeding all the estimates $\widetilde{\alpha}$, is greater than or equal to $\overline{\alpha}$. To this end let $\alpha > \widetilde{\alpha}$ for any minimal element $\widetilde{I}$ of $\mathcal{I}$. Analogously to (3) we define the tangent space $T^{I'}$ for the unfolded problem $\mathcal{P}_w^\alpha$ at $(\overline{x}, \overline{y})$ by setting

$$T^{I'} := \{\xi \in \mathbb{R}^{n+p} \mid Dg_i \cdot \xi = 0,\ i \in I',\ Dh_j \cdot \xi = 0,\ j \in J\}. \tag{20}$$

From the definition of $\widetilde{\alpha}$ and Theorem 7 we know that for any vertex (μ, λ) of Δ the restriction of the Hessian $D^2_{(x,y)(x,y)} L_w^\alpha(\overline{x}, \overline{y}, \mu, \lambda)$ to the tangent space $T^{I_+(\mu)}$ is positive definite. Here we put $L_w^\alpha(x, y, \mu, \lambda) := L(x, y, \mu, \lambda) - w^\top y + \frac{\alpha}{2}\|y - \overline{y}\|^2$. Any element $(\mu, \lambda) \in \Delta$ can be written as a convex combination of vertices of Δ. Since the Hessian matrix $D^2_{(x,y)(x,y)} L_w^\alpha$ depends linearly on (μ, λ), a representation of $(\mu, \lambda) \in \Delta$ as a convex combination of vertices of Δ directly transfers to a representation of the Hessian $D^2_{(x,y)(x,y)} L_w^\alpha$ evaluated at $(\mu, \lambda) \in \Delta$ as a convex combination of the Hessians of L_w^α evaluated at the vertices. Since the tangent space $T^{I_+(\mu)}$, for $(\mu, \lambda) \in \Delta$, is subspace of all the tangent spaces corresponding to vertices with nonvanishing contribution in the latter convex combination, it follows that $D^2_{(x,y)(x,y)} L_w^\alpha(\overline{x}, \overline{y}, \mu, \lambda)$ is positive definite on $T^{I_+(\mu)}$ for all $(\mu, \lambda) \in \Delta$. By definition of $\overline{\alpha}$ this implies $\overline{\alpha} \leq \alpha$, i.e., the claim is proved.

Now we verify the estimating inequality $\alpha_\varphi(\overline{y}) \leq \overline{\alpha}$. To this end let $\alpha > \overline{\alpha}$. We need show only that the function φ^α is convex on an appropriate small neighborhood of $\overline{y}$. To this end recall the blown-up problem defined in (11). Since $\alpha > \overline{\alpha}$, the point $(\overline{x}, \overline{y})$ is a strongly stable KKT point of $\mathcal{P}_w^\alpha$, $w \in W$, with index $\text{ind}_-(\overline{x}, \overline{y}) = \text{ind}_-(\overline{x}) = 0$; thus it is a local minimizer of $\mathcal{P}_w^\alpha$. This immediately implies that $\overline{y}$ is a local minimizer of the function φ_w^α, i.e., any $w \in W$ belongs to the subdifferential of φ^α, evaluated at $\overline{y}$. We have to prove that the subdifferential of φ^α also is nonempty at points y close to $\overline{y}$. Since $x(y)$ is a KKT point for $\mathcal{P}(y)$, the pair $(x(y), y)$ is a KKT point for $\mathcal{P}_{w(y)}^\alpha$, provided that $w(y) = D_y L(x(y), y, \mu, \lambda)$ for some $(\mu, \lambda) \in \Delta(x(y), y)$. Since $\overline{x}$ is a strongly stable KKT point of $\mathcal{P}(\overline{y})$, $(x(y), y)$ is close to $(\overline{x}, \overline{y})$ for y close to $\overline{y}$. For continuity reasons, the vector w must also be close to W. By Remark 5 and formula (12) the set W is compact. Since $(\overline{x}, \overline{y})$ is a strongly stable local minimizer of $\mathcal{P}_w^\alpha$, $w \in W$, the KKT point $(x(y), y)$ of $\mathcal{P}_{w(y)}^\alpha$ must also be a (strongly stable) local minimizer; see Remark 8. As above we see that $w(y)$ belongs to the subdifferential of φ^α at $\overline{y}$. Hence the subdifferential of φ^α is nonempty on a neighborhood of $\overline{y}$, yielding the local convexity of φ^α.

3.4. The sharpness of the estimating inequality.

This section is devoted to the proof of assertion (ii) of Theorem 14.

Let us first consider the question of which index sets $\widetilde{I}$ may become the active index set $I_0(x(y))$ at the KKT point $x(y)$ of $\mathcal{P}(y)$ after a small perturbation of the parameter $y = \overline{y}$. Recall that such $\widetilde{I}$ necessarily belong to $\mathcal{I}$, where $\mathcal{I}$ is defined in (19). This is also true in the LICQ case. For problem data in general position we have much stronger information; see [2] for details.

THEOREM 17. *For problem data (f, g, h) from an open and dense set (w.r.t. the Whitney C^2-topology) the following holds. Given any strongly stable KKT point $\overline{x}$ of any problem $\mathcal{P}(\overline{y})$ such that MFCQ holds, the following two assertions on the index set $\widetilde{I} \subset I$ are equivalent:*

(i) *There are values of the parameter $\widetilde{y}$ arbitrarily close to $\overline{y}$, such that we have $I_0(x(\widetilde{y}), \widetilde{y}) = I_+(\mu(\widetilde{y})) = \widetilde{I}$, where $\mu(\widetilde{y})$ denotes the unique Lagrange multiplier in $\Delta(x(\widetilde{y}), \widetilde{y})$.*

(ii) *$\widetilde{I} \in \mathcal{I}$ and $|\widetilde{I}| + |J| \leq n$.*

To use a common language for referring to sections 3.2 and 3.3, note that the collection of index sets $\mathcal{I}$ defined in formula (19) in section 3.3 also makes sense in the framework of section 3.2. More important, note that the index sets $\widetilde{I}$ considered in 3.2 belong to the so-defined set $\mathcal{I}$.

In both sections we have found an index set $\widetilde{I} \in \mathcal{I}$ such that $\widetilde{\alpha} = \overline{\alpha}$ and assertion (ii) of Theorem 17 holds (using $\widetilde{I}$ instead of I). Let such an index set $\widetilde{I}$ now be fixed.

In view of Corollary 16 we have $\alpha_{\widetilde{\varphi}}(\overline{y}) = \widetilde{\alpha}$. Since $\widetilde{\mathcal{P}}(\overline{y})$ is an equality constraint problem, Corollary 16 also applies after small parameter changes, i.e., letting $\widetilde{\alpha}(y)$ denote the upper bound for $\alpha_{\widetilde{\varphi}}(y)$ corresponding to $\widetilde{\mathcal{P}}(y)$, we also have $\alpha_{\widetilde{\varphi}}(y) = \widetilde{\alpha}(y)$. According to Theorem 17, $\widetilde{I}$ satisfies assertion (i) of the latter theorem. Note that $I_0(x(\widetilde{y}), \widetilde{y}) = I_+(\mu(\widetilde{y})) = \widetilde{I}$ implies the existence of a small neighborhood $\widetilde{V}$ of $\widetilde{y}$ such that $I_0(x(y), y) = \widetilde{I}$ for all $y \in \widetilde{V}$. Hence, on $\widetilde{V}$, we have $\widetilde{\varphi} = \varphi$. For $y \in \widetilde{V}$ this implies $\alpha_\varphi(y) = \widetilde{\alpha}(y)$. The function $\widetilde{\alpha}$ is continuous and, in view of Definition 1, α_φ is upper semicontinuous. Since y may be chosen arbitrarily close to $\overline{y}$, we deduce $\alpha_\varphi(\overline{y}) \geq \widetilde{\alpha} = \overline{\alpha}$. This is the opposite of the estimating inequality, thus equality holds.

Acknowledgment. The authors are indebted to the associate editor and the anonymous referees for their precise and constructive remarks.

REFERENCES

[1] J. Gauvin, *A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming*, Math. Program., 12 (1977), pp. 136–138.

[2] H. Günzel, *The crease structure of the Karush-Kuhn-Tucker set in parametric optimization*, Math. Oper. Res., 21 (1996), pp. 783–792.

[3] H. Günzel and H. Th. Jongen, *Strong Stability Implies Mangasarian-Fromovitz Constraint Qualification*, SIAM J. Optim., to appear.

[4] H. Th. Jongen, K. Meer, and E. Triesch, *Optimization Theory*, Kluwer Academic Publishers, Boston, 2004.

[5] H. Th. Jongen, T. Möbert, J. Rückmann, and K. Tammer, *On inertia and Schur complements in optimization*, Linear Algebra Appl., 95 (1987), pp. 97–109.

[6] M. Kojima, *Strongly stable stationary solutions in nonlinear programs*, in Analysis and Computation of Fixed Points, S. M. Robinson, ed., Academic Press, New York, 1980, pp. 93–138.

SIAM J. OPTIM.
Vol. 16, No. 4, pp. 1054–1075

A DERIVATIVE-FREE ALGORITHM FOR LINEARLY CONSTRAINED FINITE MINIMAX PROBLEMS*

G. LIUZZI†, S. LUCIDI†, AND M. SCIANDRONE‡

Abstract. In this paper we propose a new derivative-free algorithm for linearly constrained finite minimax problems. Due to the nonsmoothness of this class of problems, standard derivative-free algorithms can locate only points which satisfy weak necessary optimality conditions. In this work we define a new derivative-free algorithm which is globally convergent toward standard stationary points of the finite minimax problem. To this end, we convert the original problem into a smooth one by using a smoothing technique based on the exponential penalty function of Kort and Bertsekas. This technique depends on a smoothing parameter which controls the approximation to the finite minimax problem. The proposed method is based on a sampling of the smooth function along a suitable search direction and on a particular updating rule for the smoothing parameter that depends on the sampling stepsize. Numerical results on a set of standard minimax test problems are reported.

Key words. derivative-free optimization, linearly constrained finite minimax problems, nonsmooth optimization

AMS subject classifications. 90C56, 90C47, 65K05

DOI. 10.1137/040615821

1. Introduction.

Many problems of interest in real world applications can be modeled as finite minimax problems. This class of problems arises, for instance, in the solution of approximation problems, systems of nonlinear equations, nonlinear programming problems, and multiobjective problems. Many algorithms have been developed for the solution of finite minimax problems which require the knowledge of first- or second-order derivatives of the functions involved in the definition of the problem. Unfortunately, in some engineering applications, such as some of those arising in optimal design problems, the function values are obtained by direct measurements (which are often affected by numerical error or random noise) or are the result of complex simulation programs so that first-order derivatives cannot be explicitly calculated or approximated. Moreover, the nonsmoothness of the minimax problem does not allow us to employ some off-the-shelf derivative-free method, since most of these methods are based on a well-established convergence theory which, in order to guarantee convergence to a stationary point, requires first-order derivatives to be continuous, even though they cannot be computed. In particular, if the continuity assumption on the derivatives is relaxed, it is no longer possible to prove global convergence of the derivative-free method to a stationary point, but it is possible only to prove convergence towards a point where the (Clarke) generalized directional derivative is nonnegative with respect to every search direction explored by the algorithm

*Received by the editors September 27, 2004; accepted for publication (in revised form) August 12, 2005; published electronically February 3, 2006. This work was supported by MIUR-FIRB research project RBNE01WBBB "*Large-scale nonlinear optimization*," Rome, Italy.

http://www.siam.org/journals/siopt/16-4/61582.html

†Università di Roma "La Sapienza," Dipartimento di Informatica e Sistemistica "A. Ruberti," Via Buonarroti 12, 00185 Rome, Italy (liuzzi@dis.uniroma1.it, lucidi@dis.uniroma1.it).

‡Istituto di Analisi dei Sistemi ed Informatica del CNR, Viale Manzoni 30, 00185 Rome, Italy (sciandro@iasi.rm.cnr.it).

(see Appendix A for such a general result). Such points can be considered as weak stationary points in the sense that the (Clarke) generalized directional derivative can still be negative along some unexplored direction.

In this paper we consider a particular class of nonsmooth problems, namely, the problem of minimizing the maximum among a finite number of smooth functions. We recall that, for such a class of problems, the (Clarke) generalized directional derivative is proved to coincide with the directional derivative, but, also in this case, classical derivative-free codes can still be convergent toward weak stationary points (see [8] for a thorough discussion on this topic). Finite minimax problems have the valuable feature that they can be approximated by a smooth problem. This smooth approximation of the minimax problem can be achieved by using different techniques (see [19], [1], [2], [4], [7], [15], [17], [18], and [20]). In particular, we consider an approximation approach based on a so-called smoothing function which depends on a precision parameter (see [3], [16], and [11]). In order to define a solution method based on a smoothing technique, two different aspects, one computational and the other theoretical, must be considered. From a computational point of view, a trade-off should be found between the accuracy of the approximation and the problem of limiting the ill-conditioning due to the nonsmoothness of the minimax problem at the solutions. From a theoretical point of view, the algorithm should be guaranteed to converge a stationary point of the original minimax problem. In particular, a class of algorithms [16] for the solution of the minimax problem has been proposed, which takes into account the above two requirements. This is accomplished by using a feedback precision-adjustment rule which updates the precision parameter during the optimization process of the smoothing function. Roughly speaking, the idea behind the proposed updating rule is that of updating the parameter only when the minimization method has carried out a significant improvement. However, these updating rules are based upon the knowledge of the first derivatives of the problem.

In this paper we propose a derivative-free method which is based on a sampling of the smooth function along suitable search directions and on a particular updating rule for the smoothing parameter that depends on the sampling stepsize. We manage to prove convergence of the method to a stationary point of the minimax problem, while reducing the negative effects of the ill-conditioning that the smoothing approach incurs.

In section 2, we describe the minimax problem, its properties, and the smoothing function. In section 3, we report some convergence results for a general derivative-free approach to solve the minimax problem. In section 4, we report the proposed derivative-free algorithm and its convergence analysis. Finally, section 5 is devoted to some results of our method.

2. Problem, definition, and smooth approximation. In this paper we consider the solution of finite minimax problems where the variables are subject to linear inequality constraints. In particular, we consider problems of the form

$$\begin{aligned} &\min && f(x) \\ &s.t. && Ax \le b, \end{aligned} \tag{1}$$

where $x \in \Re^n$, $b \in \Re^m$, $A \in \Re^{m \times n}$, and

$$f(x) = \max_{1 \le i \le q} f_i(x).$$

We indicate by $\mathcal{F}$ the *feasible set* of problem (1), namely,

$$\mathcal{F} = \{x \in \Re^n : Ax \leq b\}.$$

We require the following standard assumption to hold, which ensures that the level sets of $f(x)$ are compact.

Assumption 1. *The functions $f_i : \Re^n \to \Re$, $i = 1, \dots, q$, are twice continuously differentiable functions on $\Re^n$, and the function $f(x)$ is radially unbounded on the feasible set $\mathcal{F}$; that is, for every sequence $\{x_k\} \subset \mathcal{F}$ satisfying* $\lim_{k\to\infty} \|x_k\| = +\infty$,

$$\lim_{k\to\infty} f(x_k) = +\infty.$$

Note that, even though every function $f_i(x)$, $i = 1, \dots, q$, is twice continuously differentiable, we assume that their gradients can be neither calculated nor approximated explicitly.

We denote by $B(x)$ the following set of indices:

$$B(x) = \{i = 1, \dots, q : f_i(x) = f(x)\}. \tag{2}$$

For every feasible point $x \in \mathcal{F}$, we define the *set of indices of active constraints* by

$$I(x) = \{j = 1, \dots, m : a_j^T x = b_i\}, \tag{3}$$

and the *cone of feasible directions*

$$T(x) = \{d \in \Re^n : a_j^T d \leq 0,\ j \in I(x)\}, \tag{4}$$

where a_j^T, $j = 1, \dots, m$, denotes the jth row of the constraints matrix A. The directional derivative of the max function f at x in the direction $d \in \Re^n$ is given by (see, e.g., [3])

$$Df(x, d) = \max_{i \in B(x)} \{\nabla f_i(x)^T d\}.$$

We define $\bar{x} \in \mathcal{F}$ a stationary point of problem (1) if

$$Df(\bar{x}, d) \geq 0 \quad \forall d \in T(\bar{x}). \tag{5}$$

In particular, the following proposition shows a different characterization of the stationary points of problem (1).

Proposition 1. *A point $\hat{x} \in \mathcal{F}$ is a stationary point of problem* (1) *if and only if there exist $\lambda_i \geq 0$, $i \in B(\hat{x})$, such that*

$$\sum_{i \in B(\hat{x})} \lambda_i = 1, \tag{6}$$

$$\left(\sum_{i \in B(\hat{x})} \lambda_i \nabla f_i(\hat{x}) \right)^T d \geq 0 \quad \forall d \in T(\hat{x}). \tag{7}$$

Proof. If $\hat{x} \in \mathcal{F}$ is a stationary point of problem (1), then there exists at least one index $j \in B(\hat{x})$ such that $\nabla f_j(\hat{x})^T d \geq 0$. Then conditions (6) and (7) hold with $\lambda_j = 1$ and $\lambda_i = 0$ for all $i \neq j$.

If $\hat{x} \in \mathcal{F}$ satisfies conditions (6) and (7), then for any $d \in T(\hat{x})$, we can write

$$0 \leq \left(\sum_{i \in B(\hat{x})} \lambda_i \nabla f_i(\hat{x}) \right)^T d \leq \max_{i \in B(\hat{x})} \nabla f_i(\hat{x})^T d,$$

which shows that $\bar{x}$ is a stationary point of problem (1). □

In order to find a stationary point of problem (1) we adopt a smoothing technique [3], [11], [16], [19] which consists of solving a sequence of smooth problems approximating the minimax problem in the limit. Let $\mu > 0$ be a smoothing parameter and define

$$f(x, \mu) = \mu \ln \sum_{i=1}^{q} \exp\left(\frac{f_i(x)}{\mu} \right),$$

which is sometimes referred to as an exponential penalty function [3]. An alternative expression for $f(x, \mu)$ is given by

$$f(x, \mu) = f(x) + \mu \ln \sum_{i=1}^{q} \exp\left(\frac{f_i(x) - f(x)}{\mu} \right).$$

We report some properties of $f(x, \mu)$ [19].

Proposition 2. *Suppose $f_i(x)$, $i = 1, \ldots, q$, are twice continuously differentiable functions. Then*

(i) *$f(x, \mu)$ is increasing with respect to μ, and*

$$f(x) \leq f(x, \mu) \leq f(x) + \mu \ln q; \tag{8}$$

(ii) *$f(x, \mu)$ is twice continuously differentiable for all $\mu > 0$, and*

$$\nabla_x f(x, \mu) = \sum_{i=1}^{q} \lambda_i(x, \mu) \nabla f_i(x), \tag{9}$$

$$\begin{aligned} \nabla_x^2 f(x, \mu) = {} & \sum_{i=1}^{q} \left(\lambda_i(x, \mu) \nabla^2 f_i(x) + \frac{1}{\mu} \lambda_i(x, \mu) \nabla f_i(x) \nabla f_i(x)^T \right) \\ & - \frac{1}{\mu} \left(\sum_{i=1}^{q} \lambda_i(x, \mu) \nabla f_i(x) \right) \left(\sum_{i=1}^{q} \lambda_i(x, \mu) \nabla f_i(x) \right)^T, \end{aligned} \tag{10}$$

where

$$\lambda_i(x, \mu) = \frac{\exp(f_i(x)/\mu)}{\sum_{j=1}^{q} \exp(f_j(x)/\mu)} \in (0, 1), \qquad \sum_{i=1}^{q} \lambda_i(x, \mu) = 1. \tag{11}$$

3. Derivative-free convergence conditions.

A derivative-free algorithm for problem (1) must account for two difficulties. The first is the nonsmoothness of problem (1). The second is that stationary points of problem (1), as stated by (5) and Proposition 1, are characterized by first-order derivatives of the component functions $f_i(x)$, $i = 1, \ldots, q$, which are not available.

In order to treat the nonsmoothness of problem (1), we employ the smooth approximating problem,

$$\min_{x \in \mathcal{F}} f(x, \mu), \tag{12}$$

where the approximating parameter μ will be adaptively reduced during the optimization process.

In order to tackle the second difficulty of unavailable first derivatives, we try to obtain first-order information by sampling the objective function along a suitable set of search directions. Specifically, we follow the approach proposed in [13], which uses a set of search directions that positively span an "ϵ-approximation" of the cone of feasible directions or, in other words, the cone of feasible directions with respect to the ϵ-active constraints.

Formally, for any $\epsilon > 0$ and $x \in \mathcal{F}$, we define the set of indices of ϵ-active constraints by

$$I(x;\epsilon) = \{j : a_j^T x \geq b_j - \epsilon\}$$

and the ϵ-approximation of the cone of feasible directions by

$$T(x;\epsilon) = \{d \in \Re^n : a_j^T d \leq 0 \quad \forall j \in I(x;\epsilon)\}.$$

The following proposition (see [13]) describes some properties of sets $I(x;\epsilon)$ and $T(x;\epsilon)$.

PROPOSITION 3. *Let $\{x_k\}$ be a sequence of iterates converging towards a point $\bar{x} \in \mathcal{F}$. Then there exists a value $\epsilon^* > 0$ (depending on $\bar{x}$ only) such that for every $\epsilon \in (0, \epsilon^*]$ there exists $\bar{k}_\epsilon$ such that*

$$I(x_k;\epsilon) = I(\bar{x}), \tag{13}$$

$$T(x_k;\epsilon) = T(\bar{x}) \tag{14}$$

for all $k \geq \bar{k}_\epsilon$.

Proof. See the proof of Proposition 1 in [13]. □

The first step toward defining a derivative-free method for the solution of problem (12) is to associate a suitable set of search directions with each point x_k produced by the algorithm. This set should have the property that the local behavior of the objective function in each direction in the set provides sufficient information to overcome the lack of the gradient. Formally, we introduce the following assumption.

ASSUMPTION 2. *Let $\{x_k\}$ be a sequence of feasible points and $\{D_k\}$ be a sequence of sets of search directions. Then, for all k,*

$$D_k = \{d_k^i : \|d_k^i\| = 1, \quad i = 1, \ldots, r_k\},$$

and, for some constant $\bar{\epsilon} > 0$,

$$cone\{D_k \cap T(x_k;\epsilon)\} = T(x_k;\epsilon) \quad \forall \epsilon \in [0, \bar{\epsilon}].$$

Moreover, $\bigcup_{k=0}^{\infty} D_k$ is a finite set and r_k is bounded.

The proposition that follows states a general convergence result. In particular, it identifies sufficient conditions on the sampling of the smoothing function along the directions d_k^i, $i = 1, \ldots, r_k$, and on the updating of the smoothing parameter, which will guarantee global convergence of the method to a stationary point of the original minimax problem (1).

PROPOSITION 4. *Let $\{x_k\}$ be a sequence of feasible points and $\bar{x}$ be a limit point of a subsequence $\{x_k\}_K$ for some infinite set $K \subseteq \{0, 1, \ldots\}$. Let $\{D_k\}$, with $D_k = \{d_k^1, \ldots, d_k^{r_k}\}$, be a sequence of sets of directions which satisfy Assumption 2 and $J_k = \{i \in \{1, \ldots, r_k\} : d_k^i \in T(x_k, \epsilon)\}$ with $\epsilon \in (0, \min\{\bar{\epsilon}, \epsilon^\star\}]$, where $\bar{\epsilon}$ and $\epsilon^\star$ are defined in Assumption 2 and Proposition 3, respectively.*

Suppose that the following conditions hold:

(i) *for each $k \in K$ and $i \in J_k$, there exist y_k^i and scalars $\xi_k^i > 0$ such that*

$$y_k^i + \xi_k^i d_k^i \in \mathcal{F}; \tag{15}$$

$$f(y_k^i + \xi_k^i d_k^i, \mu_k) \geq f(y_k^i, \mu_k) - o(\xi_k^i); \tag{16}$$

(ii) *furthermore,*

$$\lim_{k\to\infty, k\in K} \mu_k = 0; \tag{17}$$

$$\lim_{k\to\infty, k\in K} \frac{\max_{i\in J_k}\{\xi_k^i, \|x_k - y_k^i\|\}}{\mu_k} = 0. \tag{18}$$

Then $\bar{x}$ is a stationary point of the minimax problem (1).

Proof. By applying the mean-value theorem to (16), we can write

$$-o(\xi_k^i) \leq f(y_k^i + \xi_k^i d_k^i, \mu_k) - f(y_k^i, \mu_k) = \xi_k^i \nabla_x f(u_k^i, \mu_k)^T d_k^i, \qquad i \in J_k, \tag{19}$$

where $u_k^i = y_k^i + t_k^i \xi_k^i d_k^i$, with $t_k^i \in (0,1)$. By using the mean-value theorem again and the Cauchy–Schwarz inequality, we can write

$$\begin{aligned}\xi_k^i \nabla_x f(u_k^i, \mu_k)^T d_k^i &= \xi_k^i \nabla_x f(x_k, \mu_k)^T d_k^i + \xi_k^i (u_k^i - x_k)^T \nabla_x^2 f(\tilde{u}_k^i, \mu_k) d_k^i \\ &\leq \xi_k^i \nabla_x f(x_k, \mu_k)^T d_k^i + \xi_k^i \|u_k^i - x_k\| \|\nabla_x^2 f(\tilde{u}_k^i, \mu_k) d_k^i\|,\end{aligned}$$

where $\tilde{u}_k^i = x_k + \tilde{t}_k^i (u_k^i - x_k)$, with $\tilde{t}_k^i \in (0,1)$. By considering expression (10) of $\nabla_x^2 f(\tilde{u}_k^i, \mu_k)$ and the triangle inequality, we get that

$$\begin{aligned}&\xi_k^i \nabla_x f(u_k^i, \mu_k)^T d_k^i \leq \xi_k^i \nabla_x f(x_k, \mu_k)^T d_k^i \\ &+ \xi_k^i \|u_k^i - x_k\| \left\{ \left\| \sum_{j=1}^q \lambda_j(\tilde{u}_k^i, \mu_k) \nabla^2 f_j(u_k^i) d_k^i \right\| + \frac{1}{\mu_k} \left\| \sum_{j=1}^q \lambda_j(\tilde{u}_k^i, \mu_k) \nabla f_j(\tilde{u}_k^i) \nabla f_j(\tilde{u}_k^i)^T d_k^i \right.\right. \\ &- \left.\left. \left(\sum_{j=1}^q \lambda_j(\tilde{u}_k^i, \mu_k) \nabla f_j(\tilde{u}_k^i) \right) \left(\sum_{j=1}^q \lambda_j(\tilde{u}_k^i, \mu_k) \nabla f_j(\tilde{u}_k^i) \right)^T d_k^i \right\| \right\}.\end{aligned}$$

Since $\{x_k\}_K$ converges, it follows from Assumption 2 and (11) that, for all i and j, $\{x_k\}_K$, $\{\tilde{u}_k^i\}$, $\{\lambda_j(\tilde{u}_k^i, \mu_k)\}$, and $\{d_k^i\}$ are bounded sequences. Therefore, by Assumption 1, we can find constants c_1 and c_2 such that

$$-o(\xi_k^i) \leq \xi_k^i \nabla_x f(u_k^i, \mu_k)^T d_k^i \leq \xi_k^i \nabla_x f(x_k, \mu_k)^T d_k^i + \xi_k^i \|u_k^i - x_k\| \left(c_1 + \frac{1}{\mu_k} c_2 \right). \tag{20}$$

From (20) and (9), we obtain

$$\left(\sum_{j=1}^q \lambda_j(x_k, \mu_k) \nabla f_j(x_k) \right)^T d_k^i + \left(c_1 + \frac{1}{\mu_k} c_2 \right) \|u_k^i - x_k\| \geq -\frac{o(\xi_k^i)}{\xi_k^i}. \tag{21}$$

Since $\cup_{k\in K} D_k$ is a finite set by Assumption 2 and recalling the boundedness of each sequence $\{\lambda_j(x_k, \mu_k)\}$, $j=1,\dots,q$, there exist an infinite set $\bar K \subseteq K$ and, given the fact that r_k is bounded, a finite set $J \subseteq \{1,2,\dots\}$ and $\bar d^j \in \Re^n$, $j \in J$, such that

$$\lim_{\substack{k\to\infty\\ k\in\bar K}} x_k = \bar x, \tag{22}$$

$$\lim_{\substack{k\to\infty\\ k\in\bar K}} \lambda_j(x_k,\mu_k) = \bar\lambda_j, \qquad j=1,\dots,q, \tag{23}$$

$$J_k = J \quad \forall k \in \bar K, \tag{24}$$

$$d_k^j = \bar d^j \quad \forall j\in J \text{ and } \ k\in\bar K. \tag{25}$$

Moreover, recalling that $u_k^i = y_k^i + t_k^i \xi_k^i d_k^i$, with $t_k^i \in (0,1)$, we have that

$$\left(c_1 + \frac{1}{\mu_k}c_2\right)\|u_k^j - x_k\| \le \left(c_1 + \frac{1}{\mu_k}c_2\right)(\|y_k^j - x_k\| + \xi_k^j) \quad \forall j \in J,$$

which, by using (18), implies that

$$\lim_{\substack{k\to\infty\\ k\in\bar K}} \left(c_1 + \frac{1}{\mu_k}c_2\right)\|u_k^j - x_k\| = 0 \quad \forall j\in J. \tag{26}$$

We note that expression (11) can be rewritten as

$$\lambda_j(x,\mu) = \frac{\exp((f_j(x)-f(x))/\mu)}{\sum_{l=1}^q \exp((f_l(x)-f(x))/\mu)}, \qquad j=1,\dots,q,$$

so that it is easily seen that

$$\begin{aligned} \bar\lambda_j &\ge 0 \quad \forall j,\\ \bar\lambda_j &= 0 \quad \forall j \notin B(\bar x). \end{aligned} \tag{27}$$

Furthermore, since $\sum_{j=1}^q \lambda_j(x_k,\mu_k) = 1$ for all k, then

$$\sum_{j=1}^q \bar\lambda_j = 1. \tag{28}$$

Now, recalling (26) and taking limits in (21) as $k\to\infty$, $k\in\bar K$, we obtain

$$\left(\sum_{j=1}^q \bar\lambda_j \nabla f_j(\bar x)\right)^T \bar d^i \ge 0 \quad \forall i \in J. \tag{29}$$

Now, Proposition 3 and Assumption 2 imply that, for $k\in K$,

$$T(\bar x) = T(x_k;\epsilon) = cone\{D_k \cap T(x_k;\epsilon)\} = cone\{d_k^i\}_{i\in J_k}. \tag{30}$$

Hence, by (30), (24), and (25) we have that

$$T(\bar x) = cone\{\bar d^i\}_{i\in J}, \tag{31}$$

so that, for every $d \in T(\bar{x})$, there exist $\beta_i \geq 0$, for all $i \in J$, such that

$$d = \sum_{i \in J} \beta_i \bar{d}^i. \tag{32}$$

Thus, we obtain from (29) and (32) that, for every $d \in T(\bar{x})$,

$$\left(\sum_{j=1}^{q} \bar{\lambda}_j \nabla f_j(\bar{x}) \right)^T d = \sum_{i \in J} \beta_i \left(\sum_{j=1}^{q} \bar{\lambda}_j \nabla f_j(\bar{x}) \right)^T \bar{d}^i \geq 0,$$

which, along with (27) and (28), proves the proposition (see Proposition 1). □

The above proposition is a nontrivial extension of similar results established in the context of derivative-free methods for smooth optimization (see, for instance, [13]). The major novelty of Proposition 4 is (18), which relates the convergence rate of the smoothing parameter with that of the sampling stepsizes. Indeed, Proposition 4 has two crucial aspects:

1. When $x_k \to \bar{x}$ and $\mu_k \to 0$, eventually,

$$\nabla_x f(x_k, \mu_k)^T d_k^i = \left(\sum_{j=1}^{q} \lambda_j(x_k, \mu_k) \nabla f_j(x_k) \right)^T d_k^i \geq 0 \quad \forall\, i \in J_k.$$

2. The bounded sequence $\{(\lambda_1(x_k, \mu_k), \ldots, \lambda_q(x_k, \mu_k))\}$ has an accumulation point. This allows us to overcome the difficulty tied to the indefiniteness of $\nabla_x^2 f(x_k, \mu_k)$ in the limit.

The sampling of the smooth objective function along the directions d_k^i, $i \in J_k$, introduces a further difficulty, namely, that $\nabla_x f(x_k, \mu_k)^T d_k^i$ is approximated by the quantity

$$\nabla_x f(u_k^i, \mu_k)^T d_k^i = \left(\sum_{j=1}^{q} \lambda_j(u_k^i, \mu_k) \nabla f_j(u_k^i) \right)^T d_k^i,$$

where, for every index $i \in J_k$, we have different bounded sequences $\{(\lambda_1(u_k^i, \mu_k), \ldots, \lambda_q(u_k^i, \mu_k))\}$. This raises the problem that each of these sequences converges to its own limit while the optimality condition (29) requires them to have the same limit point. In order to guarantee the existence of a unique limit point of the sequences $\{(\lambda_1(u_k^i, \mu_k), \ldots, \lambda_q(u_k^i, \mu_k))\}$, for all $i \in J_k$, it is necessary that $\|u_k^i - x_k\|$, $i \in J_k$, tends to zero faster than μ_k, where $\|u_k^i - x_k\|$ can be viewed as a measure of the degree of approximation of first-order derivatives and μ_k gives a measure of the degree of approximation of the original minimax problem.

To conclude, we note that, since Proposition 4 poses only an upper bound on the convergence rate of μ_k towards zero, it allows us to choose an updating rule for the smoothing parameter which conciliates global convergence with the problem of avoiding the ill-conditioning of the smooth approximating problem.

4. A derivative-free method and global convergence result.

In this section we define an algorithm for the solution of problem (1). The proposed method stems from the union of a derivative-free approach for smooth and linearly constrained optimization with a suitable handling of the smoothing parameter μ. In particular, the derivative-free method samples the smoothing function value along a finite set of search directions and decreases the sampling stepsize and the smoothing parameter if a sufficiently improved objective function value is not attained. The sampling strategy and the updating rule for the smoothing parameter are guided by the convergence conditions of Proposition 4. The derivative-free technique for sampling the smoothing function is based on the *feasible descent method* 2 proposed in [13] for a class of smooth optimization problems, including those with linear constraints. The formal description of the algorithm is reported below.

```
                         Algorithm DF
Data. $x_0 \in \mathcal{F}$, $init\_step_0 > 0$, $\mu_0 > 0$, $\gamma > 0$, $\theta \in (0,1)$, $\bar{\epsilon} > 0$.
Step 0. Set $k = 0$.
Step 1. (Computation of search directions)
          Choose a set of directions $D_k = \{d_k^1, \dots, d_k^{r_k}\}$ satisfying Assumption 2.
Step 2. (Minimization on the cone$\{D_k\}$)
    Step 2.1. (Initialization)
              Set $i = 1$, $y_k^i = x_k$, $\tilde{\alpha}_k^i = init\_step_k$.
    Step 2.2. (Computation of the initial stepsize)
              Compute the maximum steplength $\bar{\alpha}_k^i$ such that $y_k^i + \bar{\alpha}_k^i d_k^i \in \mathcal{F}$
              and set $\hat{\alpha}_k^i = \min\{\bar{\alpha}_k^i, \tilde{\alpha}_k^i\}$.
    Step 2.3. (Test on the search direction)
              If $\hat{\alpha}_k^i > 0$ and $f(y_k^i + \hat{\alpha}_k^i d_k^i, \mu_k) \le f(y_k^i, \mu_k) - \gamma(\hat{\alpha}_k^i)^2$,
                compute $\alpha_k^i$ by the Expansion Step($\bar{\alpha}_k^i, \hat{\alpha}_k^i, y_k^i, d_k^i; \alpha_k^i$)
                and set $\tilde{\alpha}_k^{i+1} = \alpha_k^i$;
              otherwise set $\alpha_k^i = 0$ and $\tilde{\alpha}_k^{i+1} = \theta\tilde{\alpha}_k^i$.
    Step 2.4. (New point)
              Set $y_k^{i+1} = y_k^i + \alpha_k^i d_k^i$.
    Step 2.5 (Test on the minimization on the cone$\{D_k\}$)
              If $i = r_k$, go to Step 3;
              otherwise set $i = i + 1$ and go to Step 2.2.
Step 3. (Main iteration)
          Find $x_{k+1} \in \mathcal{F}$ such that $f(x_{k+1}, \mu_k) \le f(y_k^{i+1}, \mu_k)$;
          otherwise, set $x_{k+1} = y_k^{i+1}$.
          Set $init\_step_{k+1} = \tilde{\alpha}_k^{i+1}$,
          choose $\mu_{k+1} = \min\left\{\mu_k, \max_{i=1,\dots,r_k} \{(\tilde{\alpha}_k^i)^{1/2}, (\alpha_k^i)^{1/2}\}\right\}$,
          set $k = k + 1$, and go to Step 1.
```

```
          Expansion Step ($\bar{\alpha}_k^i, \hat{\alpha}_k^i, y_k^i, d_k^i; \alpha_k^i$).
Data. $\gamma > 0$, $\delta \in (0,1)$.
Step 1. Set $\alpha = \hat{\alpha}_k^i$.
Step 2. Let $\tilde{\alpha} = \min\{\bar{\alpha}_k^i, (\alpha/\delta)\}$.
Step 3. If $\alpha = \bar{\alpha}_k^i$ or $f(y_k^i + \tilde{\alpha} d_k^i, \mu_k) > f(y_k^i, \mu_k) - \gamma\tilde{\alpha}^2$, set $\alpha_k^i = \alpha$ and return.
Step 4. Set $\alpha = \tilde{\alpha}$ and go to Step 2.
```

At Step 1 a suitable set of search directions $d_k^1, \dots, d_k^{r_k}$ is determined. At Step 2 the behavior of the objective function is evaluated along each search direction. In particular, if the search direction is feasible and is of sufficient decrease, the behavior of the objective function along this direction is further investigated by executing an *Expansion Step* until a suitable decrease is no longer obtained or the trial point reaches the boundary of the feasible region.

We indicate by $init\_step_k$ the initial stepsize at iteration k, and, for every direction d_k^i, with $i = 1, \dots, r_k$, we denote

- by $\tilde{\alpha}_k^i$ the candidate initial stepsize;
- by $\bar{\alpha}_k^i$ the maximum feasible stepsize;
- by $\hat{\alpha}_k^i$ the initial stepsize;
- by α_k^i the stepsize actually taken.

At Step 3 the new point x_{k+1} can be the point y_k^{i+1} produced by Steps 1–2 or any point where the objective function is improved with respect to $f(y_k^{r_k}, \mu_k)$. This fact allows us to adopt any approximation scheme for the objective function to produce a new better point. This flexibility can be particularly useful when the evaluation of objective function is computationally expensive, in which case the objective function values produced in previous iterations can be used to build an inexpensive model of $f(x)$ to be minimized with the aim of producing a potentially good point x_{k+1}. However, we note that this option can be discarded simply by setting $x_{k+1} = y_k^{i+1}$.

Then the smoothing parameter μ_k is reduced whenever $\max_{i=1,\dots,r_k}\{(\tilde{\alpha}_k^i)^{1/2}, (\alpha_k^i)^{1/2}\}$ gets smaller than the current smoothing value μ_k. We recall that $\max_{i=1,\dots,r_k}\{(\tilde{\alpha}_k^i)^{1/2}, (\alpha_k^i)^{1/2}\}$ can be viewed as a stationarity measure of the current iterate (see [9], for example). Thus, according to the updating rule, the smoothing parameter is reduced whenever a more precise approximation of a stationary point of the smoothing function is obtained.

The following proposition describes some key properties of certain sequences generated by Algorithm DF.

PROPOSITION 5. *Let $\{x_k\}$, $\{\mu_k\}$ be the sequences generated by Algorithm DF. Then*

(i) *$\{x_k\}$ is well-defined;*

(ii) *the sequence $\{f(x_k, \mu_k)\}$ is monotonically nonincreasing;*

(iii) *the sequence $\{x_k\}$ is bounded;*

(iv) *every cluster point of $\{x_k\}$ belongs to $\mathcal{F}$;*

(v) *the sequences $\{f(x_{k+1}, \mu_k)\}$ and $\{f(x_k, \mu_k)\}$ are both convergent and have the same limit.*

Proof. To prove assertion (i), it suffices to show that the Expansion Step, when performed along a direction d_k^i from y_k^i, for $i \in \{1, \dots, r_k\}$, terminates in a finite number $\bar{\jmath}$ of steps either because $\delta^{-\bar{\jmath}}\hat{\alpha}_k^i \geq \bar{\alpha}_k^i$ or because $f(y_k^i + \delta^{-\bar{\jmath}}\hat{\alpha}_k^i d_k^i, \mu_k) > f(y_k^i, \mu_k) - \gamma\big(\delta^{-\bar{\jmath}}\hat{\alpha}_k^i\big)^2$.

If this were not true, we would have for some k and $i \in \{1, \dots, r_k\}$ that $\hat{\alpha}_k^i > 0$ and

$$\delta^{-j}\hat{\alpha}_k^i < \bar{\alpha}_k^i, \qquad y_k^i + \delta^{-j}\hat{\alpha}_k^i d_k^i \in \mathcal{F},$$
$$f(y_k^i + \delta^{-j}\hat{\alpha}_k^i d_k^i, \mu_k) \leq f(y_k^i, \mu_k) - \gamma\big(\delta^{-j}\hat{\alpha}_k^i\big)^2$$

for all $j = 0, 1, \dots$. But by (i) of Proposition 2,

$$f(y_k^i + \delta^{-j}\hat{\alpha}_k^i d_k^i) \leq f(y_k^i + \delta^{-j}\hat{\alpha}_k^i d_k^i, \mu_k) \leq f(y_k^i, \mu_k) - \gamma\big(\delta^{-j}\hat{\alpha}_k^i\big)^2,$$

for all $j = 0, 1, \dots$, which, since δ^{-j} is unbounded, violates Assumption 1.

To prove assertion (ii), we note that the instructions of the algorithm imply that

$$f(x_{k+1}, \mu_k) \leq f(x_k, \mu_k).$$

Since $\mu_{k+1} \leq \mu_k$ and $f(x, \mu)$ is increasing with respect to μ (see (i) of Proposition 2), we have

$$f(x_{k+1}, \mu_{k+1}) \leq f(x_{k+1}, \mu_k) \leq f(x_k, \mu_k), \tag{33}$$

so that assertion (ii) is proved.

By assertion (ii) we have for all k that $f(x_k, \mu_k) \leq f(x_0, \mu_0)$, and hence

$$x_k \in \{x \;|f(x, \mu_k) \leq f(x_0, \mu_0)\}.$$

Then for any x satisfying

$$f(x, \mu_k) \leq f(x_0, \mu_0)$$

we have from (i) of Proposition 2 that

$$f(x) \leq f(x_0, \mu_0).$$

Thus we can write

$$x_k \in \{x \;|f(x, \mu_k) \leq f(x_0, \mu_0)\} \subseteq \{x \;|f(x) \leq f(x_0, \mu_0)\}.$$

It follows from Assumption 1 that the set $\{x \;|f(x) \leq f(x_0, \mu_0)\}$ is bounded, which proves assertion (iii).

To prove assertion (iv), we note that the instructions of Algorithm DF imply that $x_k \in \mathcal{F}$ for all k. Since $\mathcal{F}$ is a closed set, the assertion follows.

To prove point (v), we note that, by Assumption 1, $f(x)$ is bounded from below on the feasible set $\mathcal{F}$. Therefore, by recalling (8), we have that $\{f(x_k, \mu_k)\}$ is also bounded below, and hence, by point (ii), convergent. From (33), we have that $\{f(x_{k+1}, \mu_k)\}$ converges to the same limit of $\{f(x_k, \mu_k)\}$, which proves assertion (v). □

The proposition that follows establishes some results concerning the adopted sampling technique. In particular, point (i) guarantees that the sampling points tend to cluster more and more. Point (ii) ensures the existence of sufficiently large stepsizes providing feasible points along the search directions.

PROPOSITION 6. *Let $\{x_k\}$ be the sequence produced by Algorithm DF. Then we have*

(i)

$$\lim_{k\to\infty} \max_{1\leq i\leq r_k} \left\{\alpha_k^i\right\} = 0, \tag{34}$$

$$\lim_{k\to\infty} \max_{1\leq i\leq r_k} \left\{\tilde{\alpha}_k^i\right\} = 0, \tag{35}$$

$$\lim_{k\to\infty} \max_{1\leq i\leq r_k} \left\|x_k - y_k^i\right\| = 0. \tag{36}$$

(ii) $\bar{\alpha}_k^i \geq \epsilon/c - \left\|x_k - y_k^i\right\|$ *whenever* $d_k^i \in T(x_k, \epsilon)$ *and* $\epsilon > 0$, *where* $c = \max_{j=1,\ldots,m} \|a_j\|$.

Proof. To prove assertion (i), we note from the construction of α_k^i and y_k^{i+1} in Step 2.3 that

$$f(y_k^{i+1}, \mu_k) \le f(y_k^i, \mu_k) - \gamma(\alpha_k^i)^2 \tag{37}$$

and from the construction of $\tilde{\alpha}_k^{i+1}$ that for each k and each $i \in \{1, \dots, r_k\}$, one of the following holds:

$$\tilde{\alpha}_k^{i+1} = \alpha_k^i, \tag{38}$$

$$\tilde{\alpha}_k^{i+1} = \theta \tilde{\alpha}_k^i. \tag{39}$$

Summing (37) for $i = 1, \dots, r_k$ and using the construction of x_{k+1} in Step 3 yields

$$f(x_{k+1}, \mu_k) \le f(x_k, \mu_k) - \gamma \sum_{i=1}^{r_k} (\alpha_k^i)^2.$$

Recalling point (v) of Proposition 5, $\{f(x_k, \mu_k)\}$ and $\{f(x_{k+1}, \mu_k)\}$ are both convergent and have the same limit, and $\{\sum_{i=1}^{r_k}(\alpha_k^i)^2\} \to 0$, thus proving (34).

For all k we have

$$\tilde{\alpha}_k^i = (\theta)^{p_k^i} \, \alpha_{m_k^i}^{l_k^i}, \tag{40}$$

where $m_k^i \le k$ and $l_k^i \le r_{m_k^i}$ are, respectively, the largest iteration index and the largest direction index such that (38) holds, and the exponent p_k^i is given by

$$p_k^i = \begin{cases} i - l_k^i & \text{if } m_k^i = k, \\ i + r_{k-1} + r_{k-2} + \cdots + r_{m_k^i} - l_k^i & \text{otherwise.} \end{cases} \tag{41}$$

Then let i be an arbitrary integer such that the set $K^i = \{k \in \{0, 1, \dots\} : r_k \ge i\}$ has infinitely many elements. If $m_k^i \to \infty$, as $k \to \infty$ with $k \in K^i$, then, by (40) and (34), we get (35).

On the other hand, suppose that m_k^i is bounded above. In this case, for all $k \in K^i$ sufficiently large, $m_k^i < k$, so that p_k^i is given by the second part of (41). Since $r_{m_k^i} \ge l_k^i$ and $r_l \ge 1$ for $l = m_k^i + 1, \dots, k-1$, this then implies that $p_k^i \ge i + (k - 1 - m_k^i)$, so that $p_k^i \to \infty$ as $k \to \infty$, $k \in K^i$. Hence, by (40) and $\theta \in (0, 1)$ we get (35).

Then we note from the updating formula for y_k^i in Step 2.4 that

$$x_k - y_k^i = -\sum_{l=1}^{i-1} \alpha_k^l d_k^l.$$

Then, using (34), $\|d_k^l\| = 1$ for $1 \le l \le r_k$, $i \le r_k$, and the assumption that $\{r_k\}$ is bounded, we obtain (36).

To prove assertion (ii), we note that, by the fact that $d_k^i \in T(x_k; \epsilon)$ and by the definition of $\bar{\alpha}_k^i$ in Step 2.2, either $\bar{\alpha}_k^i = +\infty$ (in which case, the result is proved) or an index $\bar{\jmath} \notin I(x_k; \epsilon)$ exists such that

$$a_{\bar{\jmath}}^T (y_k^i + \bar{\alpha}_k^i d_k^i) = b_{\bar{\jmath}}.$$

In the latter case, solving for $\bar{\alpha}_k^i$ and using $0 < a_{\bar{\jmath}}^T d_k^i \leq c$ (where $c = \max_{j=1,\dots,m} \|a_j\|$) yields

$$\begin{aligned}\bar{\alpha}_k^i &= \left(b_{\bar{\jmath}} - a_{\bar{\jmath}}^T y_k^i\right)/(a_{\bar{\jmath}}^T d_k^i)\\ &\geq \left(b_{\bar{\jmath}} - a_{\bar{\jmath}}^T y_k^i\right)/c\\ &= \left(b_{\bar{\jmath}} - a_{\bar{\jmath}}^T x_k + a_{\bar{\jmath}}^T (x_k - y_k^i)\right)/c\\ &\geq \left(\epsilon + a_{\bar{\jmath}}^T (x_k - y_k^i)\right)/c\\ &\geq \left(\epsilon - \|x_k - y_k^i\| c\right)/c,\end{aligned}$$

where the second inequality follows from $\bar{\jmath} \notin I(x_k; \epsilon)$ and the definition of $I(x_k; \epsilon)$, so that the assertion is proved. □

The next proposition establishes the convergence properties of Algorithm DF.

THEOREM 1. *Let $\{x_k\}$ be the sequence generated by Algorithm DF. Then a limit point of the sequence $\{x_k\}$ exists which is a stationary point of the minimax problem* (1).

Proof. By applying the results of Proposition 6 to Step 3 of the algorithm, we get that

$$\lim_{k\to\infty} \mu_k = 0. \tag{42}$$

Let $\{x_k\}_K$ be the subsequence corresponding to the subset of indices

$$K = \{k : \mu_{k+1} < \mu_k\}, \tag{43}$$

which, due to (42), has infinitely many elements.

Now let $\bar{x}$ be an accumulation point of the subsequence $\{x_k\}_K$, and let $\epsilon \in (0, \min\{\bar{\epsilon}, \epsilon^\star\}]$, where $\bar{\epsilon}$ and $\epsilon^\star$ are defined in Algorithm DF and Proposition 3, respectively. Let

$$J_k = \{i \in \{1, \dots, r_k\} : d_k^i \in D_k \cap T(x_k, \epsilon)\}.$$

Then Proposition 3 and Assumption 2 imply that, for $k \in K$,

$$T(\bar{x}) = T(x_k; \epsilon) = cone\{D_k \cap T(x_k; \epsilon)\} = cone\{d_k^i\}_{i \in J_k}. \tag{44}$$

For all $i \in J_k$, by definition, $d_k^i \in T(x_k; \epsilon)$ so that from point (ii) of Proposition 6 we get

$$\bar{\alpha}_k^i \geq \epsilon/c - \left\|x_k - y_k^i\right\|,$$

which, by point (i) of Proposition 6, implies that there exists an index $\bar{k}$ such that, for all $k \geq \bar{k}$ and $k \in K$,

$$\alpha_k^i/\delta < \bar{\alpha}_k^i \quad \text{and} \quad \hat{\alpha}_k^i = \min\{\bar{\alpha}_k^i, \tilde{\alpha}_k^i\} = \tilde{\alpha}_k^i < \bar{\alpha}_k^i. \tag{45}$$

Then the construction of α_k^i in Step 2.3 implies that, for each $i \in J_k$, either

$$y_k^i + \frac{\alpha_k^i}{\delta} d_k^i \in \mathcal{F}, \qquad f\left(y_k^i + \frac{\alpha_k^i}{\delta} d_k^i, \mu_k\right) > f(y_k^i, \mu_k) - \gamma \left(\frac{\alpha_k^i}{\delta}\right)^2,$$

if an Expansion Step is performed, or

$$y_k^i + \hat{\alpha}_k^i d_k^i \in \mathcal{F}, \qquad f(y_k^i + \hat{\alpha}_k^i d_k^i, \mu_k) > f(y_k^i, \mu_k) - \gamma(\hat{\alpha}_k^i)^2.$$

By letting $\xi_k^i = \alpha_k^i/\delta$ in the first case and $\xi_k^i = \hat{\alpha}_k^i$ in the second case, we have

$$f(y_k^i + \xi_k^i d_k^i, \mu_k) > f(y_k^i, \mu_k) - \gamma(\xi_k^i)^2. \tag{46}$$

From the updating formula for y_k^i in Step 2.4 of Algorithm DF, we note that

$$\|y_k^i - x_k\| \le \sum_{l=1}^{i-1} \alpha_k^l \le \delta \sum_{l=1}^{i-1} \xi_k^l \le \delta r_k \max_{j \in J_k}\{\xi_k^j\}, \tag{47}$$

from which we get

$$\max_{i \in J_k}\{\xi_k^i, \|x_k - y_k^i\|\} \le \max\{1, \delta r_k\} \max_{i \in J_k}\{\xi_k^i\}. \tag{48}$$

Since $r_k \ge 1$, $\delta \in (0,1)$, and, by definition of ξ_k^i, $\max_{i \in J_k}\{\xi_k^i\} \le \max_{i \in J_k}\{\tilde{\alpha}_k^i, \alpha_k^i/\delta\}$, we have

$$\max\{1, \delta r_k\} \max_{i \in J_k}\{\xi_k^i\} \le \frac{r_k}{\delta} \max_{i \in J_k}\{\tilde{\alpha}_k^i, \alpha_k^i\}. \tag{49}$$

Recalling the definition of K (see (43)), it follows from Step 3 of Algorithm DF that

$$\mu_k^2 > \max_{j=1,\dots,r_k} \left\{\tilde{\alpha}_k^j, \alpha_k^j\right\} = \mu_{k+1}^2, \tag{50}$$

so that, by (48), (49), and (50), we obtain $\max_{i \in J_k}\{\xi_k^i, \|x_k - y_k^i\|\} < \frac{r_k}{\delta}\mu_k^2$, from which we get

$$\lim_{k \to \infty, k \in K} \frac{\max_{i \in J_k}\{\xi_k^i, \|x_k - y_k^i\|\}}{\mu_k} = 0. \tag{51}$$

Finally, (42), (46), (51), and the result of Proposition 4 conclude the proof. □

COROLLARY 1. *Let $\{x_k\}$ be the sequence produced by Algorithm DF, and let $\{x_k\}_K$ be the subsequence corresponding to the subset of indices K such that*

$$K = \{k : \mu_{k+1} < \mu_k\}.$$

Then every accumulation point of $\{x_k\}_K$ is a stationary point of the minimax problem (1).

5. Numerical results. The aim of the computational experiments is to investigate the ability of the proposed algorithm to locate a good approximation to a solution of the finite minimax problem (1). We report numerical results obtained by Algorithm DF both on a set of 33 unconstrained minimax problems with $n \in [1, 200]$ and $q \in [2, 501]$ (see [6] and [16] for a description of these problems) and on a set of 5 linearly constrained minimax problems with $n \in [2, 20]$, $q \in [3, 14]$, and $m \in [1, 4]$ (see [14] for a description of these test problems). We used as starting points those reported in [6], [16], and [14].

Parameter values used in the algorithm were chosen as follows:

$$\begin{aligned} \textit{init\_step}_0 = 1.0, \mu_0 = 1.0, \quad \gamma = 10^{-6}, \\ \theta = 0.5, \delta = 0.5, \quad \bar{\epsilon} = 1.0\,. \end{aligned}$$

As for the search directions, in the linearly constrained setting we use the computation strategy proposed in [10], whereas, in the unconstrained case, we use $D_k = D = \{\pm e_1, \dots, \pm e_n\}$. In the latter case, we further exploit the fact that D_k is constant. First, we modify Step 2 by adopting the stepsize updating strategy proposed in [12], in which each search direction e_i, $i = 1, \dots, n$, has its own associated stepsize. Furthermore, in Step 3 a point $\hat{x}$ is computed by performing a linesearch along an additional direction described at Step 4 of Algorithm 3 in [12]. Then $x_{k+1} = \hat{x}$, provided that $f(\hat{x}, \mu_k) \le f(y_k^{i+1}, \mu_k)$; otherwise, we set $x_{k+1} = y_k^{i+1}$. We note that in the linearly constrained case we always set $x_{k+1} = y_k^{i+1}$.

For the stopping condition, we choose to stop the algorithm when $\textit{init\_step}_k \le 10^{-4}$ in the constrained case and when $\max_{i=1,\dots,n} \tilde{\alpha}_k^i \le 10^{-4}$ in the unconstrained case. Furthermore, we also stop the computation whenever the code reaches a total of 50000 function evaluations.

Table 1 shows the numerical results obtained by Algorithm DF. The table reports, for each problem, its name, number n of variables, number q of component functions, number m of linear constraints, and number nF of function evaluations required to satisfy the stopping condition. We denote by $f(\bar{x})$ the minimum value obtained by Algorithm DF, by $\bar{\mu}$ the value of the smoothing parameter when the stopping condition is met, and by $f^\star$ the value of the known solution. Furthermore, we denote by

$$\Delta = \frac{f(\bar{x}) - f^\star}{1 + |f^\star|}$$

the error at the solution obtained by Algorithm DF.

The results reported in Table 1 show that Algorithm DF is able to locate a good estimate of the minimum point of the minimax problem (1) (as reported in [14] and [16]) with a limited number of function evaluations, especially for problems with a reasonably small number of variables (e.g., fewer than 10). It is worth noting that for almost every problem, the final smoothing parameter value is of order 10^{-2} or less.

In order to better point out the efficiency of the proposed approach, we compare Algorithm DF with some reasonable modifications of it. First, it seems reasonable to test a modified version of Algorithm DF, which we call $\mathrm{DF}_{\texttt{mod1}}$, that always uses the max function $f(x)$ instead of the smooth approximation $f(x, \mu)$. This helps us to evaluate the computational benefit of our method, with its first-order stationary result, versus a modification that possesses a much weaker convergence property, as shown in Appendix A. Second, in order to judge the effectiveness of the updating rule for the smoothing parameter, we choose to compare Algorithm DF with Algorithms $\mathrm{DF}_{\texttt{mod2}}$ and $\mathrm{DF}_{\texttt{mod3}}$, which can be obtained from Algorithm DF by dropping the updating rule for μ at Step 3 and choosing $\mu_0 = 1$ and $\mu_0 = 10^{-2}$, respectively.

The complete results obtained by the three modified versions of Algorithm DF ($\mathrm{DF}_{\texttt{mod1}}$, $\mathrm{DF}_{\texttt{mod2}}$, and $\mathrm{DF}_{\texttt{mod3}}$) are reported in Appendix B. Here, for the sake of clarity, we report only a summary of the obtained results. For each algorithm, Table 2 indicates how many problems were solved to within the accuracy specified by the column labels, while Table 3 reports the number of function evaluations. In particular, for every pair of algorithms (DF, $\mathrm{DF}_{\texttt{modi}}$, $i = 1, 2, 3$), we identify those problems solved

Table 1
Numerical performance of Algorithm DF.

PROBLEM	n	q	m	nF	$f(\bar{x})$	$\bar{\mu}$	$f^\star$	Δ
crescent	2	2	0	160	3.061E-03	1.105E-02	0.000E+00	3.061E-03
polak 1	2	2	0	106	2.718E+00	7.812E-03	2.718E+00	7.654E-09
lq	2	2	0	343	-1.411E+00	7.812E-03	-1.414E+00	1.158E-03
mifflin 1	2	2	0	65	-1.000E+00	1.210E-02	-1.000E+00	0.000E+00
mifflin 2	2	2	0	188	-9.980E-01	7.813E-03	-1.000E+00	1.009E-03
char.-conn 1	2	3	0	118	1.954E+00	9.882E-03	1.952E+00	4.631E-04
char.-conn 2	2	3	0	208	2.003E+00	1.105E-02	2.000E+00	1.153E-03
demy-malo	2	3	0	84	-3.000E+00	1.105E-02	-3.000E+00	0.000E+00
ql	2	3	0	132	7.203E+00	1.105E-02	7.200E+00	3.575E-04
hald-mad. 1	2	4	0	170	1.582E-02	1.105E-02	0.000E+00	1.582E-02
rosen	4	4	0	368	-4.394E+01	7.906E-03	-4.400E+01	1.347E-03
hald-mad. 2	5	42	0	471	6.177E-03	7.906E-03	1.220E-04	6.055E-03
polak 2	10	2	0	285	5.460E+01	7.813E-03	5.459E+01	1.134E-04
maxq	20	20	0	1858	0.000E+00	1.105E-02	0.000E+00	0.000E+00
maxl	20	40	0	891	0.000E+00	1.105E-02	0.000E+00	0.000E+00
goffin	50	50	0	2045	0.000E+00	7.813E-03	0.000E+00	0.000E+00
polak 6.1	2	3	0	131	1.954E+00	1.118E-02	1.952E+00	4.760E-04
polak 6.2	20	20	0	692	2.384E-09	1.105E-02	0.000E+00	2.384E-09
polak 6.3	4	50	0	2055	6.253E-03	7.813E-03	2.637E-03	3.607E-03
polak 6.4	4	102	0	1105	9.166E-03	7.813E-03	2.650E-03	6.499E-03
polak 6.5	4	202	0	1890	9.181E-03	7.813E-03	2.650E-03	6.515E-03
polak 6.6	3	50	0	374	6.531E-03	7.813E-03	4.500E-03	2.022E-03
polak 6.7	3	102	0	335	7.141E-03	7.813E-03	4.505E-03	2.624E-03
polak 6.8	3	202	0	369	7.263E-03	7.813E-03	4.505E-03	2.746E-03
polak 6.9	2	2	0	91	1.162E-01	7.812E-03	0.000E+00	1.162E-01
polak 6.10	1	25	0	129	1.784E-01	1.105E-02	1.782E-01	1.625E-04
polak 6.11	1	51	0	136	1.784E-01	1.105E-02	1.783E-01	6.206E-05
polak 6.12	1	101	0	153	1.784E-01	1.105E-02	1.784E-01	2.368E-05
polak 6.13	1	501	0	153	1.784E-01	1.105E-02	1.784E-01	1.464E-05
polak 6.14	100	100	0	3452	3.433E-09	1.105E-02	0.000E+00	3.433E-09
polak 6.15	200	200	0	6891	3.433E-09	1.105E-02	0.000E+00	3.433E-09
polak 6.16	100	50	0	3452	5.364E-09	1.105E-02	0.000E+00	5.364E-09
polak 6.17	200	50	0	7233	1.023E-08	7.812E-03	0.000E+00	1.023E-08
mad 1	2	3	1	43	-3.896E-01	1.747E-02	-3.897E-01	5.878E-05
mad 2	2	3	1	42	-3.304E-01	1.353E-02	-3.304E-01	-9.735E-10
mad 4	2	3	2	72	-4.489E-01	1.562E-02	-4.489E-01	4.601E-07
wong 2	10	6	3	236	2.522E+01	1.948E-02	2.431E+01	3.609E-02
wong 3	20	14	4	451	1.076E+02	2.545E-02	1.337E+02	-1.938E-01

Table 2
Comparison of methods: Number of problems solved to within a given accuracy.

	$\Delta < 10^{-3}$	$10^{-3} \leq \Delta < 10^{-1}$	$\Delta \geq 10^{-1}$
DF	23	14	1
$\mathrm{DF}_{\texttt{mod1}}$	14	12	12
$\mathrm{DF}_{\texttt{mod2}}$	16	16	6
$\mathrm{DF}_{\texttt{mod3}}$	21	14	3

Table 3

Comparison of methods: Cumulative number of function evaluations to solve the same problems to within a given accuracy.

	$\Delta < 10^{-3}$	$10^{-3} \leq \Delta < 10^{-1}$	$\Delta \geq 10^{-1}$
DF	3649	947	91
$\text{DF}_{\texttt{mod1}}$	3645	703	88
DF	27662	8112	91
$\text{DF}_{\texttt{mod2}}$	27662	7736	91
DF	7586	7716	91
$\text{DF}_{\texttt{mod3}}$	22164	8252	88

with the same accuracy both by DF and $\text{DF}_{\texttt{modi}}$ and compare the sum of the required number of function evaluations.

From these results, it is clear that Algorithm DF outperformed Algorithms $\text{DF}_{\texttt{mod1}}$ and $\text{DF}_{\texttt{mod2}}$. In fact, DF solved to high accuracy ($\Delta < 10^{-3}$) a larger number of problems with a comparable number of function evaluations. Furthermore, the comparison between Algorithms DF and $\text{DF}_{\texttt{mod1}}$, in terms of number of failures ($\Delta \geq 10^{-1}$), shows the computational advantage of using an algorithm with stronger convergence properties. As for method $\text{DF}_{\texttt{mod3}}$, it has two failures ($\Delta \geq 10^{-1}$) more than DF, but it still performs well and seems to exhibit a behavior quite similar to that of DF. However, as seen in Table 3, the two algorithms perform quite differently in terms of function evaluations. This difference in performance properly points out the fundamental importance of the updating rule for the smoothing parameter μ, whose ultimate task is that of limiting the ill-conditioning of the approximating problem. Indeed, when we fix the smoothing parameter to 10^{-2}, the problem is too ill-conditioned from the beginning of the solution process. On the other hand, Algorithm DF limits the possible ill-conditioning by decreasing the smoothing parameter at a suitable rate.

6. Appendix A.

A function $f : \Re^n \to \Re$ is said to be locally *Lipschitz* near a point $x \in \Re^n$ if there exist $L > 0$ and $\delta > 0$ such that

$$|f(y_1) - f(y_2)| \leq L\|y_1 - y_2\|$$

for all y_1, y_2 belonging to the open ball $\{y \in \Re^n : \|y - x\| < \delta\}$. The (Clarke) *generalized directional derivative* [5] of f at x in the direction d is denoted by $f^\circ(x; d)$ and is defined as follows:

$$f^\circ(x; d) = \limsup_{y \to x, t \to 0^+} \frac{f(y + td) - f(y)}{t}. \tag{52}$$

Under the assumption that f is Lipschitz near x, $f^\circ(x; d)$ is well defined.

The following proposition extends to Lipschitz continuous functions analogous results reported in [12] and [13] concerning general convergence conditions for derivative-free methods.

Proposition 7. *Let $\{x_k\}$ be a sequence of feasible points, $\bar{x}$ be a limit point of a subsequence $\{x_k\}_K$ for some infinite set $K \subseteq \{0, 1, \ldots\}$, and $f(x)$ be locally Lipschitz near $\bar{x}$. Let $\{D_k\}$, with $D_k = \{d_k^1, \ldots, d_k^{r_k}\}$, be a sequence of sets of directions which satisfy Assumption 2 and $J_k = \{i \in \{1, \ldots, r_k\} : d_k^i \in T(x_k, \epsilon)\}$ with $\epsilon \in (0, \min\{\bar{\epsilon}, \epsilon^\star\}]$ (where $\bar{\epsilon}$ and $\epsilon^\star$ are defined in Assumption 2 and Proposition 3, respectively).*

Suppose that the following condition holds: for each $k \in K$ and $i \in J_k$, there exist y_k^i and scalars $\xi_k^i > 0$ such that

$$y_k^i + \xi_k^i d_k^i \in \mathcal{F}; \tag{53}$$

$$f(y_k^i + \xi_k^i d_k^i) \geq f(y_k^i) - o(\xi_k^i); \tag{54}$$

$$\lim_{k\to\infty, k\in K} \max_{i\in J_k} \{\xi_k^i\} = 0; \tag{55}$$

$$\lim_{k\to\infty} \max_{i\in J_k} \|x_k - y_k^i\| = 0. \tag{56}$$

Then

$$\lim_{k\to\infty, k\in K} \min_{i\in J_k} \left\{\min\{0, f^\circ(x_k; d_k^i)\}\right\} = 0. \tag{57}$$

Proof. Since $\bigcup_{k\in K} D_k$ is a finite set, there exist infinite subsets $K_1 \subseteq K$ and $J \subset \{1, 2, \dots\}$ and a positive integer r such that

$$J_k = J \quad \forall k \in K_1,$$

$$\{d_k^i\}_{i\in J_k} = \{\bar{d}^1, \dots, \bar{d}^r\}, \qquad \|\bar{d}^i\| = 1 \quad \forall k \in K_1.$$

By using condition (56) it follows that

$$\lim_{k\to\infty, k\in K_1} y_k^i = \bar{x}, \qquad i \in J. \tag{58}$$

Now, recalling condition (54), for all $k \in K_1$, we have

$$f(y_k^i + \xi_k^i \bar{d}^i) - f(y_k^i) \geq -o(\xi_k^i), \qquad i \in J, \tag{59}$$

from which we obtain

$$\limsup_{k\to\infty, k\in K_1} \frac{f(y_k^i + \xi_k^i \bar{d}^i) - f(y_k^i)}{\xi_k^i} \geq 0. \tag{60}$$

Since $f(x)$ is locally Lipschitz near $\bar{x}$, by using (52), (55), and (58) we can write

$$f^\circ(\bar{x}; \bar{d}^i) \geq \limsup_{k\to\infty, k\in K_1} \frac{f(y_k^i + \xi_k^i \bar{d}^i) - f(y_k^i)}{\xi_k^i}, \qquad i = 1, \dots, r,$$

so that, from (60), we obtain

$$f^\circ(\bar{x}; \bar{d}^i) \geq 0, \qquad i = 1, \dots, r, \tag{61}$$

which proves (57). $\square$

7. Appendix B.

Here we report the complete results for the modified versions of Algorithm DF, namely $\text{DF}_{\texttt{mod1}}$, $\text{DF}_{\texttt{mod2}}$, and $\text{DF}_{\texttt{mod3}}$, in Tables 4, 5, and 6.

Table 4
Numerical performance of Algorithm $DF_{\texttt{mod1}}$.

PROBLEM	n	q	m	nF	$f(\bar{x})$	$\bar{\mu}$	$f^\star$	Δ
crescent	2	2	0	78	0.000E+00	1.105E-02	0.000E+00	0.000E+00
polak 1	2	2	0	106	2.718E+00	7.812E-03	2.718E+00	7.654E-09
lq	2	2	0	86	-1.395E+00	7.812E-03	-1.414E+00	7.771E-03
mifflin 1	2	2	0	185	-1.000E+00	1.210E-02	-1.000E+00	6.358E-08
mifflin 2	2	2	0	74	-1.000E+00	7.812E-03	-1.000E+00	0.000E+00
char.-con 1	2	3	0	80	2.000E+00	7.812E-03	1.952E+00	1.618E-02
char.-con 2	2	3	0	81	2.000E+00	1.105E-02	2.000E+00	0.000E+00
demy-malo	2	3	0	84	-3.000E+00	1.105E-02	-3.000E+00	0.000E+00
ql	2	3	0	92	7.812E+00	7.812E-03	7.200E+00	7.470E-02
hald-mad 1	2	4	0	122	1.767E-01	1.105E-02	0.000E+00	1.767E-01
rosen	4	4	0	259	-4.378E+01	7.906E-03	-4.400E+01	4.821E-03
hald-mad 2	5	42	0	194	3.126E-01	7.906E-03	1.220E-04	3.124E-01
polak 2	10	2	0	285	5.460E+01	7.813E-03	5.459E+01	1.134E-04
maxq	20	20	0	7190	8.713E-03	7.813E-03	0.000E+00	8.713E-03
maxl	20	40	0	12111	3.028E-03	7.813E-03	0.000E+00	3.028E-03
goffin	50	50	0	2045	0.000E+00	7.813E-03	0.000E+00	0.000E+00
polak 6.1	2	3	0	92	1.973E+00	7.906E-03	1.952E+00	7.087E-03
polak 6.2	20	20	0	5174	1.553E-03	9.244E-03	0.000E+00	1.553E-03
polak 6.3	4	50	0	138	5.467E-01	7.813E-03	2.637E-03	5.426E-01
polak 6.4	4	102	0	138	5.497E-01	7.813E-03	2.650E-03	5.456E-01
polak 6.5	4	202	0	139	5.495E-01	7.813E-03	2.650E-03	5.454E-01
polak 6.6	3	50	0	104	5.441E-01	7.813E-03	4.500E-03	5.372E-01
polak 6.7	3	102	0	104	5.441E-01	7.813E-03	4.505E-03	5.372E-01
polak 6.8	3	202	0	104	5.441E-01	7.813E-03	4.505E-03	5.372E-01
polak 6.9	2	2	0	88	1.161E-01	7.812E-03	0.000E+00	1.161E-01
polak 6.10	1	25	0	58	1.782E-01	1.105E-02	1.782E-01	6.121E-07
polak 6.11	1	51	0	60	1.783E-01	1.105E-02	1.783E-01	6.630E-08
polak 6.12	1	101	0	61	1.784E-01	1.105E-02	1.784E-01	5.382E-07
polak 6.13	1	501	0	59	1.784E-01	1.105E-02	1.784E-01	1.021E-07
polak 6.14	100	100	0	44694	3.337E-03	7.812E-03	0.000E+00	3.337E-03
polak 6.15	200	200	0	50001	1.210E-01	1.914E-02	0.000E+00	1.210E-01
polak 6.16	100	50	0	50002	1.621E-01	2.210E-02	0.000E+00	1.621E-01
polak 6.17	200	50	0	50003	1.782E+00	3.125E-02	0.000E+00	1.782E+00
mad 1	2	3	1	105	-3.879E-01	1.235E-02	-3.897E-01	1.246E-03
mad 2	2	3	1	42	-3.304E-01	1.353E-02	-3.304E-01	-9.735E-10
mad 4	2	3	2	201	-4.461E-01	1.105E-02	-4.489E-01	1.967E-03
wong 2	10	6	3	358	2.654E+01	1.377E-02	2.431E+01	8.830E-02
wong 3	20	14	4	660	1.019E+02	1.271E-02	1.337E+02	-2.364E-01

Table 5
Numerical performance of Algorithm $DF_{\texttt{mod2}}$.

PROBLEM	n	q	m	nF	$f(\bar{x})$	$\bar{\mu}$	$f^\star$	Δ
crescent	2	2	0	78	2.418E-01	1.000E+00	0.000E+00	2.418E-01
polak 1	2	2	0	106	2.718E+00	1.000E+00	2.718E+00	7.654E-09
lq	2	2	0	95	-1.274E+00	1.000E+00	-1.414E+00	5.796E-02
mifflin 1	2	2	0	65	-1.000E+00	1.000E+00	-1.000E+00	0.000E+00
mifflin 2	2	2	0	77	-8.193E-01	1.000E+00	-1.000E+00	9.033E-02
char.-con 1	2	3	0	94	2.041E+00	1.000E+00	1.952E+00	3.017E-02
char.-con 2	2	3	0	81	2.223E+00	1.000E+00	2.000E+00	7.435E-02
demy-malo	2	3	0	84	-3.000E+00	1.000E+00	-3.000E+00	0.000E+00
ql	2	3	0	156	7.473E+00	1.000E+00	7.200E+00	3.332E-02
hald-mad 1	2	4	0	292	8.496E-03	1.000E+00	0.000E+00	8.496E-03
rosen	4	4	0	515	-4.356E+01	1.000E+00	-4.400E+01	9.842E-03
hald-mad 2	5	42	0	299	9.496E-03	1.000E+00	1.220E-04	9.372E-03
polak 2	10	2	0	285	5.460E+01	1.000E+00	5.459E+01	1.134E-04
maxq	20	20	0	1858	0.000E+00	1.000E+00	0.000E+00	0.000E+00
maxl	20	40	0	891	0.000E+00	1.000E+00	0.000E+00	0.000E+00
goffin	50	50	0	2045	0.000E+00	1.000E+00	0.000E+00	0.000E+00
polak 6.1	2	3	0	106	2.041E+00	1.000E+00	1.952E+00	3.014E-02
polak 6.2	20	20	0	692	2.384E-09	1.000E+00	0.000E+00	2.384E-09
polak 6.3	4	50	0	1527	8.864E-03	1.000E+00	2.637E-03	6.211E-03
polak 6.4	4	102	0	2260	7.785E-03	1.000E+00	2.650E-03	5.122E-03
polak 6.5	4	202	0	1428	1.106E-02	1.000E+00	2.650E-03	8.388E-03
polak 6.6	3	50	0	262	6.592E-03	1.000E+00	4.500E-03	2.083E-03
polak 6.7	3	102	0	264	8.179E-03	1.000E+00	4.505E-03	3.657E-03
polak 6.8	3	202	0	400	8.545E-03	1.000E+00	4.505E-03	4.022E-03
polak 6.9	2	2	0	91	1.162E-01	1.000E+00	0.000E+00	1.162E-01
polak 6.10	1	25	0	52	1.038E+00	1.000E+00	1.782E-01	7.300E-01
polak 6.11	1	51	0	53	1.105E+00	1.000E+00	1.783E-01	7.866E-01
polak 6.12	1	101	0	51	1.139E+00	1.000E+00	1.784E-01	8.150E-01
polak 6.13	1	501	0	57	1.167E+00	1.000E+00	1.784E-01	8.389E-01
polak 6.14	100	100	0	3452	3.433E-09	1.000E+00	0.000E+00	3.433E-09
polak 6.15	200	200	0	6891	3.433E-09	1.000E+00	0.000E+00	3.433E-09
polak 6.16	100	50	0	3452	5.364E-09	1.000E+00	0.000E+00	5.364E-09
polak 6.17	200	50	0	7233	1.023E-08	1.000E+00	0.000E+00	1.023E-08
mad 1	2	3	1	43	-3.896E-01	1.000E+00	-3.897E-01	5.878E-05
mad 2	2	3	1	42	-3.304E-01	1.000E+00	-3.304E-01	-9.735E-10
mad 4	2	3	2	72	-4.489E-01	1.000E+00	-4.489E-01	4.601E-07
wong 2	10	6	3	236	2.522E+01	1.000E+00	2.431E+01	3.609E-02
wong 3	20	14	4	451	1.076E+02	1.000E+00	1.337E+02	-1.938E-01

TABLE 6
Numerical performance of Algorithm $DF_{\texttt{mod3}}$.

PROBLEM	n	q	m	nF	$f(\bar{x})$	$\bar{\mu}$	$f^\star$	Δ
crescent	2	2	0	79	2.693E-03	1.000E-02	0.000E+00	2.693E-03
polak 1	2	2	0	106	2.718E+00	1.000E-02	2.718E+00	7.654E-09
lq	2	2	0	142	-1.412E+00	1.000E-02	-1.414E+00	1.072E-03
mifflin 1	2	2	0	65	-1.000E+00	1.000E-02	-1.000E+00	0.000E+00
mifflin 2	2	2	0	74	-9.982E-01	1.000E-02	-1.000E+00	9.172E-04
char.-con 1	2	3	0	130	1.953E+00	1.000E-02	1.952E+00	4.080E-04
char.-con 2	2	3	0	91	2.003E+00	1.000E-02	2.000E+00	1.060E-03
demy-malo	2	3	0	84	-3.000E+00	1.000E-02	-3.000E+00	0.000E+00
ql	2	3	0	148	7.203E+00	1.000E-02	7.200E+00	3.656E-04
hald-mad 1	2	4	0	165	1.270E-03	1.000E-02	0.000E+00	1.270E-03
rosen	4	4	0	812	-4.399E+01	1.000E-02	-4.400E+01	3.083E-04
hald-mad 2	5	42	0	856	6.762E-03	1.000E-02	1.220E-04	6.639E-03
polak 2	10	2	0	285	5.460E+01	1.000E-02	5.459E+01	1.134E-04
maxq	20	20	0	7153	5.821E-11	1.000E-02	0.000E+00	5.821E-11
maxl	20	40	0	9663	5.913E-05	1.000E-02	0.000E+00	5.913E-05
goffin	50	50	0	2045	0.000E+00	1.000E-02	0.000E+00	0.000E+00
polak 6.1	2	3	0	329	1.953E+00	1.000E-02	1.952E+00	3.821E-04
polak 6.2	20	20	0	1305	2.384E-09	1.000E-02	0.000E+00	2.384E-09
polak 6.3	4	50	0	1990	8.010E-03	1.000E-02	2.637E-03	5.359E-03
polak 6.4	4	102	0	865	9.830E-03	1.000E-02	2.650E-03	7.162E-03
polak 6.5	4	202	0	2284	1.063E-02	1.000E-02	2.650E-03	7.963E-03
polak 6.6	3	50	0	590	6.429E-03	1.000E-02	4.500E-03	1.921E-03
polak 6.7	3	102	0	589	7.040E-03	1.000E-02	4.505E-03	2.524E-03
polak 6.8	3	202	0	365	7.446E-03	1.000E-02	4.505E-03	2.928E-03
polak 6.9	2	2	0	88	1.161E-01	1.000E-02	0.000E+00	1.161E-01
polak 6.10	1	25	0	62	1.784E-01	1.000E-02	1.782E-01	1.625E-04
polak 6.11	1	51	0	60	1.784E-01	1.000E-02	1.783E-01	5.924E-05
polak 6.12	1	101	0	61	1.784E-01	1.000E-02	1.784E-01	2.368E-05
polak 6.13	1	501	0	60	1.784E-01	1.000E-02	1.784E-01	1.464E-05
polak 6.14	100	100	0	50005	3.713E-02	1.000E-02	0.000E+00	3.713E-02
polak 6.15	200	200	0	50002	8.690E-02	1.000E-02	0.000E+00	8.690E-02
polak 6.16	100	50	0	50001	1.617E-01	1.000E-02	0.000E+00	1.617E-01
polak 6.17	200	50	0	50001	6.276E-01	1.000E-02	0.000E+00	6.276E-01
mad 1	2	3	1	43	-3.896E-01	1.000E-02	-3.897E-01	5.878E-05
mad 2	2	3	1	42	-3.304E-01	1.000E-02	-3.304E-01	-9.735E-10
mad 4	2	3	2	72	-4.489E-01	1.000E-02	-4.489E-01	4.601E-07
wong 2	10	6	3	236	2.522E+01	1.000E-02	2.431E+01	3.609E-02
wong 3	20	14	4	451	1.076E+02	1.000E-02	1.337E+02	-1.938E-01

Acknowledgment. The authors would like to thank two anonymous referees for their careful reading of the paper and helpful comments and suggestions which led to significant improvements in the paper.

REFERENCES

[1] J. W. Bandler and C. Charalambous, *Practical least pth optimization of networks*, IEEE Trans. Microwave Theory Tech., 20 (1972), pp. 834–840.

[2] J. W. Bandler and C. Charalambous, *Nonlinear minimax optimization as a sequence of least pth optimization with finite values of p*, Internat. J. Systems Sci., 7 (1976), pp. 377–391.

[3] D. P. Bertsekas, *Constrained Optimization and Lagrange Multipliers Methods*, Academic Press, New York, 1982.

[4] C. Charalambous, *Acceleration of the least pth algorithm for minimax optimization with engineering applications*, Math. Programming, 17 (1979), pp. 270–297.

[5] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley and Sons, New York, 1983.

[6] G. Di Pillo, L. Grippo, and S. Lucidi, *A smooth method for the finite minimax problem*, Math. Programming, 60 (1993), pp. 187–214.

[7] C. Gígola and S. Gomez, *A regularization method for solving the finite convex min-max problem*, SIAM J. Numer. Anal., 27 (1990), pp. 1621–1634.

[8] T. G. Kolda, R. M. Lewis, and V. Torczon, *Optimization by direct search: New perspectives on some classical and modern methods*, SIAM Rev., 45 (2003), pp. 385–482.

[9] T. G. Kolda, R. M. Lewis, and V. Torczon, *Stationarity Results for Generating Set Search for Linearly Constrained Optimization*, Tech. report SAND2003-8550, Sandia National Laboratories, Livermore, CA, 2003, SIAM J. Optim., submitted.

[10] R. M. Lewis and V. Torczon, *Pattern search methods for linearly constrained minimization*, SIAM J. Optim., 10 (2000), pp. 917–941.

[11] X. Li, *An entropy-based aggregate method for minimax optimization*, Eng. Optim., 18 (1992), pp. 277–285.

[12] S. Lucidi and M. Sciandrone, *On the global convergence of derivative-free methods for unconstrained optimization*, SIAM J. Optim., 13 (2002), pp. 97–116.

[13] S. Lucidi, M. Sciandrone, and P. Tseng, *Objective-derivative-free methods for constrained optimization*, Math. Program., 92 (2002), pp. 37–59.

[14] L. Luksan and J. Vlcek, *Test Problems for Nonsmooth Unconstrained and Linearly Constrained Optimization*, Tech. report 798, Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague, Czech Republic, 2000.

[15] D. Q. Mayne and E. Polak, *Nondifferentiable optimization via adaptive smoothing*, J. Optim. Theory Appl., 43 (1984), pp. 601–614.

[16] E. Polak, J. O. Royset, and R. S. Womersley, *Algorithms with adaptive smoothing for finite minimax problems*, J. Optim. Theory Appl., 119 (2003), pp. 459–484.

[17] R. A. Polyak, *Smooth optimization methods for minimax problems*, SIAM J. Control Optim., 26 (1988), pp. 1274–1286.

[18] F. Guerra Vazquez, H. Gunzel, and H. T. Jongen, *On logarithmic smoothing of the maximum function*, Ann. Oper. Res., 101 (2001), pp. 209–220.

[19] S. Xu, *Smoothing method for minimax problems*, Comput. Optim. Appl., 20 (2001), pp. 267–279.

[20] I. Zang, *A smoothing technique for min-max optimization*, Math. Programming, 19 (1980), pp. 61–77.

SIAM J. OPTIM.
Vol. 16, No. 4, pp. 1076–1091

LMI APPROXIMATIONS FOR CONES OF POSITIVE SEMIDEFINITE FORMS*

LUIS F. ZULUAGA[†], JUAN VERA[‡], AND JAVIER PEÑA[§]

Abstract. An interesting recent trend in optimization is the application of semidefinite programming techniques to new classes of optimization problems. In particular, this trend has been successful in showing that under suitable circumstances, polynomial optimization problems can be approximated via a sequence of semidefinite programs. Similar ideas apply to conic optimization over the cone of copositive matrices and to certain optimization problems involving random variables with some known moment information.

We bring together several of these approximation results by studying the approximability of cones of *positive semidefinite forms* (homogeneous polynomials). Our approach enables us to extend the existing methodology to new approximation schemes. In particular, we derive a novel approximation to the cone of *copositive* forms, that is, the cone of forms that are positive semidefinite over the nonnegative orthant. The format of our construction can be extended to forms that are positive semidefinite over more general conic domains. We also construct polyhedral approximations to cones of positive semidefinite forms over a polyhedral domain. This opens the possibility of using linear programming technology in optimization problems over these cones.

Key words. positive polynomials, global optimization, linear matrix inequalities, conic programming, semidefinite programming

AMS subject classifications. 90C22, 90C25, 90C30, 90C34

DOI. 10.1137/03060151X

1. Introduction.

An interesting recent trend in optimization is the use of semidefinite programming techniques for solving or approximating new classes of optimization problems. In particular, Lasserre [15] proposed a general solution approach for polynomial optimization problems via semidefinite programming. Independently, Parrilo [20, 21] developed semidefinite programming techniques to address semialgebraic problems in control theory. In addition to the work by Lasserre and Parrilo, the idea of approximating a set of positive semidefinite polynomials is also present in the work by Bertsimas and Popescu [1], de Klerk and Pasechnik [3], Laurent [17, 18], Popescu [22], and Kojima, Kim, and Waki [14].

A fundamental ingredient underlying most of these approaches, as well as earlier related work by Shor [30], Shor and Stetsyuk [31], and Nesterov [19], is to recast the feasibility of a finite system of polynomial equations and inequalities in terms of an *alternative* polynomial identity involving squares of (unknown) polynomials. Computable relaxations (via semidefinite programming) of the feasibility problem can then be obtained by solving a degree-restricted version of the alternative polyno-

*Received by the editors October 29, 2003; accepted for publication (in revised form) September 6, 2005; published electronically February 3, 2006. This research was supported by NSF grant CCF-0092655.

http://www.siam.org/journals/siopt/16-4/60151.html

[†]Faculty of Business Administration, University of New Brunswick, Fredericton, NB E3B 5A3, Canada (lzuluaga@unb.ca). This author's work was partially supported by NSF grant DMI-0098427 and NSERC grant 31814-05. This paper was written while the author was a graduate student at the Tepper School of Business at Carnegie Mellon University.

[‡]Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213 (jvera@andrew.cmu.edu). This paper was written while the author was visiting the Universidad de los Andes in Bogotá, Colombia.

[§]Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA 15213-3890 (jfp@andrew.cmu.edu).

mial identity. For instance, the Markov–Lukacs theorem states that a single-variable polynomial is nonnegative if and only if it is a sum of squares of two single-variable polynomials. The latter can then be recast in terms of positive semidefinite matrices as discussed by Nesterov [19]. In general, for a system of finitely many polynomial equations and inequalities, the powerful Positivstellensatz theorem from real algebraic geometry (see, e.g., [2, 23]) ensures the existence of such an alternative polynomial identity. When the problem possesses additional structure, more specialized versions of the Positivstellensatz can be applied. Some such results are the theorems of Schmüdgen [29], Putinar [24], Pólya (see, e.g., [8]), Reznick [26], and Handelman [6]. These results provide a fundamental step in the development of solution techniques for various classes of polynomial optimization problems via semidefinite programming [1, 3, 14, 15, 16, 17, 18, 20, 21].

In this paper we study the approximability of the cone $P_{n,m}(D)$ of positive semidefinite forms (homogeneous polynomials) of degree m over a semialgebraic conic domain $D \subseteq \mathbb{R}^n$. This approach allows us to bring together a number of previously known approximation results for polynomial optimization problems. By considering the cone of positive semidefinite forms $P_{n,m}(D)$, we can systematically apply homogenized versions of representation theorems from algebraic geometry to show that a given cone of positive semidefinite forms can be approximated by a sequence of cones, where each cone in the sequence has a description in terms of linear matrix inequalities (LMI). This generic approximation format is an extension of the ones presented by de Klerk and Pasechnik [3] and by Lasserre [15]. De Klerk and Pasechnik show that Parrilo's hierarchy of sufficient criteria for copositivity can be seen as sequence of cones that converge to the copositive cone, where each cone in the sequence has an LMI description. Lasserre's approximation approach for polynomial optimization problems can also be phrased, after a suitable homogenization, in a similar fashion.

In addition to gathering several previously known approximation results for polynomial optimization problems, our approach to cones of positive semidefinite forms enables us to develop some new approximation results. In particular, we give a generalization of the (sufficient) criterion for copositivity proposed by Parrilo in [20] (section 4). In the approximation format above, this corresponds to a sequence of cones converging to $P_{n,m}(\mathbb{R}^n_+)$, each of which has an LMI description. The two key ideas of our construction are to *approximate* $P_{n,m}(\mathbb{R}^n_+)$ with simply described sets $E_{n,m}(\mathbb{R}^n_+)$ and to *embed* $P_{n,m}(\mathbb{R}^n_+)$ in a higher-dimensional cone $P_{n,m+r}(\mathbb{R}^n_+)$. We initially introduce $E_{n,m}(\mathbb{R}^n_+)$ as the set of n-degree forms θ such that $\theta(x_1^2, \dots, x_n^2)$ is a sum of squares; subsequently, we show (Proposition 9) that $E_{n,m}(\mathbb{R}^n_+)$ has an alternative simpler description. The latter yields an interesting new description of the successive LMI approximations to the cone of copositive matrices proposed by Parrilo [20]. In addition, it allows us to extend our ideas further, first to $P_{n,m}(D)$ for a pointed polyhedral domain D (section 5) and then to $P_{n,m}(D)$ for a pointed semialgebraic conic domain D (section 6). In sections 4 and 5, the fundamental representation theorem that ensures the convergence of the constructed approximation is Pólya's theorem. In section 6, the representation theorem ensuring the convergence of the approximation sequence is Schmüdgen's theorem. Section 3, which serves as a preamble to the main three sections, discusses the conceptually simpler case of approximating the cone $P_{n,m}(\mathbb{R}^n)$ of positive semidefinite forms over $\mathbb{R}^n$. In particular, we discuss an approximation introduced by Jibetean and de Klerk [13, section 4.3] that is based on a representation theorem due to Reznick [26].

When the domain D is polyhedral, in addition to *semidefinite* approximations, we provide *polyhedral* approximations, that is, approximations via linear inequalities only, for $P_{n,m}(D)$. This construction is an extension of the polyhedral approximations for the copositive cone proposed by de Klerk and Pasechnik [3]. Although polyhedral approximations for $P_{n,m}(D)$ are in general weaker (inclusionwise) than semidefinite approximations, they open possibilities for use of the highly developed linear programming technology. Given the limitations of current semidefinite programming solvers to handle large-scale problems, the availability of polyhedral approximations can potentially yield enhancements in the solution techniques for problems involving cones of positive semidefinite forms.

The rest of the paper is organized as follows. In section 2 we introduce some key definitions and present Theorem 1, which formally defines the format of the approximation results discussed in what follows. In sections 3 and 4 we present inner approximations for the cones $P_{n,m}(\mathbb{R}^n)$ and $P_{n,m}(\mathbb{R}^n_+)$, respectively. In the latter case, which generalizes the cone of copositive matrices, along with a sequence of semidefinite approximations, we introduce a sequence of polyhedral approximations. In section 5 we generalize the construction and key results from section 4 to the cone $P_{n,m}(D)$ when D is a pointed polyhedral cone. Section 6 discusses similar results for the more general cone of positive semidefinite forms over pointed semialgebraic cones.

2. Preliminaries.

2.1. Monomials, polynomials, and forms. We begin by recalling some standard multinomial notation and terminology. Given $\alpha := (\alpha_1, \dots, \alpha_n) \in \mathbb{N}^n$ and a vector of variables $x := (x_1, \dots, x_n)$, the expression x^α denotes the monomial

$$x_1^{a_1} \cdots x_n^{\alpha_n}.$$

We also write $|\alpha|$ for $\alpha_1 + \cdots + \alpha_n$.

Let $H_{n,m}$ denote the set of forms (homogeneous polynomials) of degree m in n variables with real coefficients. A form $\theta(x)$ in $H_{n,m}$ can be written as

$$\theta(x) = \sum_{|\alpha|=m} \theta_\alpha x^\alpha.$$

We shall identify the form $\theta(x)$ with the vector of its coefficients $\theta := (\theta_\alpha)_{|\alpha|=m}$. Formally speaking, θ denotes the *vector* $(\theta_\alpha)_{|\alpha|=m}$, and for a given $x \in \mathbb{R}^n$, $\theta(x)$ denotes the *value* $\sum_{|\alpha|=m} \theta_\alpha x^\alpha$, i.e., the value of the form θ evaluated at x. Via this identification, the set $H_{n,m}$ can in turn be identified with the Euclidean space $\mathbb{R}^{n_m}$, where

$$n_m := |\{\alpha \in \mathbb{N}^n : |\alpha| = m\}| = \binom{n+m-1}{n-1}.$$

We will make extensive use of this identification. In particular we endow $H_{n,m}$ with the dot-inner product. In other words, we define the inner product of $\theta, \phi \in H_{n,m}$ as

$$\langle \theta, \phi \rangle := \sum_{|\alpha|=m} \theta_\alpha \phi_\alpha.$$

We shall also frequently use the vector-valued function $\sigma^m : \mathbb{R}^n \to \mathbb{R}^{n_m}$ defined by

$$x \mapsto (x^\alpha)_{|\alpha|=m}.$$

Notice that by construction, the following identity holds for all $\theta \in H_{n,m}$ and $x \in \mathbb{R}^n$:

$$\theta(x) = \langle \theta, \sigma^m(x) \rangle.$$

For the special case $m = 2$, the space of 2-forms $H_{n,2} = \mathbb{R}^{n_2}$ can also be identified with the space $\mathbb{S}^n$ of $n \times n$ symmetric matrices. The identification is via the one-to-one correspondence between symmetric matrices and quadratic forms $Q \in \mathbb{S}^n \mapsto q \in H_{n,2}$, where $q(x) := x^{\mathrm{T}} Q x$.

2.2. Positive semidefinite forms. Given a cone $D \subseteq \mathbb{R}^n$, let $P_{n,m}(D)$ be the cone of m-degree forms that are *positive semidefinite* in D (psd in D), i.e.,

$$P_{n,m}(D) := \{\theta \in H_{n,m} : \theta(x) \geq 0 \text{ for all } x \in D\}.$$

When $m = 2$, $D = \mathbb{R}^n$, and $H_{n,2}$ is identified with $\mathbb{S}^n$, the cone $P_{n,2}(\mathbb{R}^n)$ corresponds precisely to the cone of psd matrices, usually denoted $\mathbb{S}^n_+$. We shall write $A \succeq 0$ for $A \in \mathbb{S}^n_+$ following the usual notation in the semidefinite programming literature (see, e.g., [33]).

If $\theta \in P_{n,m}(D)$ satisfies $\theta(x) > 0$ for all $x \in D$, $x \neq 0$, then θ is said to be *positive definite* in D (pd in D). Throughout our presentation we will frequently rely on the following straightforward characterization of the interior of $P_{n,m}(D)$.

Observation 1. Assume $D \subseteq \mathbb{R}^n$ is a closed cone. Then θ is pd in D if and only if $\theta \in \operatorname{int}(P_{n,m}(D))$.

Notice that the class of cones of psd polynomials can be seen as a subclass of cones of psd forms via homogenization. Consequently, our presentation focuses on the latter class of cones.

An advantage of working with cones of forms is the algebraic characterization of the interior of $P_{n,m}(D)$ stated in Observation 1. The analog statement in general fails for cones of polynomials due to the possibility of "zeros at infinity" (see [27]). For example, the polynomial $g(x_1, x_2) = x_1^2 + (1 - x_1 x_2)^2$ is pd in $\mathbb{R}^2$, but for all $\epsilon > 0$, $g(x_1, x_2) - \epsilon$ is not psd in $\mathbb{R}^2$ because $\lim_{x \to 0} g(x, 1/x) = 0$.

2.3. LMI approximations of $\boldsymbol{P_{n,m}(D)}$. Theorem 1 summarizes the main results discussed in this paper: For several important classes of conic domains D, the cone $P_{n,m}(D)$ can be innerly approximated by a sequence of cones definable in terms of LMI. Theorem 1 can be seen as a compilation and extension of previous approximation results for polynomial optimization problems [1, 3, 13, 14, 15, 16, 17, 18, 20, 21].

THEOREM 1. *Suppose $D = \mathbb{R}^n$, $D = \mathbb{R}^n_+$, or D is a pointed semialgebraic cone. Then we can construct a sequence of cones $K^r, r = 0, 1, \ldots,$ such that*

(i) $K^r \subseteq K^{r+1} \subseteq P_{n,m}(D)$, $r = 0, 1, \ldots,$

(ii) $\operatorname{int}(P_{n,m}(D)) \subseteq \bigcup_{r=0}^{\infty} K^r$,

(iii) *each K^r has an LMI description; in other words,*

$$K^r = \{\theta \in H_{n,m} : \exists \Phi \succeq 0 \quad \text{s.t.} \quad \mathcal{L}\theta = \mathcal{T}\Phi\} \tag{1}$$

for some suitable linear mappings $\mathcal{L}, \mathcal{T}$;

(iv) *if D is a polyhedral cone, then we can also construct a polyhedral sequence of cones approximating $P_{n,m}(D)$ as above (i.e., with $\Phi \geq 0$ in* (1)*).*

Proof. See Propositions 5, 13, and 17 in sections 3, 5, and 6, respectively. □

Throughout what follows, we shall write $K^r \uparrow P_{n,m}(D)$ as shorthand for conditions (i) and (ii) in Theorem 1. Also, whenever we say that $\{K^r, r = 0, 1, \ldots\}$ is a *sequence of inner approximations* for $P_{n,m}(D)$, it will be implicitly assumed that $\{K^r, r = 0, 1, \ldots\}$ satisfies conditions (i), (ii), and (iii) in Theorem 1.

Theorem 1 in combination with Theorem 2 readily yields numerical schemes based on semidefinite programming for computing arbitrarily close approximate solutions to primal-dual pairs of conic programs of the form

$$(\mathbb{P}) \quad \begin{array}{rll} z_{\mathbb{P}} = & \inf & \langle c, \theta \rangle \\ & \text{s.t.} & A\theta = b, \\ & & \theta \in P_{n,m}(D) \end{array} \qquad (\mathbb{D}) \quad \begin{array}{rll} z_{\mathbb{D}} = & \sup & \langle b, v \rangle \\ & \text{s.t.} & c - A^* v \in P_{n,m}(D)^* \end{array}$$

for the classes of conic domains D in Theorem 1. Here A^* is the adjoint of A and $P_{n,m}(D)^*$ is the dual cone of $P_{n,m}(D)$. This general format underlies several of the ideas and results in [15, 16, 20].

Consider the primal-dual pair of conic programs obtained when $P_{n,m}(D)$ is replaced by K^r in $(\mathbb{P})$ and $(\mathbb{D})$:

$$(\mathbb{P}_r) \quad \begin{array}{rll} z_{\mathbb{P}_r} = & \inf & \langle c, \theta \rangle \\ & \text{s.t.} & A\theta = b, \\ & & \theta \in K^r \end{array} \qquad (\mathbb{D}_r) \quad \begin{array}{rll} z_{\mathbb{D}_r} = & \sup & \langle b, v \rangle \\ & \text{s.t.} & c - A^* v \in (K^r)^*. \end{array}$$

Theorem 2 formalizes the intuitively natural fact that $(\mathbb{P})$ and $(\mathbb{D})$ are approximated when the cone $P_{n,m}(D)$ is suitably approximated by a sequence of cones. This result can be seen as a strengthening of the classical strong conic duality theorem in convex analysis (cf. [25, 28]).

THEOREM 2. *Assume* $(\mathbb{D})$ *is feasible, A is surjective, and* $(\mathbb{P})$ *is strictly feasible (i.e., there exists* $\theta \in \operatorname{int}(P_{n,m}(D))$ *such that* $A\theta = b$*). Let* $K^r \uparrow P_{n,m}(D)$ *and* $(\mathbb{P})$, $(\mathbb{P}_r)$, $(\mathbb{D})$, $(\mathbb{D}_r)$ *be as above. Then*

(i) $(\mathbb{D}_r)$ *is feasible and* $z_{\mathbb{D}_r} \geq z_{\mathbb{D}_{r+1}} \geq z_{\mathbb{D}}$ *for* $r = 0, 1, \dots;$

(ii) *for r sufficiently large,* $z_{\mathbb{P}_r} = z_{\mathbb{D}_r}$ *and* $(\mathbb{D}_r)$ *has an optimal solution* v^r*;*

(iii) $\lim_{r\to\infty} z_{\mathbb{D}_r} = z_{\mathbb{D}} = z_{\mathbb{P}}$*, the set* $\{v^r : r = 0, 1, \dots\}$ *is bounded, and every limit point of* $\{v^r : r = 0, 1, \dots\}$ *is an optimal solution of* $(\mathbb{D})$.

Proof. This is a direct consequence of conic duality. It is a dual version of [34, Thm. 2] and can be proven by a similar argument. □

3. PSD forms in $\mathbb{R}^n$.

In this section we concentrate on the cone $P_{n,2m}(\mathbb{R}^n)$, which we shall abbreviate as $P_{n,2m}$.

3.1. First approximation: Sums of squares.

Let $\Sigma_{n,2m}$ denote the cone of forms in $H_{n,2m}$ that are sum of squares (sos), that is,

$$\Sigma_{n,2m} := \operatorname{conv}\{\phi(x)^2 : \phi \in H_{n,m}\}.$$

(Here $\operatorname{conv}(S)$ denotes the convex hull of the set S.)

Notice that $\Sigma_{n,2m} \subseteq P_{n,2m}(\mathbb{R}^n)$ for all m, n. This inclusion is proper, except for some special cases. This is a classical result due to Hilbert [9].

THEOREM 3 (Hilbert). $\Sigma_{n,2m} = P_{n,2m}(\mathbb{R}^n)$ *if and only if* $n \leq 2$, *or* $m \leq 1$, *or* $(n, m) = (3, 2)$.

The inclusion $\Sigma_{n,2m} \subseteq P_{n,2m}$ gives an inner approximation of $P_{n,2m}$ and henceforth a sufficient condition for positive semidefiniteness: a form is psd if it is a sos. Notice that for $\phi \in H_{n,m}$ we have

$$\phi(x)^2 = \langle \phi, \sigma^m(x) \rangle^2 = \sigma^m(x)^{\mathrm{T}}(\phi\phi^{\mathrm{T}})\sigma^m(x).$$

This yields the following observation.

Observation 2. Let $\theta \in H_{n,2m}$. Then $\theta \in \Sigma_{n,2m}$ if and only if there exists $\Phi \in \mathbb{S}^{n_m}$, $\Phi \succeq 0$, such that $\theta(x) = \sigma^m(x)^{\mathrm{T}}\Phi\sigma^m(x)$.

Notice that the identity $\theta(x) = \sigma^m(x)^{\mathrm{T}}\Phi\,\sigma^m(x)$ corresponds to a linear system of equations in Φ and θ, and therefore optimizing a linear function with linear restrictions over $\Sigma_{n,2m}$ can be cast as a semidefinite programming problem.

The study of the relationship between psd forms and sos has a long history. The search for such kinds of connections is closely tied with Hilbert's 17th problem [10] and with advances in real algebra over the last century [23, 26]. Our work relies on some of these developments. For a detailed account of the rich history of this subject, we refer the reader to the excellent references [23, 26, 27].

3.2. Inner approximations for $P_{n,2m}$. We next present a sequence of inner approximations for $P_{n,2m}$ introduced by Jibetean and de Klerk [13, section 4.3]. The construction uses $\Sigma_{n,2m}$ as a starting point and is based on the following key representation theorem for pd forms due to Reznick [26].

THEOREM 4 (Reznick). *Let $\theta \in H_{n,2m}$. If θ is pd in $\mathbb{R}^n$, then there exists $r \in \mathbb{N}$ such that*

$$\left(\sum_{j=1}^{n} x_j^2\right)^r \theta(x) \in \Sigma_{n,2(m+r)}.$$

Proof. See [26, Thm. 3.12]. □

Theorem 4 naturally suggests the following sequence of inner approximations for $P_{n,2m}$ [13, section 4.3]: For $r = 0, 1, \ldots$ let

$$\begin{aligned} K^r_{n,2m}(\mathbb{R}^n) \quad &:= \quad \left\{\theta \in H_{n,2m} : \quad \left(\sum_{j=1}^{n} x_j^2\right)^r \theta(x) \in \Sigma_{n,2(m+r)}\right\} \\ &= \quad \left\{\theta \in H_{n,2m} : \quad \exists\, \Phi \in \mathbb{S}^{n_{m+r}},\ \Phi \succeq 0,\ \text{s.t.} \right. \\ &\qquad\qquad \left. \left(\sum_{j=1}^{n} x_j^2\right)^r \theta(x) = \sigma^{m+r}(x)^{\mathrm{T}}\Phi\sigma^{m+r}(x)\right\}. \end{aligned}$$

The last identity holds by Observation 2 and automatically gives an LMI description of $K^r_{n,2m}(\mathbb{R}^n)$:

$$K^r_{n,2m}(\mathbb{R}^n) = \{\theta \in H_{n,2m} : \exists\, \Phi \in \mathbb{S}^{n_{m+r}},\ \Phi \succeq 0, \quad \text{s.t.} \quad \mathcal{L}\theta = \mathcal{T}\Phi\}, \tag{2}$$

where $\mathcal{L} : H_{n,2m} \to H_{n,2(m+r)}$ and $\mathcal{T} : \mathbb{S}^{n_{m+r}} \to H_{n,2(m+r)}$ are the linear maps defined by $(\mathcal{L}\theta)(x) := (\sum_{j=1}^n x_j^2)^r\theta(x)$ and $(\mathcal{T}\Phi)(x) := \sigma^{m+r}(x)^{\mathrm{T}}\Phi\sigma^{m+r}(x)$.

PROPOSITION 5. $K^r_{n,2m}(\mathbb{R}^n) \uparrow P_{n,2m}$.

Proof. Let $\theta \in H_{n,2m}$. If $\theta(x) \in \Sigma_{n,2m}$, then $\sum_{j=1}^n x_j^2\,\theta(x) \in \Sigma_{n,2m+2}$. Also, if $\sum_{j=1}^n x_j^2\,\theta(x) \in \Sigma_{n,2m+2}$, then $\theta(x) \in P_{n,2m}$. These two facts imply $K^r_{n,2m}(\mathbb{R}^n) \subseteq K^{r+1}_{n,2m}(\mathbb{R}^n) \subseteq P_{n,2m}$ for all r. Finally, from Observation 1 and Theorem 4 it follows that $\operatorname{int}(P_{n,2m}) \subseteq \bigcup_{r=0}^{\infty} K^r_{n,2m}(\mathbb{R}^n)$. □

Remark 1. In general the inclusion $\bigcup_{r=0}^{\infty} K^r_{n,2m}(\mathbb{R}^n) \subseteq P_{n,2m}$ in Proposition 5 is strict. For example, it is known (see, e.g., [26]) that the form

$$\theta(x_1, x_2, x_3, x_4) = x_1^2(x_2^2x_3^4 + x_3^2x_4^4 + x_4^2x_2^4 - 3x_2^2x_3^2x_4^2) + x_4^8$$

satisfies $\theta \in P_{4,8}$, but for all r, $(\sum_{j=1}^4 x_j^2)^r\theta(x) \notin \Sigma_{4,8+2r}$. Thus $\theta \notin \bigcup_{r=0}^{\infty} K^r_{4,8}(\mathbb{R}^4)$.

Proposition 5 yields an approximation scheme for unconstrained polynomial optimization (for details see [7, 12, 13, 35]). This approximation scheme is similar to the one proposed by Lasserre in [15] but does not require the knowledge of any a priori bound on the size of the minimizer of the polynomial.

4. Copositive forms.

In this section we concentrate on the cone $P_{n,m}(\mathbb{R}^n_+)$, which we call the *cone of m-degree copositive forms* in n variables.

We describe two families of inner approximations of $P_{n,m}(\mathbb{R}^n_+)$. The first one is analogous to that of section 3. The second one is a sequence of polyhedral cones.

4.1. Inner approximations for $P_{n,m}(\mathbb{R}^n_+)$.

Let $\mathscr{S} : H_{n,m} \to H_{n,2m}$ be the mapping $\theta(x) \mapsto \theta(x^2) := \theta(x_1^2, \dots, x_n^2)$. In other words,

$$[\mathscr{S}\theta]_\alpha = \begin{cases} \theta_{\frac{1}{2}\alpha} & \text{if every } \alpha_i \text{ is even,} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Since $\theta(x) \geq 0$ for all $x \in \mathbb{R}^n_+$ if and only if $\theta(x^2) \geq 0$ for all $x \in \mathbb{R}^n$, it follows that

$$\theta \in P_{n,m}(\mathbb{R}^n_+) \Leftrightarrow \mathscr{S}\theta \in P_{n,2m}. \tag{4}$$

Inspired by (4) we define (for $r = 0, 1, \dots$)

$$K^r_{n,m}(\mathbb{R}^n_+) := \{\theta \in H_{n,m} : \mathscr{S}\theta \in K^r_{n,2m}(\mathbb{R}^n)\}.$$

From Proposition 5 and (4) it follows that the sequence of cones $K^r_{n,m}(\mathbb{R}^n_+)$ is a sequence of inner approximations of $P_{n,m}(\mathbb{R}^n_+)$.

Proposition 6. $K^r_{n,m}(\mathbb{R}^n_+) \uparrow P_{n,m}(\mathbb{R}^n_+)$.

The LMI description (2) for $K^r_{n,2m}(\mathbb{R}^n)$ yields an LMI description for $K^r_{n,m}(\mathbb{R}^n_+)$. However, a more concise description can be obtained via the cone of *elementary copositive forms* $E_{n,m}(\mathbb{R}^n_+) \subseteq P_{n,m}(\mathbb{R}^n_+)$, defined as follows:

$$E_{n,m}(\mathbb{R}^n_+) := \{\theta \in H_{n,m} : \mathscr{S}\theta \in \Sigma_{n,2m}\}. \tag{5}$$

This gives an alternative definition of $K^r_{n,m}(\mathbb{R}^n_+)$, namely

$$K^r_{n,m}(\mathbb{R}^n_+) = \left\{\theta \in H_{n,m} : \left(\sum_{j=1}^n x_j\right)^r \theta(x) \in E_{n,m+r}(\mathbb{R}^n_+)\right\}.$$

In section 4.3 we give an alternative and more concise description of $E_{n,m}(\mathbb{R}^n_+)$ without relying on the operator $\mathscr{S}$.

4.2. Polyhedral approximations.

Now we construct a sequence of *polyhedral* approximations for $P_{n,m}(\mathbb{R}^n_+)$. This construction is based in the representation theorem due to Pólya [8].

Theorem 7 (Pólya). *Let $\theta \in H_{n,2m}$. If θ is pd in $\mathbb{R}^n_+$, then there exists $r \in \mathbb{N}$ such that*

$$\left(\sum_{j=1}^n x_j\right)^r \theta(x) \text{ has nonnegative coefficients.}$$

Proof. See [8, Thm. 56]. □

It is now natural to define

$$C^r_{n,m}(\mathbb{R}^n_+) := \left\{ \theta \in H_{n,m} : \left(\sum_{j=1}^n x_j \right)^r \theta(x) \text{ has nonnegative coefficients} \right\}.$$

Notice that $C^r_{n,m}(\mathbb{R}^n_+)$ can also be viewed as a modification of the construction of $K^r_{n,m}(\mathbb{R}^n_+)$. Clearly, $C^r_{n,m}(\mathbb{R}^n_+) \subseteq K^r_{n,m}(\mathbb{R}^n_+)$, as any form of degree d with nonnegative coefficients is in $E_{n,d}(\mathbb{R}^n_+)$. By construction, $C^r_{n,m}(\mathbb{R}^n_+)$ has an LMI description. Indeed, $C^r_{n,m}(\mathbb{R}^n_+)$ is a polyhedral cone.

The sequence $C^r_{n,m}(\mathbb{R}^n_+)$, $r = 0, 1, \ldots$, also yields a sequence of inner approximations for $P_{n,m}(\mathbb{R}^n_+)$.

Proposition 8. $C^r_{n,m}(\mathbb{R}^n_+) \uparrow P_{n,m}(\mathbb{R}^n_+)$.

Proof. This follows from Proposition 6 and Theorem 7. □

Again, as in Proposition 5, the inclusion $\bigcup_{r=0}^{\infty} C^r_{n,m}(\mathbb{R}^n_+) \subseteq P_{n,m}(\mathbb{R}^n_+)$ is strict in general. For example, the form

$$\theta(x_1, x_2, x_3, x_4) = x_1(x_2 x_3^2 + x_3 x_4^2 + x_4 x_2^2 - 3x_2 x_3 x_4) + x_4^4$$

satisfies $\theta \in P_{4,4}(\mathbb{R}^4_+)$, but $\theta \notin \bigcup_{r=0}^{\infty} C^r_{4,4}(\mathbb{R}^4_+)$.

4.3. Alternative characterization of $E_{n,m}(\mathbb{R}^n_+)$.

The next proposition yields a characterization of the elementary copositive forms $E_{n,m}(\mathbb{R}^n_+)$ without relying on the operator $\mathscr{S}$. Aside from being more concise, this alternative description for $E_{n,m}(\mathbb{R}^n_+)$ has natural extensions to pointed polyhedral and semialgebraic cones (cf. sections 5 and 6).

Proposition 9. *The sets $E_{n,m}(\mathbb{R}^n_+)$ defined above satisfy the following identity:*

$$\begin{aligned} E_{n,m}(\mathbb{R}^n_+) = \operatorname{conv}\{x_{i_1} x_{i_2} \cdots x_{i_k} \psi(x)^2 : \quad & m-k \text{ is even}, \psi \in H_{n,(m-k)/2}, \\ & \text{and } i_1, \ldots, i_k \in \{1, \ldots, n\}\}. \end{aligned}$$

Proof. The "$\supseteq$" inclusion is immediate. For the reverse inclusion, assume $\theta \in E_{n,m}(\mathbb{R}^n_+)$. Hence $\theta(x^2) = \sum_{i=1}^k \phi_i(x)^2$ for some $\phi_i \in H_{n,m}$, $i = 1, \ldots, k$. Writing each ϕ_i in terms of its monomial expansion we get

$$\phi_i(x)^2 = \sum_{|\alpha|=m} \sum_{|\alpha'|=m} \phi_{i,\alpha} \phi_{i,\alpha'} x^\alpha x^{\alpha'}.$$

Now let $\operatorname{par}(\alpha) \in \{0,1\}^n$ be the *vector of parities* of α, defined by $\operatorname{par}(\alpha)_i = 0$ if α_i is even and $\operatorname{par}(\alpha)_i = 1$ otherwise. Since all monomials in $\theta(x^2)$ contain only variables with even powers, it follows that

$$\begin{aligned} \theta(x^2) &= \sum_{i=1}^k \phi_i(x)^2 \\ &= \sum_{i=1}^k \sum_{\beta \in \{0,1\}^n} \sum \{\phi_{i,\alpha} \phi_{i,\alpha'} x^\alpha x^{\alpha'} : \operatorname{par}(\alpha) = \operatorname{par}(\alpha') = \beta, |\alpha| = |\alpha'| = m\}. \end{aligned}$$

(All other terms cancel out.)

Thus,

$$\theta(x^2) = \sum_{i=1}^k \phi_i(x)^2 = \sum_{i=1}^k \sum_{\beta \in \{0,1\}^n} \left(\sum \{\phi_{i,\alpha} x^\alpha : \ \operatorname{par}(\alpha) = \beta, \ |\alpha| = m \ \} \right)^2.$$

Since $\alpha \geq \mathrm{par}(\alpha)$, we can rewrite the previous expression as

$$\theta(x^2) = \sum_{i=1}^{k} \phi_i(x)^2 = \sum_{i=1}^{k} \sum_{\beta \in \{0,1\}^n} x^{2\beta} \left(\sum \{ \phi_{i,\alpha} x^{\alpha - \beta} : \mathrm{par}(\alpha) = \beta, |\alpha| = m \} \right)^2 .$$

Finally, since all entries in $\alpha - \mathrm{par}(\alpha)$ are even, we get

$$\theta(x) = \sum_{i=1}^{k} \sum_{\beta \in \{0,1\}^n} x^{\beta} \left(\sum \{ \phi_{i,\alpha} x^{(\alpha - \beta)/2} : \mathrm{par}(\alpha) = \beta, |\alpha| = m \} \right)^2 . \qquad \square$$

The following inductive and LMI descriptions of the sets $E_{n,m}(\mathbb{R}^n_+)$ readily follow from Proposition 9 and Observation 2.

Corollary 10.

(i) *The sets $E_{n,m}(\mathbb{R}^n_+)$ satisfy the following recursive relationships:*

$$\begin{aligned} &E_{n,1}(\mathbb{R}^n_+) = \mathrm{conv}\{x_j : j = 1, \ldots, n\}, \\ &E_{n,2}(\mathbb{R}^n_+) = \mathrm{conv}(\{(a^{\mathrm{T}}x)^2 : a \in \mathbb{R}^n\} \cup \{x_i x_j : 1 \leq i < j \leq n\}), \\ &E_{n,2k+1}(\mathbb{R}^n_+) = \mathrm{conv}\{x_j \theta(x) : \theta \in E_{n,2k},\ j = 1, \ldots, n\}, \\ &E_{n,2k+2}(\mathbb{R}^n_+) = \mathrm{conv}(\Sigma_{n,2(k+1)} \cup \{x_j \theta(x) : \theta \in E_{n,2k+1},\ j = 1, \ldots, n\}). \end{aligned}$$

(ii) *The sets $E_{n,m}(\mathbb{R}^n)$ can be defined via the following LMI identities:*

$$\begin{aligned} E_{n,1}(\mathbb{R}^n_+) &:= \{\theta \in H_{n,1} : \quad \exists a \in \mathbb{R}^n_+ \text{ s.t. } \theta(x) = a^{\mathrm{T}} x\}, \\ E_{n,2}(\mathbb{R}^n_+) &:= \{\theta \in H_{n,2} : \quad \exists M, N \in \mathbb{S}^n,\ M \succeq 0,\ N \geq 0, \text{ s.t. } \\ &\qquad\qquad \theta(x) = x^{\mathrm{T}}(M + N)x\}, \\ E_{n,3}(\mathbb{R}^n_+) &:= \Bigg\{ \theta \in H_{n,3} : \quad \exists M^i, N^i \in \mathbb{S}^n,\ M^i \succeq 0,\ N^i \geq 0,\ i = 1, \ldots, n, \\ &\qquad\qquad \text{s.t. } \theta(x) = \sum_i x_i (x^{\mathrm{T}}(M^i + N^i)x) \Bigg\}, \\ &\ \ \vdots \end{aligned}$$

The map $\mathscr{S}$ (see (3)) establishes a parallel between the pairs $(\Sigma_{n,2m}, P_{n,2m})$ and $(E_{n,m}(\mathbb{R}^n_+), P_{n,m}(\mathbb{R}^n_+))$. Extending this parallel, note that the inclusion $E_{n,m}(\mathbb{R}^n) \subseteq P_{n,m}(\mathbb{R}^n_+)$ is proper, except for the special cases described in Proposition 11. This result follows from Theorem 3 (Hilbert's theorem) and a classical result on copositive forms due to Diananda [4, Thm. 2]. For details see [35].

Proposition 11. *$E_{n,m}(\mathbb{R}^n) = P_{n,m}(\mathbb{R}^n_+)$ if and only if $n \leq 2$, or $m = 1$, or $(n, m) = (3, 2)$, or $(n, m) = (4, 2)$.*

The particular case $m = 2$ in Proposition 6 yields the hierarchy of sufficient conditions for copositivity of matrices proposed by Parrilo [20]: a symmetric matrix $A \in \mathbb{S}^n$ is copositive (i.e., $a(x) := x^{\mathrm{T}} A x \in P_{n,2}(\mathbb{R}^n_+)$) if the following r-criterion holds:

$$\left(\sum_{j=1}^{n} x_j^2 \right)^r (x^2)^{\mathrm{T}} A\, x^2 \in \Sigma_{n,4+2r}. \tag{6}$$

The LMI description of the sets $E_{n,m}(\mathbb{R}^n_+)$ in Corollary 10 yields an alternative LMI formulation of Parrilo's r-criterion for $r = 0, 1, 2, 3, \ldots$. In particular, a new succinct derivation of the criterion for copositivity proposed by Parrilo in [20] can be obtained [35].

5. PSD forms over pointed polyhedral cones. We next construct approximation schemes for the cone $P_{n,m}(D)$ in the case when D is a pointed polyhedral cone. Throughout this section we shall assume that the domain D is the polyhedral cone

$$D = \{x : a_i^{\mathrm{T}} x \geq 0,\ i = 1, \ldots, q\}$$

for some matrix $\begin{bmatrix} a_1 & \ldots & a_q \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{q \times n}$. We shall also assume that D is *pointed*; i.e., it contains no lines. It is easy to see that the latter condition is equivalent to $\operatorname{rank}(\begin{bmatrix} a_1 & \ldots & a_q \end{bmatrix}^{\mathrm{T}}) = n$.

Notice that

$$\mathbb{R}^n_+ = \{x : e_i^{\mathrm{T}} x \geq 0,\ i = 1, \ldots, n\},$$

and therefore this section is an extension of the previous one. To get inner approximating sequences of cones for $P_{n,m}(D)$ we first extend Pólya's theorem to this context. This generalization can be obtained as a consequence of a representation theorem for polynomials positive on compact polyhedra due to Handelman [6]. (A constructive proof of such a theorem is presented in [26, Thm. 2].) At the end of this section we give a proof of this result that relies exclusively on elementary tools and Pólya's theorem.

Proposition 12. *Assume* $D = \{x : a_i^{\mathrm{T}} x \geq 0,\ i = 1, \ldots, q\}$ *is pointed and* $\theta \in \operatorname{int}(P_{n,m}(D))$. *Then, for* N *sufficiently large,*

$$(a_1^{\mathrm{T}} x + \cdots + a_q^{\mathrm{T}} x)^N \theta(x) = \phi(a_1^{\mathrm{T}} x, \ldots, a_q^{\mathrm{T}} x)$$

for some $\phi \in H_{q,m+N}$ *with* $\phi \geq 0$.

Here $\phi \geq 0$ means that the form ϕ has nonnegative coefficients.

Now we can extend the ideas of section 4 in a natural fashion. Let

$$E_{n,m}(D) := \operatorname{conv}\Big\{(a_{i_1}^{\mathrm{T}} x) \cdots (a_{i_k}^{\mathrm{T}} x)\, \psi(x)^2 : \quad m-k \text{ is even},\ \psi \in H_{n,(m-k)/2}, \\ \text{and } i_1, \ldots, i_k \in \{1, \ldots, q\}\Big\},$$

$$K^r_{n,m}(D) := \left\{\theta \in H_{n,m} : \left(\sum_{i=1}^{q} a_i^{\mathrm{T}} x\right)^r \theta(x) \in E_{n,m+r}(D)\right\},$$

and

$$C^r_{n,m}(D) := \left\{\theta \in H_{n,m} : \quad \exists \phi \in H_{n,m+r},\ \phi \geq 0,\ \text{s.t.} \\ \left(\sum_{i=1}^{q} a_i^{\mathrm{T}} x\right)^r \theta(x) = \phi(a_1^{\mathrm{T}} x, \ldots, a_q^{\mathrm{T}} x)\right\}.$$

By construction, both $K^r_{n,m}(D)$ and $C^r_{n,m}(D)$ have LMI descriptions. Indeed, we have that $C^r_{n,m}(D)$ is a polyhedral cone. Notice also that $C^r_{n,m}(D) \subseteq K^r_{n,m}(D)$.

The natural extensions of Propositions 6 and 8 hold.

Proposition 13. $C^r_{n,m}(D) \uparrow P_{n,m}(D)$ and $K^r_{n,m}(D) \uparrow P_{n,m}(D)$.

Proof. The first claim follows from Proposition 12. The second claim follows from the first one and the inclusions $C^r_{n,m}(D) \subseteq K^r_{n,m}(D) \subseteq P_{n,m}(D)$. □

Proposition 12 also yields Proposition 14, which is a natural analogue of Theorem 4. Notice that both the second part of Proposition 13 and Proposition 6 can be obtained as a consequence of Proposition 14.

PROPOSITION 14. *Assume $D = \{x : a_i^{\mathrm{T}} x \geq 0,\ i = 1, \ldots, q\}$ is pointed. Let $\theta \in H_{n,m}$. If θ is pd in D, then there exists $r \in \mathbb{N}$ such that*

$$\left(\sum_{i=1}^{n} a_i^{\mathrm{T}} x \right)^{r} \theta(x) \in E_{n,m+r}(D).$$

Proof. This readily follows from Proposition 12. □

Proof of Proposition 12. Let $A = QR$ be the full QR factorization of A (see, e.g., [5, 32]); i.e., $Q \in \mathbb{R}^{q \times q}$ is orthogonal and $R \in \mathbb{R}^{q \times n}$ is upper triangular. Since $\operatorname{rank}(A) = n$ (as D is pointed), the matrix R is of the form $\begin{bmatrix} U^{\mathrm{T}} & 0 \end{bmatrix}^{\mathrm{T}}$, where $U \in \mathbb{R}^{n \times n}$ is upper triangular and nonsingular. Put $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$, where Q_1 is the block of the first n columns of Q. Now let $\gamma \in H_{q,m}$ be defined as

$$\gamma(y) := \theta(U^{-1} Q_1^{\mathrm{T}} y).$$

Notice that $U^{-1} Q_1^{\mathrm{T}} A = I$, so in particular $\gamma(Ax) = \theta(x)$.

Claim. For $c \geq 0$ sufficiently large, $\gamma(y) + c(y^{\mathrm{T}} Q_2 Q_2^{\mathrm{T}} y) > 0$ for all $y \in \Delta_q := \{y \in \mathbb{R}_+^q : \sum y_i = 1\}$. Here is a proof of the claim: let $F := \{y \in \Delta_q : \gamma(y) \leq 0\}$. If $F = \emptyset$, then take $c = 0$. Otherwise, observe that any given $y \in F$ cannot be of the form Ax, so it is also not of the form $Q_1 z$. Therefore $Q_2^{\mathrm{T}} y \neq 0$ for all $y \in F$. Let $m_1 := \min\{\gamma(y) : y \in F\} \leq 0$, and $m_2 := \min\{(y^{\mathrm{T}} Q_2 Q_2^{\mathrm{T}} y) : y \in F\} > 0$. (These minima are attained because F is compact.) The claim then follows by taking $c > -m_1/m_2$.

Let $c > 0$ be sufficiently large so that $\gamma(y) + c(y^{\mathrm{T}} Q_2 Q_2^{\mathrm{T}} y) > 0$ for all $y \in \Delta_q$. Applying Pólya's theorem to

$$\gamma(y) + c(y_1 + \cdots + y_q)^{m-2}(y^{\mathrm{T}} Q_2 Q_2^{\mathrm{T}} y)$$

we conclude that for N sufficiently large there exists $\phi \in H_{q,m+N}$, $\phi \geq 0$, such that

$$(y_1 + \cdots + y_q)^N (\gamma(y) + c(y_1 + \cdots + y_q)^{m-2}(y^{\mathrm{T}} Q_2 Q_2^{\mathrm{T}} y)) = \phi(y_1, \ldots, y_q).$$

Thus, since $\gamma(Ax) = \theta(x)$ and $Q_2^{\mathrm{T}} A x = 0$, plugging $y = Ax$ we get

$$(a_1^{\mathrm{T}} x + \cdots + a_q^{\mathrm{T}} x)^N \theta(x) = \phi(a_1^{\mathrm{T}} x, \ldots, a_q^{\mathrm{T}} x). \qquad \square$$

6. PSD forms over pointed semialgebraic cones. Consider a domain of the form

$$D = \{x \in \mathbb{R}^n : \phi_i(x) \geq 0,\ i = 1, \ldots, q\},$$

where $\phi_i \in H_{n,m_i},\ i = 1, \ldots, q$. We shall restrict our attention to domains that are pointed; i.e., we shall assume that 0 cannot be obtained as a sum of nonzero elements of D.

Since D is semialgebraic, it is closed. Thus, D is pointed if and only if $\{0\}$ is an *exposed face* of D (see, e.g., [11, 28]), i.e., if and only if there exists a nonzero vector $a \in \mathbb{R}^n$ such that

$$D \subseteq \{x \in \mathbb{R}^n : a^{\mathrm{T}} x \geq 0\} \tag{7}$$

and

$$D \cap \{x \in \mathbb{R}^n : a^{\mathrm{T}}x = 0\} = \{0\}. \tag{8}$$

Furthermore, (7) and (8) imply that $D \cap \{x \in \mathbb{R}^n : a^{\mathrm{T}}x = 1\}$ is compact.

Assumption 1.

(a) Assume that a vector $a \in \mathbb{R}^n$ satisfying (7) and (8) is available.

(b) Assume also that $a^{\mathrm{T}}x \geq 0$ is included in the definition of D. In other words, assume $\phi_i(x) = a^{\mathrm{T}}x$ for some $i \in \{1, \ldots, q\}$. (This can be assumed without loss of generality because we can always include the redundant inequality $a^{\mathrm{T}}x \geq 0$ in the definition of D.)

We now extend the constructions in sections 4 and 5 in a natural fashion. Let

$$\begin{aligned} E_{n,m}(D) := \mathrm{conv}\{\phi_{i_1}(x)\cdots\phi_{i_k}(x)\,\psi(x)^2 : \quad & m - (m_{i_1} + \cdots + m_{i_k}) \text{ is even,} \\ & \psi \in H_{n,(m-(m_{i_1}+\cdots+m_{i_k}))/2}, \\ & \text{and } i_1, \ldots, i_k \in \{1, \ldots, q\}\}, \end{aligned} \tag{9}$$

and let

$$K^r_{n,m}(D) := \{\theta \in H_{n,m} : (a^{\mathrm{T}}x)^r\theta(x) \in E_{n,m+r}(D)\}. \tag{10}$$

The heart of our construction is the following natural extension of Proposition 14.

Proposition 15. *Assume $D = \{x \in \mathbb{R}^n : \phi_i(x) \geq 0,\ i = 1, \ldots, q\}$ is such that Assumption* 1 *holds. Let $\theta \in H_{n,m}$. If θ is pd in D, then there exists $r \in \mathbb{N}$ such that*

$$(a^{\mathrm{T}}x)^r\theta(x) \in E_{n,m+r}(D).$$

The proof of Proposition 15 relies on the following fundamental representation theorem due to Schmüdgen [29]. In the following statement, Σ_n denotes the set of polynomials in n variables that are sos.

Theorem 16 (Schmüdgen). *Let $f, h_1, \ldots, h_s$ be polynomials in n variables such that $\mathcal{D} = \{x \in \mathbb{R}^n : h_i(x) \geq 0, i = 1, \ldots, s\}$ is compact and $f(x) > 0$ for all $x \in \mathcal{D}$. Then there exist $g_\nu \in \Sigma_n$, $\nu \in \{0,1\}^s$, such that*

$$f(x) = \sum_{\nu\in\{0,1\}^s} h_1(x)^{\nu_1}\cdots h_s(x)^{\nu_s} g_\nu(x).$$

Proof. See [29, Cor. 3]. □

Proof of Proposition 15. First assume $a = e_n := \begin{bmatrix}0 & \cdots & 0 & 1\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^n$. To simplify notation, we shall let $\bar{x}$ denote a generic vector $(x_1, \ldots, x_{n-1}) \in \mathbb{R}^{n-1}$. Assume $\theta \in H_{n,m}$ is pd in D. Let $f(\bar{x}) = \theta(\bar{x}, 1)$, and let $\mathcal{D} = \{\bar{x} \in \mathbb{R}^{n-1} : (\bar{x}, 1) \in D\} = \{\bar{x} \in \mathbb{R}^{n-1} : \phi_i(\bar{x}, 1) \geq 0\}$. By Assumption 1 and since θ is pd in D, the set $\mathcal{D}$ is compact and $f(\bar{x}) > 0$ for all $\bar{x} \in \mathcal{D}$. Thus by Theorem 16 there exist $g_\nu \in \Sigma_{n-1}, \nu \in \{0,1\}^q$, such that

$$f(\bar{x}) = \sum_{\nu\in\{0,1\}^q} \phi_1(\bar{x}, 1)^{\nu_1}\cdots\phi_q(\bar{x}, 1)^{\nu_q} g_\nu(\bar{x}). \tag{11}$$

Let $m_\nu = \sum_{\nu_i=1}\deg(\phi_i)$ for each $\nu \in \{0,1\}^q$, and let $N = \max_\nu(m_\nu + \deg(g_\nu))$.

From (11) we get

$$\begin{aligned} x_n^{N-m}\theta(x) &= x_n^N f(\bar{x}/x_n) \\ &= \sum_{\nu\in\{0,1\}^q} \phi_1(\bar{x}/x_n,1)^{\nu_1}\cdots\phi_q(\bar{x}/x_n,1)^{\nu_q} g_\nu(\bar{x}/x_n)x_n^N \\ &= \sum_{\nu\in\{0,1\}^q} \phi_1(x)^{\nu_1}\cdots\phi_q(x)^{\nu_q}\breve{g}_\nu(x)x_n^{N-m_\nu-\deg(g_\nu)}, \end{aligned}$$

where $\breve{g}_\nu$ is the homogenization of g_ν. It thus follows that $\theta \in K_{n,m}^{N-m}(D)$.

The general case can be reduced to this special case via a "change of coordinates" as follows. Without loss of generality assume $a^{\mathrm{T}}a = 1$. Let $B \in \mathbb{R}^{n\times n}$ be an orthogonal matrix whose last column is a. Taking $\tilde{D} := \{y : By \in D\}$, $\tilde{\phi}_i(y) := \phi_i(By)$, $\tilde{a} := e_n$, we are in the previous case, and hence the statement above holds for $\tilde{D}$, $\tilde{\phi}_i$, $\tilde{a}$. The general result then follows for D, ϕ_i, a, after changing back to the original coordinates by taking $D = \{By : y \in \tilde{D}\}$, $\phi_i(x) := \tilde{\phi}_i(B^{\mathrm{T}}x)$, $a = B\tilde{a}$. □

We can now extend the second part of Proposition 13, which yields our most general result.

Proposition 17. $K_{n,m}^r(D) \uparrow P_{n,m}(D)$.

Proof. This follows from Proposition 15 and the following two facts:

$$\begin{aligned} \theta(x) \in E_{n,m}(D) &\Rightarrow (a^{\mathrm{T}}x)\theta(x) \in E_{n,m+1}(D), \\ (a^{\mathrm{T}}x)\theta(x) \in E_{n,m+1}(D) &\Rightarrow \theta(x) \in P_{n,m}(D). \quad □ \end{aligned}$$

Remark 2. Notice that the definitions of $E_{n,m}(\cdot)$ and $K_{n,m}^r(\cdot)$ in section 4, section 5, and section 6 are consistent; i.e., if we apply (9) and (10) to the special cases $D = \mathbb{R}_+^n = \{x : x_j \geq 0\}$, $a = e := [1 \ \cdots \ 1]^{\mathrm{T}}$ and to $D = \{x \in \mathbb{R}^n : a_i^{\mathrm{T}}x \geq 0,\ i = 1,\dots,q\}$, $a = a_1 + \cdots + a_q$, we recover the sets defined in sections 4 and 5. Indeed, Proposition 6 and the second part of Proposition 13 are special cases of Proposition 17.

Using a stronger representation theorem for positive polynomials due to Jacobi and Prestel (see [23, Thm. 6.3.4]), another sequence of inner approximations can be constructed. Assume $m_1,\dots,m_q$ have the same parity. Thus by Assumption 1(a), all $m_1,\dots,m_q$ must be odd.

Define $\mathcal{E}_{n,m}(D)$ as follows: For m odd,

$$\mathcal{E}_{n,m}(D) := \operatorname{conv}\{\phi_i(x)\psi(x)^2 : i \in \{1,\dots,q\},\ \psi \in H_{n,(m-m_i)/2}\},$$

and for m even,

$$\mathcal{E}_{n,m}(D) := \Sigma_{n,m}.$$

Finally, let

$$\mathcal{K}_{n,m}^r(D) := \{\theta \in H_{n,m} : (a^{\mathrm{T}}x)^r\theta(x) \in \mathcal{E}_{n,m+r}(D)\}.$$

Proposition 18. $\mathcal{K}_{n,m}^r(D) \uparrow P_{n,m}(D)$.

Proof. Under the assumptions made above, the representation theorem due to Jacobi and Prestel (see [23, Thm. 6.3.4(i)]) implies that for any given $\theta \in H_{n,m}$ pd in D there exists $r \in \mathbb{N}$ such that

$$(a^{\mathrm{T}}x)^r\theta(x) \in \mathcal{E}_{n,m+r}(D).$$

Now proceeding as in the proof of Proposition 17, the result follows. □

Remark 3. When the m_i's do not have the same parity, the construction above still works, provided that we take

$$\mathcal{E}_{n,m}(D) := \operatorname{conv}(F_{n,m}(D) \cup G_{n,m}(D) \cup \Sigma_{n,m}),$$

where

$$F_{n,m}(D) := \{\phi_i(x)\psi(x)^2 : i \in \{1, \dots, q\},\ m - m_i \text{ is even}, \psi \in H_{n,(m-m_i)/2}\},$$

$$G_{n,m}(D) := \{\phi_i(x)\phi_j(x)\psi(x)^2 : \begin{array}{l} i < j \in \{1, \dots, q\}, \\ m - m_i - m_j \text{ is even}, \\ \text{and } \psi \in H_{n,(m-m_i-m_j)/2}\}. \end{array}$$

In this case, [23, Thm. 6.3.4(ii)] applies.

Example 1 (constrained polynomial optimization). Let $g(x)$ and $g_i(x)$, $i = 1, \dots, q$, be given polynomials in n variables (not necessarily homogeneous) and consider the problem of finding

$$g^* \ := \ \min\{g(x) : g_i(x) \geq 0,\ i = 1, \dots, q\}. \tag{12}$$

We shall assume that the following technical condition holds:

$$\text{For all } x \in \mathbb{R}^n \setminus \{0\} \text{ there exists } i \in \{1, \dots, q\} \text{ s.t. } \tilde{g}_i(x) < 0, \tag{13}$$

where $\tilde{g}_i(x)$ is the homogeneous component of $g_i(x)$ of highest total degree.

Without loss of generality assume the constant term of $g(x)$ is zero, i.e., $g_{\vec{0}} = 0$. Thus, by homogenizing, it can be shown that (12) is equivalent to

$$\begin{array}{ll} \max & -\theta_{(\vec{0},2m)} \\ \text{s.t.} & \theta_{(\alpha,\alpha_{n+1})} = g_\alpha \text{ for all } |(\alpha, \alpha_{n+1})| = 2m,\ \alpha \neq \vec{0}, \\ & \theta \in P_{n+1,m}(D), \end{array}$$

where $D = \{(x, x_{n+1}) : x_{n+1} \geq 0 \text{ and } \breve{g}_i(x, x_{n+1}) \geq 0,\ i = 1, \dots, q\}$. It is easy to see that the domain $D \subseteq \mathbb{R}^{n+1}$ satisfies Assumption 1(a) for $a = e_{n+1}$ if and only if condition (13) holds.

For each nonnegative integer r, consider

$$\begin{array}{lll} g^r & := \max & -\theta_{(\vec{0},2m)} \\ & \quad \text{s.t.} & \theta_{(\alpha,\alpha_{n+1})} = g_\alpha \text{ for all } |(\alpha, \alpha_{n+1})| = 2m,\ \alpha \neq \vec{0}, \\ & & \theta \in K^r_{n+1,m}(D). \end{array}$$

By the construction of $K^r_{n+1,m}(D)$ above, this is a semidefinite program. Furthermore, by Proposition 17 and Theorem 2, $g^r \uparrow g^*$. This also holds if $K^r_{n+1,m}(D)$ is changed to $\mathcal{K}^r_{n+1,m}(D)$.

Remark 4. The sequence of inner approximations $\mathcal{K}^r_{n,m}(D)$ is closely related to Lasserre's construction in [15], which relies on a theorem of Putinar [24, Thm. 1.4]. However, as pointed out in [23, p. 159], the proof of Putinar's theorem works only in the case when all m_i's are even and requires the hypothesis (13), which is slightly stronger than the hypothesis made in [24] and in [15].

Acknowledgments. We thank Monique Laurent for her careful reading and comments on a preliminary version of this paper. In particular, we are indebted to her for pointing out a gap in a previous proof of Proposition 17.

REFERENCES

[1] D. Bertsimas and I. Popescu, *Optimal inequalities in probability theory: A convex optimization approach*, SIAM J. Optim., 15 (2005), pp. 780–804.

[2] J. Bochnak, M. Coste, and M-F. Roy, *Real Algebraic Geometry*, Springer-Verlag, Berlin, 1998.

[3] E. de Klerk and D. V. Pasechnik, *Approximation of the stability number of a graph via copositive programming*, SIAM J. Optim., 12 (2002), pp. 875–892.

[4] P. H. Diananda, *On nonnegative forms in real variables some or all of which are nonnegative*, Math. Proc. Cambridge Philos. Soc., 58 (1962), pp. 17–25.

[5] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[6] D. Handelman, *Representing polynomials by positive linear functions on compact convex polyhedra*, Pacific J. Math., 132 (1988), pp. 35–62.

[7] B. Hanzon and D. Jibetean, *Global minimization of a multivariate polynomial using matrix methods*, J. Global Optim., 27 (2003), pp. 1–23.

[8] G. Hardy, J. Littlewood, and G. Pólya, *Inequalities*, 2nd ed., Cambridge University Press, New York, 1988.

[9] D. Hilbert, *Uber die darstellung definiter formen als summe von formenquadraten*, Math. Ann., 32 (1888), pp. 342–350.

[10] D. Hilbert, *Mathematical problems*, Bull. Amer. Math. Soc., 8 (1902), pp. 437–479.

[11] J. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, 1993.

[12] D. Jibetean, *Algebraic Optimization with Applications to System Theory*, Ph.D. thesis, Vrije Universiteit, Amsterdam, The Netherlands, 2003.

[13] D. Jibetean and E. de Klerk, *Global optimization of rational functions: A semidefinite programming approach*, Math. Program., 106 (2006), pp. 93–109.

[14] M. Kojima, S. Kim, and H. Waki, *A general framework for convex relaxation of polynomial optimization problems over cones*, J. Oper. Res. Soc. Japan, 46 (2003), pp. 125–144.

[15] J. B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM J. Optim., 11 (2001), pp. 796–817.

[16] J. B. Lasserre, *Bounds on measures satisfying moment conditions*, Ann. Appl. Probab., 12 (2002), pp. 1114–1137.

[17] M. Laurent, *A comparison of the Sherali–Adams, Lovász–Schrijver, and Lasserre relaxations for* 0-1 *integer programming*, Math. Oper. Res., 28 (2003), pp. 470–496.

[18] M. Laurent, *Semidefinite representations for finite varieties*, Math. Program., to appear.

[19] Y. Nesterov, *Structure of Nonnegative Polynomials and Optimization Problems*, Tech. report 9749, CORE, Louvain-la-Neuve, Belgium, 1997.

[20] P. Parrilo, *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*, Ph.D. thesis, Department of Control and Dynamical Systems, California Institute of Technology, Pasadena, CA, 2000.

[21] P. Parrilo, *Semidefinite programming relaxations for semialgebraic problems*, Math. Program., 96 (2003), pp. 293–320.

[22] I. Popescu, *A semidefinite programming approach to optimal–moment bounds for convex classes of distributions*, Math. Oper. Res., 30 (2005), pp. 632–657.

[23] A. Prestel and C. N. Delzell, *Positive Polynomials: From Hilbert's* 17*th problem to Real Algebra*, Springer Monographs in Mathematics, Springer-Verlag, Berlin, 2001.

[24] M. Putinar, *Positive polynomials on compact sets*, Indiana Univ. Math. J., 42 (1993), pp. 969–984.

[25] J. Renegar, *A Mathematical View of Interior-Point Methods in Convex Optimization*, MPS/SIAM Ser. Optim. 3, SIAM, Philadelphia, 2001.

[26] B. Reznick, *Uniform denominators in Hilbert's* 17*th problem*, Math. Z., 220 (1995), pp. 75–97.

[27] B. Reznick, *Some concrete aspects of Hilbert's* 17*th problem*, in Real Algebraic Geometry and Ordered Structures, C. N. Delzell and J. J. Madden, eds., Contemp. Math. 253, AMS, Providence, RI, 2000, pp. 251–272.

[28] T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[29] K. Schmüdgen, *The k-moment problem for compact semi-algebraic sets*, Math. Ann., 289 (1991), pp. 203–206.

[30] N. Shor, *Class of global minimum bounds of polynomial functions*, Cybernetics, 23 (1987), pp. 731–734.

[31] N. Shor and P. Stetsyuk, *The use of a modification of the r-algorithm for finding the global minimum of polynomial functions*, Cybernet. Systems Anal., 33 (1997), pp. 482–497.

[32] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[33] H. Wolkowicz, R. Saigal, and L. Vandenberghe, eds., *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*, Kluwer Academic Publishers, Dordrecht, Boston, MA, 2000.

[34] L. Zuluaga and J. Peña, *A conic programming approach to generalized Tchebycheff inequalities*, Math. Oper. Res., 30 (2005), pp. 369–388.

[35] L. F. Zuluaga, *A Conic Programming Approach to Polynomial Optimization Problems: Theory and Applications*, Ph.D. thesis, The Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA, 2004.

SIAM J. OPTIM.
Vol. 16, No. 4, pp. 1092–1109

A NEW NOTION OF WEIGHTED CENTERS FOR SEMIDEFINITE PROGRAMMING*

CHEK BENG CHUA†

Abstract. The notion of weighted centers is essential in V-space interior-point algorithms for linear programming. Although there were some successes in generalizing this notion to semidefinite programming via weighted center equations, we still do not have a generalization that preserves two important properties—(1) each choice of weights uniquely determines a pair of primal-dual weighted centers, and (2) the set of all primal-dual weighted centers completely fills up the relative interior of the primal-dual feasible region. This paper presents a new notion of weighted centers for semidefinite programming that possesses both uniqueness and completeness. Furthermore, it is shown that under strict complementarity, these weighted centers converge to weighted centers of optimal faces. Finally, this convergence result is applied to homogeneous cone programming, where the central paths defined by a certain class of optimal barriers for homogeneous cones are shown to converge to analytic centers of optimal faces in the presence of strictly complementary solutions.

Key words. weighted center, semidefinite programming, homogeneous cone programming, facial structure, Cholesky decomposition

AMS subject classifications. 90C22 90C25 90C51

DOI. 10.1137/040613378

1. Introduction.

This paper presents a new generalization of the notion of weighted centers from linear programming (LP) to semidefinite programming (SDP). We consider the following primal-dual pair of SDP problems,

$$\text{(P)}\qquad \begin{array}{rl} \inf & \mathbf{C} \bullet \mathbf{X} \\ \text{subject to} & \mathbf{A}^{(i)} \bullet \mathbf{X} = \mathbf{b}_i, \quad i = 1, \dots, m, \\ & \mathbf{X} \succeq \mathbf{0}, \end{array}$$

and

$$\text{(D)}\qquad \begin{array}{rl} \sup & \mathbf{b}^T \mathbf{y} \\ \text{subject to} & \mathbf{S} = \mathbf{C} - \sum_{i=1}^{m} \mathbf{A}^{(i)} \mathbf{y}_i, \\ & \mathbf{S} \succeq \mathbf{0}, \end{array}$$

where the $\mathbf{A}^{(i)}$ and $\mathbf{C}$ are symmetric matrices, $\mathbf{b} = (\mathbf{b}_1, \dots, \mathbf{b}_m)^T$ and $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_m)^T$ are real m-vectors, $\bullet : (\mathbf{A}, \mathbf{B}) \mapsto tr \mathbf{A}^T \mathbf{B}$ is the trace inner product, and $\mathbf{X} \succeq \mathbf{0}$ means that $\mathbf{X}$ is symmetric and positive semidefinite.

The notion of weighted centers for LP is very useful in interior-point algorithms that use the V-space approach (see [10, 11]). These weighted centers can be characterized in the following two ways:

*Received by the editors August 13, 2004; accepted for publication (in revised form) October 6, 2005; published electronically February 3, 2006. This research was supported in part by a grant from the Faculty of Mathematics, University of Waterloo, and by a Discovery Grant from NSERC.
http://www.siam.org/journals/siopt/16-4/61337.html

†Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada (cbchua@math.uwaterloo.ca).

1. as minimizers of shifted, weighted logarithmic barriers
$$(\mathbf{x},\mathbf{s}) \in \mathbb{R}^n \times \mathbb{R}^n \mapsto -\sum_{i=1}^{n} w_i \log \mathbf{x}_i - \sum_{i=1}^{n} w_i \log \mathbf{s}_i + \mathbf{x}^T\mathbf{s}$$
over the primal-dual feasible region $\{(\mathbf{x},\mathbf{s}) \in \mathbb{R}^n \times \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{s} = \mathbf{c} - A^T\mathbf{y},\ \mathbf{y} \in \mathbb{R}^m,\ \mathbf{x} \geq \mathbf{0},\ \mathbf{s} \geq \mathbf{0}\}$, and
2. as solutions to weighted center equations
$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{s} = \mathbf{c} - A^T\mathbf{y} \quad \text{for some } \mathbf{y} \in \mathbb{R}^m,$$
$$\mathbf{x}\mathbf{s} = \mathbf{w}, \quad \mathbf{x} > \mathbf{0}, \quad \text{and} \quad \mathbf{s} > \mathbf{0},$$
where $\mathbf{w} = (w_1, \ldots, w_n)^T$ and $\mathbf{x}\mathbf{s}$ denotes the componentwise product of $\mathbf{x}$ and $\mathbf{s}$.

A main obstacle in generalizing weighted centers to SDP is the lack of proper weighted barriers. Nonetheless, there were some successes in generalizing weighted center equations to SDP. Monteiro and Pang [15] considered the weighted Alizadeh–Haeberly–Overton (AHO) centers, where the equation $\mathbf{XS} + \mathbf{SX} = 2\mathbf{W}$ replaces $\mathbf{x}\mathbf{s} = \mathbf{w}$. Every symmetric, positive definite matrix $\mathbf{W}$ uniquely determines a weighted AHO center. However, unlike LP, these weighted centers do not fill up the whole relative interior of the primal-dual feasible region, i.e., not every strictly feasible pair of matrices $(\mathbf{X}, \mathbf{S})$ is a pair of weighted AHO centers. Sturm and Zhang [21] considered a different generalization that is based on the Nesterov–Todd (NT) scaling point. This generalization replaces $\mathbf{x}\mathbf{s} = \mathbf{w}$ with $\Lambda(\mathbf{XS}) = \mathbf{W}$, where $\Lambda(\mathbf{XS})$ denotes the diagonal matrix with the eigenvalues of $\mathbf{XS}$ on its diagonal, and $\mathbf{W}$ is a positive, diagonal matrix. In contrast with the weighted AHO centers, these weighted NT centers completely fill up the relative interior of the primal-dual feasible region as $\mathbf{W}$ ranges over all positive, diagonal matrices but lacks uniqueness, i.e., the equations may have more than one solution for each positive, diagonal matrix $\mathbf{W}$.

We shall describe an alternative generalization of weighted centers to SDP that possesses both uniqueness and completeness. While this generalization, which is based on Cholesky factors, is similar to a generalization considered by Monteiro and Zanjácomo [17], the main difference lies in the choice of $\mathbf{W}$. In [17], $\mathbf{W}$ is required to be "close" to multiples of the identity matrix in order for the weighted center equation to have a unique solution. On the other hand, we use positive, diagonal matrices $\mathbf{W}$ to ensure uniqueness. By restricting to diagonal matrices, the weighted centers can be characterized as minimizers of certain shifted, weighted logarithmic barriers over the primal-dual feasible region. In each generalization, the collection of weighted centers does not completely fill up the relative interior of the primal-dual feasible region. This drawback can be easily rectified in our generalization by considering orthonormal similarity transformations. Thus, for the first time, we have a notion of weighted centers for SDP that possesses two useful properties—uniqueness and completeness. This lays the foundation for future extensions of V-space algorithms to SDP.

Besides having both uniqueness and completeness, these weighted centers converge to weighted centers of optimal faces under strict complementarity. This generalizes the same property of usual central paths for SDP. Similar results were shown in [12, 13] and by Prieß and Stoer [20] for notions of weighted centers defined by the maps $(\mathbf{X}, \mathbf{S}) \mapsto (\mathbf{XS} + \mathbf{SX})/2$ and $(\mathbf{X}, \mathbf{S}) \mapsto \mathbf{X}^{1/2}\mathbf{S}\mathbf{X}^{1/2}$, respectively.

Yet another reason for considering this generalization is that our weighted centers include the analytic centers defined by a certain class of optimal barriers for homogeneous cones. Consequently, we can apply the above convergence result to homogeneous cone programming (HCP). Specifically, we show that under strict complementarity, central paths defined by this class of optimal barriers converge to analytic centers of optimal faces.

This paper is organized as follows. The next section starts with some basics of SDP, including a discussion on the facial structures of positive definite cones and the notion of strict complementarity. In section 3, a generalization, based on Cholesky factors, of weighted centers to SDP is presented and a characterization of limit points of these weighted centers under strict complementarity is given. This result is applied to HCP in section 4, where it is shown that central paths defined by a certain class of optimal barriers for HCP converge to analytic centers of optimal faces in the presence of strictly complementary solutions.

Notation and conventions. Throughout this paper, we use the following notation.

The space of symmetric matrices of order n is denoted by $\mathbb{S}^n$ and the cone of symmetric, positive semidefinite (resp., positive definite) matrices of order n is denoted by $\mathbb{S}^n_+$ (resp., $\mathbb{S}^n_{++}$). If $\mathbf{X} \in \mathbb{S}^n$, then the statement $\mathbf{X} \succeq \mathbf{0}$ (resp., $\mathbf{X} \succ \mathbf{0}$) means that $\mathbf{X} \in \mathbb{S}^n_+$ (resp., $\mathbf{X} \in \mathbb{S}^n_{++}$).

For any m-by-n matrix $\mathbf{M}$ and any subsets of indices $I \subset \{1, \ldots, m\}$ and $J \subset \{1, \ldots, n\}$, the submatrix of $\mathbf{M}$ with row indices in I and column indices in J is denoted by $\mathbf{M}_{IJ}$. If $I = \{i\}$ (or $J = \{j\}$) is a singleton, we may also write i (or j) in place of $\{i\}$ (or $\{j\}$).

The identity matrix of appropriate size (in the context used) is denoted by $\mathbf{I}$. For any subset B of positive integer indices, $\mathbf{I}_B$ denotes the 0-1 diagonal matrix of appropriate size with $(\mathbf{I}_B)_{ii} = 1$ if and only if $i \in B$, and $\mathbf{I}_B^c$ denotes $\mathbf{I} - \mathbf{I}_B$.

For each $\mathbf{X} \in \mathbb{S}^n$, $\mathcal{R}(\mathbf{X})$ denotes the range space of $\mathbf{X}$ and $\mathcal{N}(\mathbf{X})$ denotes the null space of $\mathbf{X}$.

For each topological subspace S, $\operatorname{relint}(S)$ denotes the relative interior of S and $\operatorname{cl}(S)$ denotes the closure of S.

For each sequence $x_1, \ldots, x_n$ of real numbers, $\operatorname{Diag}(x_1, \ldots, x_n)$ denotes the diagonal matrix with $x_1, \ldots, x_n$ on its diagonal.

2. Optimal faces and strict complementarity of SDP.

It is well known that each face of $\mathbb{S}^n_+$ can be uniquely identified with a subspace of $\mathbb{R}^n$ as follows: F is a face of $\mathbb{S}^n_+$ if and only if $F = \{\mathbf{X} \in \mathbb{S}^n_+ : \mathcal{R}(\mathbf{X}) \subset \mathbb{V}\}$ for some linear subspace $\mathbb{V} \subset \mathbb{R}^n$. Moreover, for any face $F = \{\mathbf{X} \in \mathbb{S}^n_+ : \mathcal{R}(\mathbf{X}) \subset \mathbb{V}\}$ of $\mathbb{S}^n_+$, $\tilde{\mathbf{X}} \in \operatorname{relint}(F)$ if and only if $\mathcal{R}(\tilde{\mathbf{X}}) = \mathbb{V}$ (see [1]). Thus, matrices in the relative interior of any face of $\mathbb{S}^n_+$ are characterized by having maximal rank among all matrices in the face.

An alternative characterization, based on Cholesky factors, of the relative interior of a face shall now be given.

It is a well-known fact that every symmetric, positive definite matrix $\mathbf{X}$ has a unique Cholesky factor (i.e., a lower triangular matrix $\mathbf{L}$ with nonnegative diagonal satisfying $\mathbf{X} = \mathbf{L}\mathbf{L}^T$). When $\mathbf{X}$ is symmetric and positive semidefinite, it still has a Cholesky factor. However, the Cholesky factor may not be unique when $\mathbf{X}$ is not positive definite. The next proposition shows that we can recover uniqueness by posing an additional condition on $\mathbf{L}$.

PROPOSITION 1. *Every symmetric, positive semidefinite matrix $\mathbf{X}$ has a unique Cholesky factor $\mathbf{L}_{\mathbf{X}}$ satisfying*

$$(\mathbf{L}_{\mathbf{X}})_{ii} = 0 \implies (\mathbf{L}_{\mathbf{X}})_{ji} = 0 \; \forall j, \tag{2.1}$$

i.e., every column of $\mathbf{L}_{\mathbf{X}}$ either is a zero column or has a positive diagonal entry.

Proof. Existence. Suppose $\mathbf{X} \in \mathbb{S}^n_+$. We shall prove by induction on n that

$$\begin{aligned}&\forall \mu_k \downarrow 0,\ \forall \{\mathbf{Y}(k)\}_{k=1}^{\infty} \subset \mathbb{S}^n_{++} \text{ with } \mathbf{Y}(\infty) := \lim_{k\to\infty} \mathbf{Y}(k) \in \mathbb{S}^n_{++},\\ &\text{all limit points of } \{\mathbf{L}(k) := \mathbf{L}_{\mathbf{X}+\mu_k \mathbf{Y}(k)}\}_{k=1}^{\infty} \text{ satisfy (2.1)},\end{aligned} \tag{2.2}$$

where $\mathbf{L}_{\mathbf{X}+\mu_k\mathbf{Y}(k)}$ denotes the unique Cholesky factor of $\mathbf{X} + \mu_k \mathbf{Y}(k) \in \mathbb{S}^n_{++}$. Since the sequence $\{\mathbf{L}(k)\}$ is bounded and hence has at least one limit point, the existence of $\mathbf{L}_\mathbf{X}$ follows by taking, say, $\mu_k = 1/k$ and $\mathbf{Y}(k) \equiv \mathbf{I}$.

The case $n = 1$ is trivially true. Suppose that for some $n \geq 1$, (2.2) holds for all $\mathbf{X} \in \mathbb{S}^n_+$. Consider the case $\mathbf{X} \in \mathbb{S}^{n+1}_+$. Let $\mathbf{L}$ denote an arbitrary limit point of $\{\mathbf{L}(k)\}$. By considering a subsequence if necessary, we may assume without any loss of generality that $\lim_{t\to\infty} \mathbf{L}(k) = \mathbf{L}$. We consider two cases.

If $\mathbf{L}_{11} = 0$, then the entries in the first column and row of $\mathbf{X}$ are zeros, whence $\mathbf{L}(k)_{j1} = \sqrt{\mu_k}(\mathbf{Y}(k)_{11})^{-1/2}\mathbf{Y}(k)_{j1}$ for all $j \in \{2, \ldots, n+1\}$. Since $\mathbf{Y}(\infty) \in \mathbb{S}^n_{++}$, it follows that $(\mathbf{Y}(k)_{11})^{-1/2}\mathbf{Y}(k)_{j1} \to (\mathbf{L}_{\mathbf{Y}(\infty)})_{j1}$, whence $\mathbf{L}_{j1} = \lim_{k\to\infty} \mathbf{L}(k)_{j1} = 0$. We then apply (2.2) on the remaining columns and rows of $\mathbf{X}$ and $\mathbf{Y}(k)$ to conclude (2.2) for $\mathbf{X}$.

If $\mathbf{L}_{11} > 0$, then $\mathbf{X}_{11} > 0$ and we only need to show that $\mathbf{L}_{JJ}$ satisfy (2.1) where J denotes the set $\{2, \ldots, n+1\}$. Now

$$\begin{aligned}\mathbf{L}(k)_{JJ} &= \mathbf{L}_{\mathbf{X}_{JJ}+\mu_k\mathbf{Y}(k)_{JJ}-(\mathbf{X}_{J1}+\mu_k\mathbf{Y}(k)_{J1})(\mathbf{X}_{11}+\mu_k\mathbf{Y}(t)_{11})^{-1}(\mathbf{X}_{J1}+\mu_k\mathbf{Y}(k)_{J1})^T}\\ &= (\mathbf{X}_{11} + \mu_k\mathbf{Y}(k)_{11})^{-1/2}\mathbf{L}_{\hat{\mathbf{X}}+\mu_k\hat{\mathbf{Y}}(k)+\mu_k^2\mathbf{Z}(k)},\end{aligned}$$

where

$$\begin{aligned}\hat{\mathbf{X}} &= \mathbf{X}_{11}\mathbf{X}_{JJ} - \mathbf{X}_{J1}\mathbf{X}_{J1}^T \in \mathbb{S}^n_+,\\ \hat{\mathbf{Y}}(k) &= \mathbf{Y}(k)_{11}\mathbf{X}_{JJ} + \mathbf{X}_{11}\mathbf{Y}(k)_{JJ} - \mathbf{Y}(k)_{J1}(\mathbf{X}_{J1})^T - \mathbf{X}_{J1}(\mathbf{Y}(k)_{J1})^T, \text{ and}\\ \mathbf{Z}(k) &= \mathbf{Y}(k)_{11}\mathbf{Y}(k)_{JJ} - (\mathbf{Y}(k)_{J1})(\mathbf{Y}(k)_{J1})^T \in \mathbb{S}^n_{++}.\end{aligned}$$

Let $\hat{\mathbf{Y}}(\infty)$ denote $\lim_{k\to\infty} \hat{\mathbf{Y}}(k)$. For each $k \in \{1, 2, \ldots, \infty\}$ and each $\mathbf{v}(\neq \mathbf{0}) \in \mathbb{R}^n$,

$$\begin{aligned}&\mathbf{v}^T\big(\hat{\mathbf{Y}}(k)\big)\mathbf{v}\\ &= \mathbf{Y}(k)_{11}\mathbf{v}^T\mathbf{X}_{JJ}\mathbf{v} + \mathbf{X}_{11}\mathbf{v}^T\mathbf{Y}(k)_{JJ}\mathbf{v} - 2\big(\mathbf{v}^T\mathbf{Y}(k)_{J1}\big)\big(\mathbf{v}^T\mathbf{X}_{J1}\big)\\ &> \mathbf{Y}(k)_{11}\frac{(\mathbf{v}^T\mathbf{X}_{J1})^2}{\mathbf{X}_{11}} + \mathbf{X}_{11}\frac{(\mathbf{v}^T\mathbf{Y}(k)_{J1})^2}{\mathbf{Y}(k)_{11}} - 2\big(\mathbf{v}^T\mathbf{Y}(k)_{J1}\big)\big(\mathbf{v}^T\mathbf{X}_{J1}\big) \geq 0,\end{aligned}$$

where we have used $\hat{\mathbf{Y}}(k) \in \mathbb{S}^n_{++}$, $\mathbf{X} \in \mathbb{S}^n_+$, and $\mathbf{X}_{11} > 0$ in the strict inequality and the arithmetic-geometric mean inequality in the last inequality. Thus we may apply (2.2) to $\hat{\mathbf{X}} \in \mathbb{S}^n_+$ and $\{\hat{\mathbf{Y}}(k) + \mu_k\mathbf{Z}(k)\} \subset \mathbb{S}^n_{++}$ to deduce that $\lim_{k\to\infty} \mathbf{L}_{\hat{\mathbf{X}}+\mu_k\hat{\mathbf{Y}}(k)+\mu_k^2\mathbf{Z}(k)}$ satisfies (2.1). Consequently, $\mathbf{L}_{JJ} = \lim_{k\to\infty} \mathbf{L}(k)_{JJ}$ also satisfies (2.1).

Uniqueness. First, consider the case when $\mathbf{X}$ is a nonnegative diagonal matrix. Let B denote the set of indices of positive diagonal entries of $\mathbf{X}$. Suppose that $\mathbf{L}$ is a Cholesky factor of $\mathbf{X}$ satisfying (2.1). Since $\mathbf{X}_{ii} = 0$ for all $i \notin B$, the ith row of $\mathbf{L}$ must be a row of zeros. Thus, $\mathbf{L}_{ij} = 0$ whenever $i \notin B$ or $j \notin B$. Consequently, $\mathbf{L}_{BB}\mathbf{L}_{BB}^T = \mathbf{X}_{BB}$ is a positive, diagonal matrix. Thus, $\mathbf{L}$ is unique. Now, suppose that $\mathbf{X} \succeq 0$ is arbitrary. Suppose that $\mathbf{L}$ and $\mathbf{L}'$ are Cholesky factors of $\mathbf{X}$ satisfying (2.1). Let B be the set of indices of nonzero columns of $\mathbf{L}$. It is clear that $\mathbf{L}\mathbf{I}_B = \mathbf{L}$, and thus, $(\mathbf{L}+\mathbf{I}_B^c)\mathbf{I}_B = \mathbf{L}$. Therefore,

$$\begin{aligned}\mathbf{I}_B &= [(\mathbf{L}+\mathbf{I}_B^c)^{-1}\mathbf{L}][(\mathbf{L}+\mathbf{I}_B^c)^{-1}\mathbf{L}]^T\\ &= (\mathbf{L}+\mathbf{I}_B^c)^{-1}\mathbf{X}(\mathbf{L}+\mathbf{I}_B^c)^{-T}\\ &= [(\mathbf{L}+\mathbf{I}_B^c)^{-1}\mathbf{L}'][(\mathbf{L}+\mathbf{I}_B^c)^{-1}\mathbf{L}']^T.\end{aligned}$$

Since $((\mathbf{L}+\mathbf{I}_B^c)^{-1}\mathbf{L}')_{ii} = 0 \implies \mathbf{L}'_{ii} = 0 \implies \mathbf{L}'_{ji} = 0\ \forall j \implies ((\mathbf{L}+\mathbf{I}_B^c)^{-1}\mathbf{L}')_{ji} = 0\ \forall j$, we have that both $(\mathbf{L}+\mathbf{I}_B^c)^{-1}\mathbf{L}'$ and $(\mathbf{L}+\mathbf{I}_B^c)^{-1}\mathbf{L}$ are Cholesky factors of $\mathbf{I}_B$ satisfying (2.1). Thus, $\mathbf{L} = \mathbf{L}'$. □

In a similar way, we can prove the followinig.

PROPOSITION 2. *Every symmetric, positive semidefinite matrix* $\mathbf{X}$ *has a unique inverse Cholesky factor* $\mathbf{U}_\mathbf{X}$ *(i.e., an upper triangular matrix* $\mathbf{U}$ *with nonnegative diagonal satisfying* $\mathbf{X} = \mathbf{U}\mathbf{U}^T$*) satisfying*

$$(\mathbf{U}_\mathbf{X})_{ii} = 0 \implies (\mathbf{U}_\mathbf{X})_{ji} = 0\ \forall j. \tag{2.3}$$

Henceforth, the unique Cholesky factor of $\mathbf{X}$ that satisfies (2.1) is denoted by $\mathbf{L}_\mathbf{X}$, and the unique inverse Cholesky factor of $\mathbf{X}$ that satisfies (2.3) is denoted by $\mathbf{U}_\mathbf{X}$.

We now describe the faces of $\mathbb{S}_+^n$ based on these Cholesky factors.

Suppose that F is a face of $\mathbb{S}_+^n$ and $\tilde{\mathbf{X}} \in \operatorname{relint}(F)$ is arbitrary. From the proof of uniqueness, we see that $(\mathbf{L}_{\tilde{\mathbf{X}}}+\mathbf{I}_B^c)^{-1}\tilde{\mathbf{X}}(\mathbf{L}_{\tilde{\mathbf{X}}}+\mathbf{I}_B^c)^{-T} = \mathbf{I}_B$, where B is the set of indices of nonzero columns of $\mathbf{L}_{\tilde{\mathbf{X}}}$. Since $\mathbf{X} \mapsto (\mathbf{L}_{\tilde{\mathbf{X}}}+\mathbf{I}_B^c)^{-1}\mathbf{X}(\mathbf{L}_{\tilde{\mathbf{X}}}+\mathbf{I}_B^c)^{-T}$ is a linear automorphism of $\mathbb{S}_+^n$, it maps F to some face F' of $\mathbb{S}_+^n$ with $\mathbf{I}_B \in \operatorname{relint}(F')$. Therefore, for any $\mathbf{X} \in \mathbb{S}_+^n$, $\mathbf{X} \in F'$ if and only if $\mathcal{R}(\mathbf{X}) \subset \mathcal{R}(\mathbf{I}_B)$, which holds if and only if $(i \notin B) \vee (j \notin B) \implies \mathbf{X}_{ij} = 0$. Consequently,

$$F = \{(\mathbf{L}_{\tilde{\mathbf{X}}}+\mathbf{I}_B^c)\mathbf{X}(L_{\tilde{\mathbf{X}}}+\mathbf{I}_B^c)^T : \mathbf{X} \succeq \mathbf{0},\ (i \notin B) \vee (j \notin B) \implies \mathbf{X}_{ij} = 0\}. \tag{2.4}$$

From this representation of the face F, we deduce the following.

PROPOSITION 3. *If* F *is a face of* $\mathbb{S}_+^n$, $B = \{i : \exists \mathbf{X} \in F, (\mathbf{L}_\mathbf{X})_{ii} \neq 0\}$, *and* $\tilde{\mathbf{X}} \in F$, *then*

1. $(\mathbf{L}_{\tilde{\mathbf{X}}})_{ii} = 0\ \forall i \notin B$ *and*
2. $\tilde{\mathbf{X}} \in \operatorname{relint}(F) \iff (\mathbf{L}_{\tilde{\mathbf{X}}})_{ii} > 0\ \forall i \in B$.

Similarly, we can use inverse Cholesky factors to characterize the relative interiors of the faces of $\mathbb{S}_+^n$.

PROPOSITION 4. *If* F *is a face of* $\mathbb{S}_+^n$, $B = \{i : \exists \mathbf{X} \in F, (\mathbf{U}_\mathbf{X})_{ii} \neq 0\}$, *and* $\tilde{\mathbf{X}} \in F$, *then*

1. $(\mathbf{U}_{\tilde{\mathbf{X}}})_{ii} = 0\ \forall i \notin B$ *and*
2. $\tilde{\mathbf{X}} \in \operatorname{relint}(F) \iff (\mathbf{U}_{\tilde{\mathbf{X}}})_{ii} > 0\ \forall i \in B$.

We now turn our attention to the primal-dual SDP problems.

Let $\boldsymbol{\mathcal{A}} : \mathbb{S}^n \to \mathbb{R}^m$ denote the linear operator $\mathbf{X} \mapsto (\mathbf{A}^{(i)} \bullet \mathbf{X})_{i=1}^m$, and let $\boldsymbol{\mathcal{A}}^*$ denote its adjoint operator $\mathbf{y} \mapsto \sum_{i=1}^m \mathbf{A}^{(i)}\mathbf{y}_i$.

We assume the following Slater condition.

ASSUMPTION 5. *There are symmetric, positive definite matrices* $\mathbf{X}$ *and* $\mathbf{S}$ *satisfying* $\boldsymbol{\mathcal{A}}(\mathbf{X}) = \mathbf{b}$, *and* $\mathbf{S} = \mathbf{C} - \boldsymbol{\mathcal{A}}^*(\mathbf{y})$ *for some* $\mathbf{y} \in \mathbb{R}^m$.

This condition implies that the sets of optimal primal and dual solutions are nonempty and bounded and $\tilde{\mathbf{X}}\tilde{\mathbf{S}} = 0$ for any optimal solutions $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{S}}$. The sets of optimal primal and dual solutions are called the *primal optimal face* and the *dual optimal face* respectively, and are denoted by O_p and O_d, respectively. Let F_p and F_d

denote the minimal faces of $\mathbb{S}_+^n$ containing O_p and O_d, respectively. If we take any $\tilde{\mathbf{X}} \in \operatorname{relint}(O_p)$, then $\tilde{\mathbf{X}} \in \operatorname{relint}(F_p)$, and thus

$$O_p = \{\mathbf{X} \in \mathbb{S}_+^n : \mathcal{R}(\mathbf{X}) \subset \mathbb{V}_p,\ \boldsymbol{\mathcal{A}}(\mathbf{X}) = \mathbf{b}\},$$

where $\mathbb{V}_p$ denotes $\mathcal{R}(\tilde{\mathbf{X}})$. Similarly,

$$O_d = \{\mathbf{S} \in \mathbb{S}_+^n : \mathcal{R}(\mathbf{S}) \subset \mathbb{V}_d,\ \mathbf{S} = \mathbf{C} - \boldsymbol{\mathcal{A}}^*(\mathbf{y}),\ \mathbf{y} \in \mathbb{R}^m\},$$

where $\mathbb{V}_d$ denotes $\mathcal{R}(\tilde{\mathbf{S}})$ for any $\tilde{\mathbf{S}} \in \operatorname{relint}(O_d)$.

Let B and N denote the sets $\{i : \exists \mathbf{X} \in F_p, (\mathbf{L_X})_{ii} \neq 0\}$ and $\{i : \exists \mathbf{S} \in F_d, (\mathbf{U_S})_{ii} \neq 0\}$, respectively.

Since the sets O_p and O_d are orthogonal, we have $\mathcal{R}(\mathbf{X}) \subset \mathcal{N}(\mathbf{S})$ and $\mathcal{R}(\mathbf{S}) \subset \mathcal{N}(\mathbf{X})$ for any $(\mathbf{X}, \mathbf{S}) \in O_p \times O_d$. Thus, $\mathbb{V}_p \perp \mathbb{V}_d$. When $\mathbb{V}_p + \mathbb{V}_d = \mathbb{R}^n$, we say that each $(\tilde{\mathbf{X}}, \tilde{\mathbf{S}}) \in \operatorname{relint}(O_p) \times \operatorname{relint}(O_d)$ is a *pair of strictly complementary solutions*. In terms of the index sets B and N, the orthogonality of O_p and O_d implies $B \cap N = \emptyset$ (and thus $|B| + |N| \leq n$), and the existence of strictly complementary solutions can be characterized by $B \cup N = \{1, \ldots, n\}$, i.e., $|B| + |N| = n$. Let T denote the set $\{1, \ldots, n\} \setminus (B \cup N)$ so that $T = \emptyset$ if and only if there are strictly complementary solutions.

We end this section with a useful lemma.

Lemma 6. *If $\tilde{\mathbf{X}} \in \operatorname{relint}(O_p)$ and $\tilde{\mathbf{S}} \in \operatorname{relint}(O_d)$, then there exists a lower triangular, square matrix $\mathbf{L}(\tilde{\mathbf{X}}, \tilde{\mathbf{S}})$ with positive diagonal such that*

$$\mathbf{L}(\tilde{\mathbf{X}}, \tilde{\mathbf{S}})\tilde{\mathbf{X}}\mathbf{L}(\tilde{\mathbf{X}}, \tilde{\mathbf{S}})^T = \mathbf{I}_B$$

and

$$\mathbf{L}(\tilde{\mathbf{X}}, \tilde{\mathbf{S}})^{-T}\tilde{\mathbf{S}}\mathbf{L}(\tilde{\mathbf{X}}, \tilde{\mathbf{S}})^{-1} = \mathbf{I}_N.$$

Proof. In the proof of uniqueness for Proposition 1, we see that $\mathbf{L}^{-1}\tilde{\mathbf{X}}\mathbf{L}^{-T} = \mathbf{I}_B$, where $\mathbf{L} = \mathbf{L}_{\tilde{\mathbf{X}}} + \mathbf{I}_{N \cup T}$. From the positive semidefiniteness and complementarity of $\mathbf{L}^{-1}\tilde{\mathbf{X}}\mathbf{L}^{-T}$ and $\mathbf{L}^T\tilde{\mathbf{S}}\mathbf{L}$, we conclude that $(\mathbf{L}^T\tilde{\mathbf{S}}\mathbf{L})_{ii} = 0$ whenever $i \in B$. Thus, the ith row of $\mathbf{U}_{\mathbf{L}^T\tilde{\mathbf{S}}\mathbf{L}}$ is a zero row whenever $i \in B$. Consequently, $(\mathbf{U}^T\mathbf{L}^{-1})\tilde{\mathbf{X}}(\mathbf{U}^T\mathbf{L}^{-1})^T = \mathbf{U}^T\mathbf{I}_B\mathbf{U} = \mathbf{I}_B$, where $\mathbf{U} = \mathbf{U}_{\mathbf{L}^T\tilde{\mathbf{S}}\mathbf{L}} + \mathbf{I}_{B \cup T}$. Finally, $(\mathbf{U}^T\mathbf{L}^{-1})^{-T}\tilde{\mathbf{S}}(\mathbf{U}^T\mathbf{L}^{-1})^{-1} = \mathbf{U}^{-1}(\mathbf{L}^T\tilde{\mathbf{S}}\mathbf{L})\mathbf{U}^{-T} = \mathbf{I}_N$. □

3. Weighted centers for SDP.

One of the many existing notions of weighted centers for SDP is the weighted centers defined by the following set of equations:

$$\begin{aligned} \boldsymbol{\mathcal{A}}(\mathbf{X}) = \mathbf{b}, \quad \mathbf{S} = \mathbf{C} - \boldsymbol{\mathcal{A}}^*(\mathbf{y}) \quad \text{for some } \mathbf{y} \in \mathbb{R}^m, \\ \mathbf{L}_\mathbf{X}^T\mathbf{S}\mathbf{L}_\mathbf{X} = \mathbf{W}, \quad \mathbf{X} \succ \mathbf{0}, \quad \text{and} \quad \mathbf{S} \succ \mathbf{0}. \end{aligned} \tag{3.1}$$

Here, the symmetric matrix $\mathbf{W}$ plays the role of the weights. We recover the usual analytic centers by setting $\mathbf{W}$ to a positive multiple of $\mathbf{I}$, in which case any solution is the unique minimizer of a shifted logarithmic determinant barrier, which is strictly convex over the primal-dual feasible region.

When $\mathbf{W}$ is not a positive multiple of $\mathbf{I}$, a result of Monteiro and Zanjácomo [16], which was improved upon by Tunçel and Wolkowicz [22], states that (3.1) has locally unique solutions when $\|\mathbf{W} - \mu\mathbf{I}\|_2 < (\sqrt{3} - 1)\mu$. This result was recently extended by the author and Tunçel [5] to include all $\mathbf{W}$ satisfying $\|\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} -$

$\mu\mathbf{I}\|_2 < \sqrt{\alpha_{\min}/(2\alpha_{\max})}\mu$ for any diagonal matrix $\mathbf{D} = \operatorname{Diag}(\alpha_1, \dots, \alpha_n)$ with positive diagonal entries, where $\alpha_{\min}$ and $\alpha_{\max}$ denote $\min\{\alpha_1, \dots, \alpha_n\}$ and $\max\{\alpha_1, \dots, \alpha_n\}$, respectively. This extension includes all positive, diagonal matrices $\mathbf{W}$.

In the case when $\mathbf{W}$ is a positive, diagonal matrix, we shall further prove that (3.1) has a (globally) unique solution by showing that any solution is the unique minimizer of some shifted, weighted logarithmic barrier that is strictly convex over the primal-dual feasible region.

3.1. Weighted barriers for semidefinite cones. Fix some arbitrary positive constants $w_1, \dots, w_n$ and consider the barrier f on the cone $\mathbb{S}^n_{++}$ defined by

$$\mathbf{X} \mapsto -\sum_{i=1}^{n} w_i \log(\mathbf{L_X})^2_{ii}.$$

This is called the *weighted barrier with weights w_i*. The following proposition shows that the weighted barrier is strictly convex.

Proposition 7. *The gradient and Hessian of the weighted barrier f are respectively given by*

$$g(\mathbf{X}) = -\mathbf{L_X}^{-T}\mathbf{W}\mathbf{L_X}^{-1} \quad \textit{and} \quad H(\mathbf{X}) : \mathbf{V} \mapsto \mathbf{L_X}^{-T}\mathcal{M}^2(\mathbf{L_X}^{-1}\mathbf{V}\mathbf{L_X}^{-T})\mathbf{L_X}^{-1},$$

where $\mathbf{W} = \operatorname{Diag}(w_1, \dots, w_n)$ and $\mathcal{M}$ denotes the map on $\mathbb{S}^n$ defined by $\mathbf{V}_{ij} \mapsto \sqrt{w_{\min\{i,j\}}}\mathbf{V}_{ij}$.

Proof. For the proof, see [5, section 6]. □

As a consequence, the shifted barrier $f_+ : \mathbf{X} \mapsto f(\mathbf{X}) + \mathbf{C} \bullet \mathbf{X}$ has a unique minimizer $\mathbf{X}$ over the primal feasible region, and $\mathbf{X}$ satisfies the Karush–Kuhn–Tucker conditions

$$\boldsymbol{\mathcal{A}}(\mathbf{X}) = \mathbf{b}, \quad \mathbf{S} = \mathbf{C} - \boldsymbol{\mathcal{A}}^*(\mathbf{y}), \quad \text{and} \quad \mathbf{L}_\mathbf{X}^T\mathbf{S}\mathbf{L_X} = \mathbf{W}$$

for some $\mathbf{S} \in \mathbb{S}^n_{++}$ and some $\mathbf{y} \in \mathbb{R}^m$. The matrix $\mathbf{S}$ is also uniquely determined and can be characterized as the unique minimizer of the shifted barrier $f_+^* : \mathbf{S} \mapsto f^*(\mathbf{S}) + \bar{\mathbf{X}} \bullet \mathbf{S}$ over the dual feasible region, where

$$f^* : \mathbf{S} \mapsto -\sum_{i=1}^{n} w_i \log(\mathbf{U_S})^2_{ii}$$

is the conjugate functional of f and $\bar{\mathbf{X}}$ is an arbitrary primal feasible solution. Thus, we have proven the following.

Theorem 8. *The weighted analytic center equation* (3.1) *uniquely determines a pair of solutions* $(\mathbf{X}, \mathbf{S})$ *whenever* $\mathbf{W}$ *is a positive, diagonal matrix.*

Hence, given positive weights $w_1, \dots, w_n$, we can define the *primal-dual weighted analytic centers* either via the weighted centers equations (3.1), where $\mathbf{W}$ is the diagonal matrix $\operatorname{Diag}(w_1, \dots, w_n)$, or as minimizers of the shifted barriers $f_+ + f_+^*$ over the primal-dual feasible region.

Unfortunately, unlike the weighted centers for LP, these weighted centers do not fill up the whole relative interior of the primal-dual feasible region, i.e., not all strictly feasible solutions $(\mathbf{X}, \mathbf{S})$ are weighted centers. This drawback can be easily rectified by considering orthonormal similarity transformations on both primal and dual problems.[1]

[1] In fact, we can also use general similarity transformations $\mathtt{P} : \mathbf{X} \mapsto \mathbf{P}^{-1}\mathbf{X}\mathbf{P}$, where $\mathbf{P}$ is an invertible matrix, but orthonormal similarity transformations are sufficient for the purpose of this paper.

THEOREM 9. *For each pair of primal-dual strictly feasible solutions* $(\mathbf{X}, \mathbf{S})$*, there exists an orthogonal matrix* $\mathbf{Q}$ *such that under the orthonormal similarity transformation* $\mathcal{Q} : \mathbf{Z} \mapsto \mathbf{Q}^T\mathbf{Z}\mathbf{Q}$ *on both primal and dual problems, the resulting pair of strictly feasible solutions* $(\mathcal{Q}(\mathbf{X}), \mathcal{Q}(\mathbf{S}))$ *is a pair of weighted centers whose weights are eigenvalues of* $\mathbf{XS}$*.*

Proof. Consider a Schur decomposition $\mathbf{Q}^T\mathbf{X}\mathbf{S}\mathbf{Q} = \mathbf{L}$ of the product $\mathbf{XS}$, where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{L}$ is a lower triangular matrix with eigenvalues of $\mathbf{XS}$ on its diagonal. Under the orthonormal similarity transformation $\mathcal{Q} : \mathbf{X} \mapsto \mathbf{Q}^T\mathbf{X}\mathbf{Q}$, we see that $\mathcal{Q}(\mathbf{X})\mathcal{Q}(\mathbf{S}) = \mathbf{L}$. Thus, $\mathbf{L}^T_{\mathcal{Q}(\mathbf{X})}\mathcal{Q}(\mathbf{S})\mathbf{L}_{\mathcal{Q}(\mathbf{X})} = \mathbf{L}^{-1}_{\mathcal{Q}(\mathbf{X})}\mathbf{L}\mathbf{L}_{\mathcal{Q}(\mathbf{X})}$ is both symmetric and lower triangular, and hence diagonal. Clearly, this diagonal matrix shares the same diagonal entries with $\mathbf{L}$. □

Therefore, we can obtain a collection of weighted centers that "fills up" the whole interior of the primal-dual feasible region by generalizing the notion of weighted centers to include all primal-dual pairs $(\mathbf{X}, \mathbf{S})$ satisfying

$$\begin{gathered} \mathbf{A}^{(i)} \bullet \mathbf{X} = \mathbf{b}_i \text{for } i = 1, \ldots, m, \\ \mathbf{S} + \sum_{i=1}^{m} \mathbf{A}^{(i)}\mathbf{y}_i = \mathbf{C} \,\text{for some } \mathbf{y} \in \mathbb{R}^m, \\ \mathbf{L}^T_{\mathcal{Q}(\mathbf{X})}\mathcal{Q}(\mathbf{S})\mathbf{L}_{\mathcal{Q}(\mathbf{X})} = \mathbf{W}, \quad \mathbf{X} \succ \mathbf{0}, \quad \text{and} \quad \mathbf{S} \succ \mathbf{0}, \end{gathered} \tag{3.2}$$

for some orthonormal similarity transformation $\mathcal{Q} : \mathbf{X} \mapsto \mathbf{Q}^T\mathbf{X}\mathbf{Q}$ and some positive, diagonal matrix $\mathbf{W}$. These weighted centers can alternatively be defined as the unique minimizers of the shifted, weighted barriers

$$(\mathbf{X}, \mathbf{S}) \mapsto -\sum_{i=1}^{n} w_i \log(\mathbf{L}_{\mathbf{Q}^T\mathbf{X}\mathbf{Q}})^2_{ii} - \sum_{i=1}^{n} w_i \log(\mathbf{U}_{\mathbf{Q}^T\mathbf{S}\mathbf{Q}})^2_{ii} + \mathbf{X} \bullet \mathbf{S}$$

over the primal-dual feasible region, where $\mathbf{Q}$ ranges over all orthogonal matrices of order n and $(w_1, \ldots, w_n)^T$ ranges over all positive n-vectors.

3.2. Target map. A natural and useful consequence of weighted centers is the collection of weighted central paths $\mathcal{WCP}(\mathbf{W}, \mathbf{Q}) := \{(\mathbf{X}(\mu, \mathbf{W}, \mathbf{Q}), \mathbf{S}(\mu, \mathbf{W}, \mathbf{Q})) : \mu > 0\}$, where $(\mathbf{X}(\mu, \mathbf{W}, \mathbf{Q}), \mathbf{S}(\mu, \mathbf{W}, \mathbf{Q})$ is the solution to (3.2) with $\mathbf{W}$ replaced by $\mu\mathbf{W}$.

Since Schur decomposition is generally not unique, we may have two or more weighted central paths passing through the same pair of weighted centers. We address this ambiguity by considering only those Schur decompositions involving lower triangular matrices with diagonal entries in nonincreasing order. In another words, we consider only weighted central paths corresponding to those $\mathbf{W}$ with diagonal entries arranged in nonincreasing order.

Suppose $(\mathbf{X}, \mathbf{S}) = (\mathbf{X}(1, \mathbf{W}, \mathbf{Q}), \mathbf{S}(1, \mathbf{W}, \mathbf{Q}))$ is a pair of weighted centers on the weighted central path $\mathcal{WCP}(\mathbf{W}, \mathbf{Q})$. Consider any weighted central path $\mathcal{WCP}(\mathbf{W}', \mathbf{Q}')$ passing through $(\mathbf{X}, \mathbf{S})$, say, $(\mathbf{X}, \mathbf{S}) = (\mathbf{X}(\mu', \mathbf{W}', \mathbf{Q}'), \mathbf{S}(\mu', \mathbf{W}', \mathbf{Q}'))$ for some $\mu' > 0$. By scaling $\mathbf{W}'$ appropriately, we may assume without any loss of generality that $\mu' = 1$. Now pick an arbitrary pair of weighted centers $(\mathbf{X}(\mu, \mathbf{W}, \mathbf{Q}), \mathbf{S}(\mu, \mathbf{W}, \mathbf{Q}))$ on $\mathcal{WCP}(\mathbf{W}, \mathbf{Q})$. By definition of weighted centers,

$$\mathbf{Q}^T\mathbf{X}(1, \mathbf{W}, \mathbf{Q})\mathbf{S}(1, \mathbf{W}, \mathbf{Q})\mathbf{Q} = \mathbf{L},$$

$$\mathbf{Q}^T\mathbf{X}(\mu, \mathbf{W}, \mathbf{Q})\mathbf{S}(\mu, \mathbf{W}, \mathbf{Q})\mathbf{Q} = \mu\check{\mathbf{L}},$$

and

$$(\mathbf{Q}')^T\mathbf{X}(1,\mathbf{W}',\mathbf{Q}')\mathbf{S}(1,\mathbf{W}',\mathbf{Q}')\mathbf{Q}' = \mathbf{L}'$$

are Schur decompositions, where the diagonal entries of both $\mathbf{L}$ and $\check{\mathbf{L}}$ are precisely the diagonal entries of $\mathbf{W}$ in the same nonincreasing order, and the diagonal entries of $\mathbf{L}'$ are those of $\mathbf{W}'$ in the same order. Since

$$\mathbf{L} = \mathbf{Q}^T\mathbf{X}\mathbf{S}\mathbf{Q} = \mathbf{Q}^T(\mathbf{Q}')\mathbf{L}'(\mathbf{Q}')^T\mathbf{Q}$$

and the diagonal entries of both $\mathbf{L}$ and $\mathbf{L}'$ are in nonincreasing order, it follows from Proposition 21 (Appendix A) that the diagonal entries of $\mathbf{L}$ and $\mathbf{L}'$ (whence those of $\mathbf{W}$ and $\mathbf{W}'$) coincide, and $(\mathbf{Q}')^T\mathbf{Q}$ is a block-diagonal matrix where the size of the ith block is the multiplicity of the ith largest (distinct) eigenvalue of $\mathbf{XS}$. It then follows from Proposition 21 that

$$\check{\mathbf{L}}' := \mu^{-1}(\mathbf{Q}')^T\mathbf{X}(\mu,\mathbf{W},\mathbf{Q})\mathbf{S}(\mu,\mathbf{W},\mathbf{Q})\mathbf{Q}' = (\mathbf{Q}')^T\mathbf{Q}\check{\mathbf{L}}\mathbf{Q}^T\mathbf{Q}'$$

is a lower triangular matrix and has diagonal entries in nonincreasing order (and thus shares the same diagonal as $\check{\mathbf{L}}$). Thus

$$(\mathbf{X}(\mu,\mathbf{W},\mathbf{Q}),\mathbf{S}(\mu,\mathbf{W},\mathbf{Q})) = (\mathbf{X}(\mu,\mathbf{W}',\mathbf{Q}'),\mathbf{S}(\mu,\mathbf{W}',\mathbf{Q}')) \in \mathcal{WCP}(\mathbf{W}',\mathbf{Q}').$$

Since $(\mathbf{X}(\mu,\mathbf{W},\mathbf{Q}),\mathbf{S}(\mu,\mathbf{W},\mathbf{Q}))$ is arbitrary, we conclude that $\mathcal{WCP}(\mathbf{W},\mathbf{Q}) \subseteq \mathcal{WCP}(\mathbf{W}',\mathbf{Q}')$. Repeating the argument on any arbitrary pair of weighted centers in $\mathcal{WCP}(\mathbf{W}',\mathbf{Q}')$ shows that

$$\mathcal{WCP}(\mathbf{W},\mathbf{Q}) = \mathcal{WCP}(\mathbf{W}',\mathbf{Q}').$$

We have thus proved the next theorem.

THEOREM 10. *For each pair of primal-dual strictly feasible solutions* $(\mathbf{X},\mathbf{S})$*, there exists exactly one weighted central path* $\mathcal{WCP}(\mathbf{W},\mathbf{Q})$ *passing through* $(\mathbf{X},\mathbf{S})$*, where diagonal entries of* $\mathbf{W}$ *are in nonincreasing order .*

As a direct consequence of the above argument, we have the next theorem.

THEOREM 11. *The map*

$$(\mathbf{X},\mathbf{S}) \mapsto \mathbf{Q}\mathbf{D}\mathbf{Q}^T,$$

where $\mathbf{Q}^T\mathbf{X}\mathbf{S}\mathbf{Q} = \mathbf{L}$ *is a Schur decomposition of* $\mathbf{XS}$ *with diagonal entries of* $\mathbf{L}$ *in nonincreasing order, and* $\mathbf{D}$ *is the diagonal matrix sharing the same diagonal with* $\mathbf{L}$*, is a bijection between the set of primal-dual strictly feasible solutions and the cone* $\mathbb{S}^n_{++}$.

Proof. Using Proposition 21 as before, if

$$\mathbf{Q}^T\mathbf{X}\mathbf{S}\mathbf{Q} = \mathbf{L}$$

and

$$(\mathbf{Q}')^T\mathbf{X}\mathbf{S}\mathbf{Q}' = \mathbf{L}'$$

are two Schur decompositions of $\mathbf{XS}$ where the diagonal entries of both Schur forms $\mathbf{L}$ and $\mathbf{L}'$ are arranged in nonincreasing order, then the diagonal entries of $\mathbf{L}$ and $\mathbf{L}'$ coincide, and $(\mathbf{Q}')\mathbf{Q}$ is a block-diagonal matrix where the size of the ith block is the multiplicity of the ith largest (distinct) eigenvalue of $\mathbf{XS}$. Consequently

$$\mathbf{Q}^T\mathbf{D}\mathbf{Q} = (\mathbf{Q}')\mathbf{D}\mathbf{Q}',$$

where $\mathbf{D}$ is the diagonal matrix sharing the same diagonal with $\mathbf{L}$ and $\mathbf{L}'$. □

3.3. Weighted central paths under strict complementarity. The main result in this subsection states that every (primal) weighted central path $\{\mathbf{X}(\mu) : \mu > 0\}$ converges to weighted analytic centers of optimal faces, where $(\mathbf{X}(\mu), \mathbf{S}(\mu))$ is the solution to (3.1) with $\mathbf{W} = \mu \mathrm{Diag}(w_1, \ldots, w_n)$.

We begin by proving a result on the limit points of weighted central paths.

LEMMA 12. *All limit points of the weighted central path lie in the relative interior of the primal optimal face.*

Proof. Suppose that $\mathbf{X}$ is a limit point of the weighted central path. Clearly, from the Karush–Kuhn–Tucker conditions, $\mathbf{X} \in O_p$. So, it suffices to show that $\mathrm{rank}(\mathbf{X}) = |B|$. Let $\{\mathbf{X}(\mu_k)\}_{k=1}^{\infty}$ be a subsequence converging to $\mathbf{X}$. Since $\{\mathbf{X}(\mu_k)\}$ is bounded, so is $\{\mathbf{L}_{\mathbf{X}(\mu_k)}\}$. So, by choosing a subsequence of $\{\mathbf{X}(\mu_k)\}$ if necessary, we may assume that $\{\mathbf{L}_{\mathbf{X}(\mu_k)}\}$ converges to some lower triangular matrix $\mathbf{L}$. Clearly, $\mathbf{X} = \mathbf{L}\mathbf{L}^T$. Let $\tilde{\mathbf{X}} \in \mathrm{relint}(O_p)$ and $\tilde{\mathbf{S}} \in O_d$ be arbitrary. Now, $(\mathbf{X}(\mu_k) - \tilde{\mathbf{X}}) \bullet (\mathbf{S}(\mu_k) - \tilde{\mathbf{S}}) = 0$, $\mathbf{X}(\mu_k) \bullet \mathbf{S}(\mu_k) = \mu_k \sum_{i=1}^n w_i$, and $\tilde{\mathbf{X}} \bullet \tilde{\mathbf{S}} = 0$ imply that $\mathbf{X}(\mu_k) \bullet \tilde{\mathbf{S}} + \mathbf{S}(\mu_k) \bullet \tilde{\mathbf{X}} = \mu_k \sum_{i=1}^n w_i$. Consequently,

$$\begin{aligned} \mu_k \sum_{i=1}^n w_i &\geq \mathbf{S}(\mu_k) \bullet \tilde{\mathbf{X}} \\ &= tr[\sqrt{\mu_k}\sqrt{\mathbf{W}}(\mathbf{L}_{\mathbf{X}(\mu_k)})^{-1}\mathbf{L}_{\tilde{\mathbf{X}}}][\sqrt{\mu_k}\sqrt{\mathbf{W}}(\mathbf{L}_{\mathbf{X}(\mu_k)})^{-1}\mathbf{L}_{\tilde{\mathbf{X}}}]^T, \end{aligned}$$

from which it follows that $\sqrt{\sum_{i=1}^n w_i} \geq \sqrt{w_i}(\mathbf{L}_{\tilde{\mathbf{X}}})_{ii}/(\mathbf{L}_{\mathbf{X}(\mu_k)})_{ii}$. Since $\tilde{\mathbf{X}}$ lies in the relative interior of F_p, it follows from Proposition 3 that $(\mathbf{L}_{\tilde{\mathbf{X}}})_{ii} > 0$ for all $i \in B$. Thus,

$$\mathbf{L}_{ii} = \lim_{k\to\infty} (\mathbf{L}_{\mathbf{X}(\mu_k)})_{ii} \geq \frac{\sqrt{w_i}(\mathbf{L}_{\tilde{\mathbf{X}}})_{ii}}{\sqrt{\sum_{i=1}^n w_i}} > 0 \quad \forall i \in B.$$

This implies that $\mathrm{rank}(\mathbf{L}) \geq |B|$, and hence $\mathrm{rank}(\mathbf{X}) = |B|$. □

Under strict complementarity, the central path for an SDP problem converges to the analytic center of the optimal face (see [8, 6, 14]). We now generalize this result to the weighted central paths.

Recall from Proposition 3 that for any $\mathbf{X} \in F_p$, $\mathbf{X}$ is in the relative interior of F_p if and only if $(\mathbf{L}_\mathbf{X})_{ii} > 0\ \forall i \in B$. Thus, the functional $f_p : \mathrm{relint}(F_p) \to \mathbb{R}$ defined by

$$\mathbf{X} \mapsto -\sum_{i\in B} w_i \log(\mathbf{L}_\mathbf{X})_{ii}^2$$

induces a barrier for the primal optimal face O_p. We shall show that under strict complementarity, every limit point of the weighted central path solves

$$\min\{f_p(\mathbf{X}) : \boldsymbol{\mathcal{A}}(\mathbf{X}) = \mathbf{b},\ \mathbf{X} \in \mathrm{span}(F_p)\},$$

where $\mathrm{span}(F_p)$ denotes the smallest linear subspace containing F_p.

LEMMA 13. *If the primal-dual pair of SDP problems has strictly complementary solutions, and the subsequence $\{(\mathbf{X}(\mu_k), \mathbf{S}(\mu_k))\}$ converges to $(\mathbf{I}_B, \mathbf{I}_N)$, then*

1. $(\mathbf{L}_{\mathbf{X}(\mu_k)})_{ij} = o(1)\ \forall i \in B,\ j \neq i$, *and* $(\mathbf{U}_{\mathbf{S}(\mu_k)})_{ij} = o(1)\ \forall i \in N,\ j \neq i$, *and*
2. $(\mathbf{L}_{\mathbf{X}(\mu_k)})_{ij} = o(\sqrt{\mu_k})\ \forall i \in N,\ j \neq i$, *and* $(\mathbf{U}_{\mathbf{S}(\mu_k)})_{ij} = o(\sqrt{\mu_k})\ \forall i \in B,\ j \neq i$.

Proof. From $(\mathbf{X}(\mu_k) - \mathbf{I}_B) \bullet (\mathbf{S}(\mu_k) - \mathbf{I}_N) = 0$, it follows that $tr\mathbf{X}(\mu_k)_N + tr\mathbf{S}(\mu_k)_B = \mu_k \sum_{i=1}^n w_i$. Expanding the left-hand side gives

$$\begin{aligned}&\sum_{\substack{i\in N\\ j\le i}}(\mathbf{L}_{\mathbf{X}(\mu_k)})_{ij}^2 + \sum_{\substack{i\in B\\ j\ge i}}(\mathbf{U}_{\mathbf{S}(\mu_k)})_{ij}^2\\ &\quad = \sum_{i\in N}(\mathbf{L}_{\mathbf{X}(\mu_k)})_{ii}^2 + \sum_{i\in B}(\mathbf{U}_{\mathbf{S}(\mu_k)})_{ii}^2 + \sum_{\substack{i\in N\\ j< i}}(\mathbf{L}_{\mathbf{X}(\mu_k)})_{ij}^2 + \sum_{\substack{i\in B\\ j> i}}(\mathbf{U}_{\mathbf{S}(\mu_k)})_{ij}^2.\end{aligned}$$

From $(\mathbf{L}_{\mathbf{X}(\mu_k)})^T\mathbf{S}(\mu_k)\mathbf{L}_{\mathbf{X}(\mu_k)} = \mu_k\mathbf{W}$, we get $(\mathbf{U}_{\mathbf{S}(\mu_k)})_{ii}(\mathbf{L}_{\mathbf{X}(\mu_k)})_{ii} = \sqrt{\mu_k w_i}$. Therefore,

$$\begin{aligned}\sum_{i=1}^n w_i &= \sum_{i\in B}\frac{w_i}{(\mathbf{L}_{\mathbf{X}(\mu_k)})_{ii}^2} + \sum_{i\in N}\frac{w_i}{(\mathbf{U}_{\mathbf{S}(\mu_k)})_{ii}^2} + \frac{1}{\mu_k}\left(\sum_{\substack{i\in N\\ j<i}}(\mathbf{L}_{\mathbf{X}(\mu_k)})_{ij}^2 + \sum_{\substack{i\in B\\ j>i}}(\mathbf{U}_{\mathbf{S}(\mu_k)})_{ij}^2\right)\\ &= \sum_{i\in B}\frac{w_i}{\mathbf{X}(\mu_k)_{ii}} + \sum_{i\in N}\frac{w_i}{\mathbf{S}(\mu_k)_{ii}} + \sum_{\substack{i\in N\\ j<i}}\frac{(\mathbf{L}_{\mathbf{X}(\mu_k)})_{ij}^2}{\mu_k} + \sum_{\substack{i\in B\\ j>i}}\frac{(\mathbf{U}_{\mathbf{S}(\mu_k)})_{ij}^2}{\mu_k}\\ &\quad + \sum_{i\in B}\left(\frac{w_i}{(\mathbf{L}_{\mathbf{X}(\mu_k)})_{ii}^2} - \frac{w_i}{\mathbf{X}(\mu_k)_{ii}}\right) + \sum_{i\in N}\left(\frac{w_i}{(\mathbf{U}_{\mathbf{S}(\mu_k)})_{ii}^2} - \frac{w_i}{\mathbf{S}(\mu_k)_{ii}}\right).\end{aligned}$$

Since

$$\mathbf{S}(\mu_k)_{ii} = \sum_{j>i}((\mathbf{U}_{\mathbf{S}(\mu_k)})_{ij})^2 + ((\mathbf{U}_{\mathbf{S}(\mu_k)})_{ii})^2 \ge ((\mathbf{U}_{\mathbf{S}(\mu_k)})_{ii})^2$$

and

$$\mathbf{X}(\mu_k)_{ii} = \sum_{j<i}((\mathbf{L}_{\mathbf{X}(\mu_k)})_{ij})^2 + ((\mathbf{L}_{\mathbf{X}(\mu_k)})_{ii})^2 \ge ((\mathbf{L}_{\mathbf{X}(\mu_k)})_{ii})^2,$$

the summands in the last two sums are nonnegative. Thus, the right-hand side is at least the sum $\sum_{i\in N}\frac{w_i}{\mathbf{S}(\mu_k)_{ii}} + \sum_{i\in B}\frac{w_i}{\mathbf{X}(\mu_k)_{ii}}$, which, under strict complementarity and the assumptions $\mathbf{X}(\mu_k) \to \mathbf{I}_B$ and $\mathbf{S}(\mu_k) \to \mathbf{I}_N$, converges to the left-hand side as $k \to \infty$. This can occur only when all summands in the last four sums converge to zero. □

We now give the main theorem of this section.

THEOREM 14. *If there are strictly complementary solutions to the primal-dual SDP problems, then the weighted central path for the primal problem converges to the solution of*

$$\begin{aligned}\min\quad & -\sum_{i\in B} w_i \log(\mathbf{L}_\mathbf{X})_{ii}^2\\ s.t.\quad & \mathbf{A}^{(i)} \bullet \mathbf{X} = \mathbf{b}_i, \quad i = 1,\ldots,m,\\ & \mathbf{X} \in \operatorname{span}(F_p),\end{aligned} \tag{3.3}$$

where F_p is the minimal face of $\mathbb{S}_+^n$ containing the primal optimal face and $B = \{i : \exists\mathbf{X} \in F_p,\ (\mathbf{L}_\mathbf{X})_{ii} \neq 0\}$.

Proof. Suppose $\tilde{\mathbf{X}}$ is an arbitrary limit point of the weighted central path. By Lemma 12, $\tilde{\mathbf{X}} \in \operatorname{relint}(F_p)$. Since $\mathbf{S}(\mu)$ is bounded as $\mu \downarrow 0$, we can choose a sequence

$\{\mu_k\}$ of positive real numbers converging to zero such that $\mathbf{X}(\mu_k) \to \tilde{\mathbf{X}}$, and $\mathbf{S}(\mu_k)$ is convergent with limit $\tilde{\mathbf{S}}$. Let $\mathbf{L}$ denote the matrix $\mathbf{L}(\tilde{\mathbf{X}}, \tilde{\mathbf{S}})$ in the statement of Lemma 6. Since f_p is invariant, up to an additive constant, under the transformation $\boldsymbol{\mathcal{G}} : \mathbf{X} \mapsto \mathbf{L}\mathbf{X}\mathbf{L}^T$, the limit point $\tilde{\mathbf{X}}$ solves (3.3) if and only if $\mathbf{I}_B = \boldsymbol{\mathcal{G}}(\tilde{\mathbf{X}})$ solves

$$\min \left\{ -\sum_{i \in B} w_i \log(\mathbf{L}_{\mathbf{X}})_{ii}^2 : \boldsymbol{\mathcal{A}}(\boldsymbol{\mathcal{G}}^{-1}(\mathbf{X})) = \mathbf{b},\ \mathbf{X} \in \operatorname{span}(\boldsymbol{\mathcal{G}}(F_p)) \right\}. \tag{3.4}$$

The matrix $\mathbf{I}_B$ solves (3.4) if and only if the optimality condition

$$\nabla f_p(\mathbf{I}_B) \in \operatorname{span}(\{\boldsymbol{\mathcal{G}}^{-*}(\mathbf{A}^{(1)}), \ldots, \boldsymbol{\mathcal{G}}^{-*}(\mathbf{A}^{(m)})\}) + \operatorname{span}(\boldsymbol{\mathcal{G}}(F_p))^{\perp}$$

holds, where $\boldsymbol{\mathcal{G}}^{-*}$ denotes the adjoint of the inverse of $\boldsymbol{\mathcal{G}}$. Let $\hat{\mathbf{X}}(\mu_k)$ and $\hat{\mathbf{S}}(\mu_k)$ denote $\boldsymbol{\mathcal{G}}(\mathbf{X}(\mu_k))$ and $\boldsymbol{\mathcal{G}}^{-*}(\mathbf{S}(\mu_k))$, respectively. Let $\hat{\mathbf{A}}^{(\mathbf{i})}$ denote $\boldsymbol{\mathcal{G}}^{-*}(\mathbf{A}^{(i)})$. From the description (2.4) of faces of $\mathbb{S}^n_+$ and the assumption that $\mathbf{I}_B \in \operatorname{relint}(\boldsymbol{\mathcal{G}}(F_p))$, we deduce that $\operatorname{span}(\boldsymbol{\mathcal{G}}(F_p))^{\perp} = \{\mathbf{X} \in \mathbb{S}^n_+ : \mathbf{X}_{BB} = \mathbf{0}\}$. Thus, the optimality condition is equivalent to

$$\nabla f_p(\mathbf{I}_B)_{BB} (= \mathbf{W}_{BB}) \in \operatorname{span}(\{(\hat{\mathbf{A}}^{(\mathbf{1})})_{BB}, \ldots, (\hat{\mathbf{A}}^{(\mathbf{m})})_{BB}\}).$$

Let $\mathbb{V}$ denote the subspace $\operatorname{span}(\{(\hat{\mathbf{A}}^{(\mathbf{1})})_{BB}, \ldots, (\hat{\mathbf{A}}^{(\mathbf{m})})_{BB}\})$. Since $\hat{\mathbf{S}}(\mu_k) - \mathbf{I}_N = \boldsymbol{\mathcal{G}}^{-*}(\mathbf{S}(\mu_k) - \hat{\mathbf{S}}) \in \operatorname{span}(\{\hat{\mathbf{A}}^{(\mathbf{1})}, \ldots, \hat{\mathbf{A}}^{(\mathbf{m})}\})$, we have that $(\hat{\mathbf{S}}(\mu_k))_{BB} \in \mathbb{V}$. Dividing by μ_k gives

$$\frac{(\hat{\mathbf{S}}(\mu_k))_{BB}}{\mu_k} \in \mathbb{V}.$$

By Lemma 13, $(\mathbf{U}_{(\hat{\mathbf{S}}(\mu_k))})_{ij}/\sqrt{\mu_k} \to 0$ for all $i \in B$ and $j > i$, and $(\mathbf{L}_{(\hat{\mathbf{X}}(\mu_k))})_{ij} \to 0$ for all $i \in B$ and $j < i$. Together with $\sum_{j=1}^{i} (\mathbf{L}_{\hat{\mathbf{X}}(\mu_k)})_{ij}^2 = (\hat{\mathbf{X}}(\mu_k))_{ii} \to 1$ for all $i \in B$, it follows that $(\mathbf{L}_{\hat{\mathbf{X}}(\mu_k)})_{ii} \to 1$. Thus, we deduce from $(\mathbf{U}_{\hat{\mathbf{S}}(\mu_k)})_{ii}(\mathbf{L}_{\hat{\mathbf{X}}(\mu_k)})_{ii} = \sqrt{\mu_k w_i}$ that $(\hat{\mathbf{S}}(\mu_k))_{BB}/\mu_k \to \mathbf{W}_{BB}$. Finally, since $\mathbb{V}$ is closed, the theorem follows. ∎

4. Application to homogeneous cone programming.

In this section, we consider the following primal-dual pair of HCP problems:

$$\begin{aligned} \inf \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & (\mathbf{a}^{(i)})^T\mathbf{x} = \mathbf{b}_i \quad \text{for } i = 1, \ldots, m, \\ & \qquad \mathbf{x} \in \operatorname{cl}(K), \end{aligned}$$

and

$$\begin{aligned} \sup \quad & \mathbf{b}^T\mathbf{y} \\ \text{s.t.} \quad & \mathbf{s} = \mathbf{c} - \sum_{i=1}^{m} (\mathbf{a}^{(i)})\mathbf{y}_i, \\ & \mathbf{s} \in \operatorname{cl}(K^*), \end{aligned}$$

where K is a d-dimensional homogeneous cone (i.e., a pointed, open, convex cone whose group of automorphisms acts transitively on it), $K^* := \{\mathbf{s} : \mathbf{x}^T\mathbf{s} > 0\ \forall \mathbf{x} \in K\}$ is its dual cone, the $\mathbf{a}^{(i)}$, $\mathbf{c}$, and $\mathbf{x}$ are real d-vectors, and $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_m)^T$ and $\mathbf{y} = (\mathbf{y}_i, \ldots, \mathbf{y}_m)^T$ are real m-vectors.

As before, we assume the following Slater condition.

ASSUMPTION 15. *There exist an* $\mathbf{x} \in K$ *and an* $\mathbf{s} \in K^*$ *satisfying* $(\mathbf{a}^{(i)})^T\mathbf{x} = \mathbf{b}_i$ *for* $i = 1, \ldots, m$ *and* $\mathbf{s} = \mathbf{c} - \sum_{i=1}^m (\mathbf{a}^{(i)})\mathbf{y}_i$ *for some* $y = (\mathbf{y}_i, \ldots, \mathbf{y}_m)^T \in \mathbb{R}^m$.

It was shown by the author [3] that all homogeneous cones are SDP-representable, i.e., for each homogeneous cone K, there exists a linear map $\mathbf{M} : \mathbb{R}^d \to \mathbb{S}^n$ such that $\mathbf{x} \in K$ if and only if $\mathbf{M}(\mathbf{x}) \succ \mathbf{0}$. Thus, the primal HCP problem can be reformulated as the primal SDP problem

$$\begin{aligned} \min \quad & \mathbf{M}^{-*}(\mathbf{c}) \bullet \mathbf{X} \\ \text{s.t.} \quad & \mathbf{M}^{-*}(\mathbf{a}^{(i)}) \bullet \mathbf{X} = \mathbf{b}_i, \quad i = 1, \ldots, m, \\ & \mathbf{X} \in \mathbf{M}(\mathbb{R}^d), \\ & \mathbf{X} \succeq \mathbf{0}. \end{aligned}$$

Furthermore, it was shown by the author and Tunçel [5] that HCP problems inherit strict complementarity from the corresponding SDP formulations, i.e., a HCP problem has strictly complementary solutions if and only if any SDP reformulation has such solutions. These establish the foundation for applying Theorem 14 to HCP problems.

4.1. SDP-representability of homogeneous cones. Each d-dimensional homogeneous cone K of rank r can be associated with a T-algebra $\mathbb{A} = \bigoplus_{i,j=1}^r \mathbb{A}_{ij}$ with involution * such that K is the cone containing elements of the form $\mathbf{ll}^*$, where $\mathbf{l}$ is a lower triangular element with positive diagonal (see [23]). In fact, each $\mathbf{x} \in K$ uniquely determines a lower triangular element $\mathbf{l}$ with positive diagonal such that $\mathbf{x} = \mathbf{ll}^*$. The reader is strongly encouraged to refer to [3] and [23] for more details.

For each $(i, j) \in \{1, \ldots, r\}^2$, let n_{ij} denote the dimension of $\mathbb{A}_{ij}$ as a vector subspace of $\mathbb{A}$ and let $\mathbf{x}_{ij}$ denote the component of $\mathbf{x} \in \mathbb{A}$ in $\mathbb{A}_{ij}$. From the definition of T-algebras, we have $n_{ij} = n_{ji}$ and $n_{ii} = 1$. Also, $\sum_{i=1}^r \sum_{j=1}^i n_{ij} = d$.

Let $\mathbb{T}$ denote the subspace $\bigoplus_{1 \le j \le i \le r} \mathbb{A}_{ij}$ of lower triangular elements of $\mathbb{A}$. With each $\mathbf{x} \in \mathbb{A}$, we associate the linear operator $\mathcal{M}(\mathbf{x}) : \mathbb{T} \to \mathbb{T}$ defined by $\mathcal{M}(\mathbf{x}) : \mathbf{l} \mapsto \mathrm{Pr}_{\mathbb{T}}\, \mathbf{xl}$, where $\mathrm{Pr}_{\mathbb{T}}$ denotes the orthogonal projection onto $\mathbb{T}$ under the inner product $\langle \cdot, \cdot \rangle : (\mathbf{x}, \mathbf{y}) \mapsto tr\mathbf{xy}^*$. The author [3] proved that $\mathbf{x} \in K$ if and only if $\mathcal{M}(\mathbf{x})$ is self-adjoint and positive definite. Thus, for any choice of ordered basis $\mathfrak{B}$ for $\mathbb{T}$, the map

$$\mathbf{M}_{\mathfrak{B}} : \mathbb{R}^d \to \mathbb{S}^n : \mathbf{x} \mapsto \mathbf{M}_{\mathfrak{B}}(\mathbf{x}), \tag{4.1}$$

where $\mathbf{M}_{\mathfrak{B}}(\mathbf{x})$ is the matrix representing $\mathcal{M}(\mathbf{x})$ under $\mathfrak{B}$, is an SDP-representation of K.

Let $\mathbf{l} \in \mathbb{T}$ be arbitrary. Consider the orthogonal decomposition $\bigoplus_{j=1}^r \big(\bigoplus_{i=j}^r \mathbb{A}_{ij}\big)$ of $\mathbb{T}$ into columns. Fix an arbitrary $j \in \{1, \ldots, r\}$ and consider the restriction of $\mathcal{M}(\mathbf{l})$ to the jth column $\bigoplus_{i=j}^r \mathbb{A}_{ij}$. For each $i \in \{j, \ldots, r\}$, let $\mathfrak{B}_{ij}$ denote a basis for $\mathbb{A}_{ij}$. Since $\mathbf{ly}_{ij} = \sum_{k=i}^r \mathbf{l}_{ki}\mathbf{y}_{ij} \in \bigoplus_{k=i}^r \mathbb{A}_{kj}$ for each $\mathbf{y}_{ij} \in \mathfrak{B}_{ij}$, the operator $\mathbf{y} \mapsto \mathbf{ly}$ on $\bigoplus_{i=j}^r \mathbb{A}_{ij}$ is represented by a lower block-triangular matrix $\mathbf{L}^{(j)}$ under the ordered basis $(\mathfrak{B}_{jj}, \ldots, \mathfrak{B}_{rj})$ of $\bigoplus_{i=j}^r \mathbb{A}_{ij}$, where elements in each $\mathfrak{B}_{ij}$ are arbitrarily ordered. Furthermore, $\mathrm{Pr}_{\mathbb{A}_{ij}}\, \mathbf{ly}_{ij} = \rho_i(\mathbf{l})\mathbf{y}_{ij}$ for each $\mathbf{y}_{ij} \in \mathfrak{B}_{ij}$ implies that the $(i - j + 1)$st diagonal block in $\mathbf{L}^{(j)}$ is $\rho_i(\mathbf{l})\mathbf{I}$, where $\rho_i(\mathbf{l})$ is the value of the ith entry on the diagonal of $\mathbf{l}$. Thus, $\mathbf{L}^{(j)}$ is in fact a lower triangular matrix with n_{ij} copies of $\rho_i(\mathbf{l})$ on the diagonal for $i = \{j, \ldots, r\}$. Since for each $j \in \{1, \ldots, r\}$, $\mathcal{M}(\mathbf{l})$ maps the jth column $\bigoplus_{i=j}^r \mathbb{A}_{ij}$ into itself, it follows that the linear operator $\mathcal{M}(\mathbf{l})$ can be represented by a lower triangular matrix $\mathbf{L}$ with $\sum_{j=1}^i n_{ij}$ copies of $\rho_i(\mathbf{l})$ on the diagonal for $i = 1, \ldots, r$.

LEMMA 16. *There exists an ordered basis $\mathfrak{B}$ for $\mathbb{T}$ such that for each $\mathbf{l} \in \mathbb{T}$ with nonnegative diagonal values, the lower triangular matrix $\mathbf{M}_{\mathfrak{B}}(\mathbf{l})$ is a Cholesky factor of the matrix $\mathbf{M}_{\mathfrak{B}}(\mathbf{l}\mathbf{l}^*)$. Moreover, the matrix $\mathbf{M}_{\mathfrak{B}}(\mathbf{l})$ has $\sum_{j=1}^{i} n_{ij}$ copies of $\rho_i(\mathbf{l})$ on its diagonal.*

Proof. Let $\mathfrak{B}$ be the ordered basis $(\mathfrak{B}_{11}, \dots, \mathfrak{B}_{1r}, \mathfrak{B}_{22}, \dots, \mathfrak{B}_{2r}, \dots, \mathfrak{B}_{rr})$. It remains to show that $\mathcal{M}(\mathbf{x}) = \mathcal{M}(\mathbf{l}) \circ \mathcal{M}(\mathbf{l})^*$. This is a special case of Proposition 3.4(iii) of [3]. □

Henceforth, we shall use the ordered basis in the lemma to define the SDP representation in (4.1), and drop the subscript $\mathfrak{B}$.

4.2. Optimal faces and strict complementarity of HCP. In this subsection, we extend some results in section 2 to the optimal faces of HCP problems. These extensions rely heavily on the appropriate choice of the ordered basis $\mathfrak{B}$ in Lemma 16.

LEMMA 17. *Each $\mathbf{x} \in cl(K)$ has a unique Cholesky factor $\mathbf{l}_\mathbf{x}$ (i.e., a lower triangular element $\mathbf{l}_\mathbf{x}$ with nonnegative diagonal values such that $\mathbf{x} = \mathbf{l}\mathbf{l}^*$) satisfying*

$$\rho_i(\mathbf{l}_\mathbf{x}) = 0 \implies (\mathbf{l}_\mathbf{x})_{ji} = 0. \tag{4.2}$$

Proof. Suppose that $\mathbf{x} \in cl(K)$. Therefore $\mathbf{M}(\mathbf{x})$ is symmetric and positive semidefinite. From the proof of existence of Proposition 1, we see that $\mathbf{L}_{\mathbf{M}(\mathbf{x})+\mu\mathbf{I}} \to \mathbf{L}_{\mathbf{M}(\mathbf{x})}$ as $\mu \to 0$. Since $\mathbf{M}(\mathbf{x}) + \mu\mathbf{I} = \mathbf{M}(\mathbf{x} + \mu\mathbf{e})$, where $\mathbf{e}$ is the unit of the T-algebra $\mathbb{A}$, it follows from Lemma 16 that for each positive μ, $\mathbf{L}_{\mathbf{M}(\mathbf{x})+\mu\mathbf{I}} = \mathbf{M}(\mathbf{l}_{\mathbf{x}+\mu\mathbf{e}})$. Consequently, $\mathbf{L}_{\mathbf{M}(\mathbf{x})} = \mathbf{M}(\mathbf{l}_\mathbf{x})$, where $\mathbf{l}_\mathbf{x} \in \mathbb{T}$ is any limit point of $\{\mathbf{l}_{\mathbf{x}+\mu\mathbf{e}}\}_{\mu>0}$. The limit point $\mathbf{l}_\mathbf{x}$ is clearly a Cholesky factor of $\mathbf{x}$. Property (4.2) for $\mathbf{l}_\mathbf{x}$ can be deduced from the same property of $\mathbf{L}_{\mathbf{M}(\mathbf{x})}$ in Proposition 1 and the choice of $\mathfrak{B}$ in Lemma 16. Finally, the uniqueness of $\mathbf{l}_\mathbf{x}$ follows straightforwardly from the choice of $\mathfrak{B}$ in Lemma 16 and the uniqueness of $\mathbf{L}_{\mathbf{M}(\mathbf{x})}$. □

PROPOSITION 18. *If F is a face of K, $B = \{i : \exists \mathbf{x} \in F, \rho_i(\mathbf{l}_\mathbf{x}) \neq 0\}$ and $\tilde{\mathbf{x}} \in F$, then*

1. $\rho_i(\mathbf{l}_{\tilde{\mathbf{x}}}) = 0 \; \forall i \notin B$ *and*
2. $\tilde{\mathbf{x}} \in \operatorname{relint}(F) \iff \rho_i(\mathbf{l}_{\tilde{\mathbf{x}}}) > 0 \; \forall i \in B$.

Proof. If F is a face of K, then there exists some face F' of $\mathbb{S}^n_{++}$ such that $\mathbf{M}(F) = \mathbf{M}(\mathbb{R}^d) \cap F'$. Thus, using the description (2.4) of F' we may describe F as

$$F = \{((\mathbf{l}_{\tilde{\tilde{\mathbf{x}}}} + \mathbf{e}_B^c)\mathbf{l}_\mathbf{x})(\mathbf{l}_\mathbf{x}^*(\mathbf{l}_{\tilde{\tilde{\mathbf{x}}}} + \mathbf{e}_B^c)^*) : \mathbf{x} \in cl(K),\ (i \notin B) \vee (j \notin B) \implies \mathbf{x}_{ij} = 0\},$$

where $\tilde{\tilde{\mathbf{x}}} \in \operatorname{relint}(F)$ is arbitrary and $\mathbf{e}_B^c$ denotes the diagonal element of $\mathbb{A}$ with 0-1 diagonal such that $\rho_i(\mathbf{e}_B^c) = 1$ if and only if $i \notin B$. The theorem then follows from this description. □

Since every HCP problem can be reformulated as an SDP problem, we may naturally generalize the notion of strict complementarity from SDP to HCP. However, in order for this generalization to be well defined, different SDP reformulations of the same HCP problem should not result in different conclusions on the existence of strictly complementary solutions. Indeed, the author and Tunçel [5] showed that the existence of strictly complementary solutions is independent of the SDP formulation used. Furthermore, this notion of strictly complementary solutions coincides with a more general notion introduced by Pataki [18], which was shown to be a generic property of linear optimization problems over convex cones by Pataki and Tunçel [19].

4.3. Limit points of central paths for HCP. By reformulating HCP problems as SDP problems, any algorithm for SDP translates directly to an algorithm for

HCP. However, from the perspective of theoretical complexity, it is advantageous for algorithms to use optimal barriers for homogeneous cones. In this subsection, we consider a certain class of optimal barriers for homogeneous cones, and we characterize the limit points of the central paths defined by this class of optimal barriers under strict complementarity.

Since each $\mathbf{x} \in K$ uniquely determines a lower triangular element $\mathbf{l_x}$ with positive diagonal such that $\mathbf{x} = \mathbf{l_x l_x^*}$, the functional $f : K \mapsto \mathbb{R}$ defined by $f : \mathbf{x} \mapsto -\sum_{i=1}^{r} \log \rho_i(\mathbf{l_x})^2$ is well defined. Furthermore, it is an r-logarithmically homogeneous, self-concordant barrier for K (see [2]). In fact, we know from a result of Güler and Tunçel [9] that it is optimal for K. We shall now relate this barrier with a weighted barrier of the SDP representation given by (4.1).

For each i, let $J(i)$ denote the set of the indices of the $\bar{n}_i := \sum_{j=1}^{i} n_{ij}$ copies of $\rho_i(\mathbf{l_x})$ on the diagonal of $\mathbf{L} = \mathbf{M}(\mathbf{l})$, i.e., $\mathbf{L}_{jj} = \rho_i(\mathbf{l_x})$ for all $i \in \{1, \dots, r\}$, all $j \in J(i)$, and all $\mathbf{x} \in K$. Since $\{J(i)\}_{i=1}^{r}$ is a partition of $\{1, \dots, n\}$, where $n := \sum_{1 \leq j \leq i \leq r} n_{ij}$, we may define a map $\pi : \{1, \dots, n\} \to \{1, \dots, r\}$ such that $j \in J(\pi(j))$ for all $j \in \{1, \dots, n\}$. For each $i \in \{1, \dots, r\}$,

$$\log \rho_i(\mathbf{l_x})^2 = \frac{1}{\bar{n}_i} \bar{n}_i \log \rho_i(\mathbf{l_x})^2 = \sum_{j \in J(i)} \frac{1}{\bar{n}_{\pi(j)}} \log(\mathbf{L}_{\mathbf{M}(\mathbf{x})})_{jj}^2, \tag{4.3}$$

from which we deduce that the optimal barrier

$$f(\mathbf{x}) = -\sum_{i=1}^{r} \log \rho_i(\mathbf{l_x})^2 = -\sum_{i=1}^{n} \bar{n}_{\pi(i)}^{-1} \log(\mathbf{L}_{\mathbf{M}(\mathbf{x})})_{ii}^2$$

coincides with the restriction of the weighted barrier for the SDP representation with weights $\bar{n}_{\pi(1)}^{-1}, \dots, \bar{n}_{\pi(n)}^{-1}$. Consequently, as a corollary to Theorem 14, we have the following.

COROLLARY 19. *If a pair of primal-dual HCP problems has strictly complementary solutions, then the central path converges to the solution of*

$$\begin{aligned} \min \quad & -\sum_{i \in B} \log \rho_i(\mathbf{l_x})^2 \\ \textit{s.t.} \quad & (\mathbf{a}^{(i)})^T \mathbf{x} = \mathbf{b}_i, \quad i = 1, \dots, m, \\ & \mathbf{x} \in \operatorname{span}(F_p), \end{aligned}$$

where F_p is the minimal face of K containing the primal optimal face and $B = \{i : \exists \mathbf{x} \in F_p,\ \rho_i(\mathbf{l_x}) \neq 0\}$.

Proof. Since the image of the central path under $\mathbf{M}$ is the path defined by the weighted barrier $\mathbf{X} \mapsto -\sum_{i=1}^{n} \log \bar{n}_{\pi(i)}^{-1} (\mathbf{L_X})^2$ for the SDP representation (4.1), it follows from Theorem 14 that when the HCP problem has strictly complementary solutions, the image of the central path under $\mathbf{M}$ converges to the solution of

$$\begin{aligned} \min \quad & -\sum_{i \in J(B)} \bar{n}_{\pi(i)}^{-1} \log(\mathbf{L_X})_{ii}^2 \\ \text{s.t.} \quad & \mathbf{M}^{-*}(\mathbf{a}^{(i)}) \bullet \mathbf{X} = \mathbf{b}_i, \quad i = 1, \dots, m, \\ & \mathbf{X} \in \mathbf{M}(\mathbb{R}^d), \\ & \mathbf{X} \in \operatorname{span}(F_p'), \end{aligned}$$

where $J(B)$ denotes $\cup_{i \in B} J(i)$ and F_p' is the face of $\mathbb{S}_{++}^n$ such that $F_p' \cap \mathbf{M}(\mathbb{R}^d) = F_p$. The theorem then follows from (4.3). □

5. Conclusion.

We end this paper with some open questions and directions for future research.

1. Since the notion of weighted centers introduced in this paper possesses both uniqueness and completeness, we may use them for future development of V-space algorithms for SDP. One approach is to consider the target map $(\mathbf{X}, \mathbf{S}) \mapsto \mathbf{QDQ}^T$, where $\mathbf{Q}^T\mathbf{XSQ} = \mathbf{L}$ is a Schur decomposition of $\mathbf{XS}$ with diagonal entries of $\mathbf{L}$ arranged in nonincreasing order and $\mathbf{D}$ is a diagonal matrix that shares the same diagonal entries with $\mathbf{L}$, which is a bijection between the primal-dual strictly feasible region and the cone $\mathbb{S}^n_{++}$. Another approach would be to linearly transform the primal-dual problems via the orthonormal similarity transformation $\boldsymbol{\mathcal{Q}} : \mathbf{X} \mapsto \mathbf{Q}^T\mathbf{XQ}$ so that $\mathbf{L}^T_{\boldsymbol{\mathcal{Q}}(\mathbf{X})}\boldsymbol{\mathcal{Q}}(\mathbf{S})\mathbf{L}_{\boldsymbol{\mathcal{Q}}(\mathbf{X})}$ is diagonal and use the locally injective map $(\mathbf{X}, \mathbf{S}) \mapsto \mathbf{L}^T_{\mathbf{X}}\mathbf{SL}_{\mathbf{X}}$ as the V-space map.
2. The limit points of weighted centers for SDP were characterized in this paper only under strict complementarity. In the absence of strict complementarity, the limit point of the usual central path can be characterized either as the analytic center of a certain subset of the optimal face (see [6]) or as the unique minimizer of the logarithmic determinant barrier for the optimal face with an additional term (see [7]). Future extensions of these results to weighted central paths would complete the characterization of their limit points.
3. By treating central paths for HCP problems as weighted central paths for the SDP reformulations, any V-space algorithm that follows weighted central paths naturally translates to a primal-dual algorithm that follows central paths of HCP problems. However, without exploiting the structure of homogeneous cones in the analysis of the algorithm, its theoretical complexity will generally be no better than algorithms that follow the usual central path of the SDP reformulation. Thus, some nontrivial work is needed to improve the analysis of these V-space algorithms for HCP.
4. In [12, 13, 20], the analyticity of various notions of weighted central paths were studied. In [4], we investigate the analyticity of the weighted central paths introduced in this paper.

Appendix A. Technical results.

LEMMA 20. *If* $\mathbf{L}$ *is a real, lower triangular, diagonalizable matrix with nonincreasing diagonal entries, and*

$$\mathbf{P}^{-1}\mathbf{LP} = \mathbf{D}$$

is a diagonalization of $\mathbf{L}$ *where* $\mathbf{D}$ *has nonincreasing diagonal entries, then* $\mathbf{P}$ *is lower block-diagonal where the size of the* k*th block on the diagonal is the multiplicity of the* k*th largest diagonal entry of* $\mathbf{L}$*.*

Proof. We shall prove by induction on the number of distinct diagonal entries of $\mathbf{L}$, which is the number of distinct eigenvalues of $\mathbf{L}$.

When $\mathbf{L}$ has only one distinct eigenvalue, the lemma is trivial.

Suppose that the lemma is true whenever $\mathbf{L}$ has at most p distinct eigenvalues. Consider the case where $\mathbf{L}$ has $p + 1$ distinct eigenvalues. Let m denote the multiplicity of its largest eigenvalue $\lambda_{\max}$. We write all matrices in the 2-by-2 block form $\mathbf{M} = [\begin{smallmatrix}\mathbf{M}_{11} & \mathbf{M}_{12}\\ \mathbf{M}_{21} & \mathbf{M}_{22}\end{smallmatrix}]$, where $\mathbf{M}_{11}$ is m-by-m. The diagonals of $\mathbf{L}$ and $\mathbf{D}$ (which contain eigenvalues of the similar matrices $\mathbf{L}$ and $\mathbf{D}$) coincide since they are arranged in nonincreasing order. Now

$$\begin{bmatrix}\mathbf{L}_{11} & \\ \mathbf{L}_{21} & \mathbf{L}_{22}\end{bmatrix}\begin{bmatrix}\mathbf{P}_{12}\\ \mathbf{P}_{22}\end{bmatrix} = \begin{bmatrix}\mathbf{P}_{12}\\ \mathbf{P}_{22}\end{bmatrix}\mathbf{D}_{22} \implies \mathbf{L}_{11}\mathbf{P}_{12} = \mathbf{P}_{12}\mathbf{D}_{22}$$

implies that each nonzero column of $\mathbf{P}_{12}$ is an eigenvector of $\mathbf{L}_{11}$ whose associated eigenvalue is a diagonal entry of $\mathbf{D}_{22}$. Since the only eigenvalue of $\mathbf{L}_{11}$ is $\lambda_{\max}$ and all diagonal entries of $\mathbf{D}_{22}$ are strictly less than $\lambda_{\max}$, it follows that $\mathbf{P}_{12}$ is a zero matrix. Consequently, it follows from

$$\begin{bmatrix}\mathbf{P}_{11} & \\ \mathbf{P}_{21} & \mathbf{P}_{22}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{L}_{11} & \\ \mathbf{L}_{21} & \mathbf{L}_{22}\end{bmatrix}\begin{bmatrix}\mathbf{P}_{11} & \\ \mathbf{P}_{21} & \mathbf{P}_{22}\end{bmatrix}=\begin{bmatrix}\mathbf{D}_{11} & \\ & \mathbf{D}_{22}\end{bmatrix}$$

that $\mathbf{L}_{22}=\mathbf{P}_{22}\mathbf{D}_{22}\mathbf{P}_{22}^{-1}$ is diagonalizable. Since $\mathbf{L}_{22}$ has p distinct eigenvalues, we may apply the induction hypothesis to conclude that $\mathbf{P}_{22}$ is lower block-diagonal where the size of the kth block on the diagonal is the multiplicity of the kth largest diagonal entry of $\mathbf{L}_{22}$. ◻

PROPOSITION 21. *Suppose* $\mathbf{L}$ *is a real, lower triangular, diagonalizable matrix with nonincreasing diagonal entries and* $\mathbf{Q}$ *is an orthogonal matrix of the same size as* $\mathbf{L}$*. Then* $\mathbf{Q}^T\mathbf{L}\mathbf{Q}$ *is a lower triangular matrix with nonincreasing diagonal entries if and only if* $\mathbf{Q}$ *is block-diagonal where the size of the* k*th block is the multiplicity of the* k*th largest diagonal entry of* $\mathbf{L}$.

Proof. "only if": Suppose $\mathbf{Q}^T\mathbf{L}\mathbf{Q}$ is lower triangular with nonincreasing diagonal entries. Let $\mathbf{P}^{-1}(\mathbf{Q}^T\mathbf{L}\mathbf{Q})\mathbf{P}=\mathbf{D}$ be a diagonalization of $\mathbf{Q}^T\mathbf{L}\mathbf{Q}$ where $\mathbf{D}$ has nonincreasing diagonal entries, and hence shares the same diagonal as $\mathbf{L}$. It follows from the preceding lemma that $\mathbf{P}$ is lower block-diagonal where the size of the kth block is the multiplicity of the kth largest diagonal entry of $\mathbf{L}$. Since

$$(\mathbf{QP})^{-1}\mathbf{L}(\mathbf{QP})=\mathbf{D}$$

is a diagonalization of $\mathbf{L}$, we conclude, using the preceding lemma, that $\mathbf{QP}$, whence $\mathbf{Q}$, is lower block-diagonal where the size of the kth block is the multiplicity of the kth largest diagonal entry of $\mathbf{L}$.

"if": Suppose $\mathbf{Q}$ is block-diagonal where the size of the kth block is the multiplicity of the kth largest diagonal entry of $\mathbf{L}$. Let $\mathbf{P}^{-1}\mathbf{L}\mathbf{P}=\mathbf{D}$ be a diagonalization of $\mathbf{L}$ where $\mathbf{D}$ has nonincreasing diagonal entries, and hence shares the same diagonal as $\mathbf{L}$. By the preceding lemma, $\mathbf{P}$, whence $\mathbf{Q}^T\mathbf{P}$, is lower block-diagonal. Thus $\mathbf{Q}^T\mathbf{L}\mathbf{Q}=(\mathbf{Q}^T\mathbf{P})\mathbf{D}(\mathbf{Q}^T\mathbf{P})^{-1}$ is lower block-diagonal. Since the kth diagonal block of $\mathbf{D}$ is a multiple of the identity matrix, so is the kth diagonal block of $(\mathbf{Q}^T\mathbf{P})\mathbf{D}(\mathbf{Q}^T\mathbf{P})^{-1}$. Consequently $\mathbf{Q}^T\mathbf{L}\mathbf{Q}=(\mathbf{Q}^T\mathbf{P})\mathbf{D}(\mathbf{Q}^T\mathbf{P})^{-1}$ is actually lower triangular and shares the same diagonal as $\mathbf{D}$. ◻

Acknowledgment. The author thanks an anonymous referee for pointing out the related works [12, 13, 20] and for a suggestion that lead to a simpler existence proof for Proposition 1.

REFERENCES

[1] G. P. BARKER AND D. CARLSON, *Cones of diagonally dominant matrices*, Pacific J. Math., 57 (1975), pp. 15–32.

[2] C. B. CHUA, *An Algebraic Perspective on Homogeneous Cone Programming, and the Primal-Dual Second Order Cone Approximations Algorithm for Symmetric Cone Programming*, Ph.D. thesis, Cornell University, Ithaca, NY, 2003.

[3] C. B. CHUA, *Relating homogeneous cones and positive definite cones via T-algebras*, SIAM J. Optim., 14 (2003), pp. 500–506.

[4] C. B. CHUA, *Analyticity of Weighted Central Path and Error Bound for Semidefinite Programming*, CORR 2005-15, Department of Combinatorics and Optimization, Faculty of Mathematics, University of Waterloo, Canada, 2005.

[5] C. B. Chua and L. Tunçel, *Invariance and Efficiency of Convex Representations*, Research Report CORR 2004-18, Department of Combinatorics and Optimization, Faculty of Mathematics, University of Waterloo, Canada, 2004.

[6] E. de Klerk, M. Halická, and C. Roos, *Limiting behavior of the central path in semidefinite optimization*, Optim. Methods Softw., 20 (2005), pp. 99–113.

[7] A. Forsgren and G. Sporre, *Characterization of the Limit Point of the Central Path in Semidefinite Programming*, Report trita-mat-2002-os12, Department of Mathematics, Royal Institute of Technology, Stockholm, Sweden, 2002.

[8] D. Goldfarb and K. Scheinberg, *Interior point trajectories in semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 871–886.

[9] O. Güler and L. Tunçel, *Characterization of the barrier parameter of homogeneous convex cones*, Math. Program., 81 (1998), pp. 55–76.

[10] B. Jansen, C. Roos, T. Terlaky, and J.-P. Vial, *Primal-dual target-following algorithms for linear programming*, Ann. Oper. Res., 62 (1996), pp. 197–231.

[11] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise, *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems*, Lecture Notes in Comput. Sci. 538, Springer-Verlag, Berlin, Heidelberg, New York, 1991.

[12] Z. Lu and R. D. C. Monteiro, *Limiting Behavior of the Alizadeh-Haeberly-Overton Weighted Paths in Semidefinite Programming*, working paper, School of ISyE, Georgia Tech, 2003.

[13] Z. Lu and R. D. C. Monteiro, *Error bounds and limiting behaviour of weighted paths associated with the SDP map $X^{1/2}SX^{1/2}$*, SIAM J. Optim., 15 (2004), pp. 348–374.

[14] Z.-Q. Luo, J. F. Sturm, and S. Zhang, *Superlinear convergence of a symmetric primal-dual path following algorithm for semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 59–81.

[15] R. D. C. Monteiro and J.-S. Pang, *On two interior-point mappings for nonlinear semidefinite complementarity problems*, Math. Oper. Res., 23 (1998), pp. 39–60.

[16] R. D. C. Monteiro and P. Zanjácomo, *Implementation of primal-dual methods for semidefinite programming based on Monteiro and Tsuchiya Newton directions and their variants*, Optim. Methods Softw., 11/12 (1999), pp. 91–140.

[17] R. D. C. Monteiro and P. Zanjácomo, *General interior-point maps and existence of weighted paths for nonlinear semidefinite complementarity problems*, Math. Oper. Res., 25 (2000), pp. 382–399.

[18] G. Pataki, *Cone LP's and semidefinite programs: Goemetry and a simplex type method*, in Proceedings of the Fifth IPCO Conference, W. H. Cunningham, S. T. McCormick, and M. Queyranne, eds., Springer-Verlag, New York, 1996, pp. 161–174.

[19] G. Pataki and L. Tunçel, *On the generic properties of convex optimization problems in conic form*, Math. Program., 89 (2001), pp. 449–457.

[20] M. Preiß and J. Stoer, *Analysis of infeasible-interior-point paths arising with semidefinite linear complementarity problems*, Math. Program., 99 (2004), pp. 499–520.

[21] J. F. Sturm and S. Zhang, *On weighted centers for semidefinite programming*, Eur. J. Oper. Res., 126 (2000), pp. 391–407.

[22] L. Tunçel and H. Wolkowicz, *Strengthened existence and uniqueness conditions for search directions in semidefinite programming*, Linear Algebra Appl., 400 (2005), pp. 31–60.

[23] E. B. Vinberg, *The theory of convex homogeneous cones*, Tr. Mosk. Mat. Obs., 12 (1963), pp. 303–358.

SIAM J. OPTIM.
Vol. 16, No. 4, pp. 1110–1136

A FULL-NEWTON STEP $O(n)$ INFEASIBLE INTERIOR-POINT ALGORITHM FOR LINEAR OPTIMIZATION*

C. ROOS†

Abstract. We present a primal-dual infeasible interior-point algorithm. As usual, the algorithm decreases the duality gap and the feasibility residuals at the same rate. Assuming that an optimal solution exists, it is shown that at most $O(n)$ iterations suffice to reduce the duality gap and the residuals by the factor $1/e$. This implies an $O(n \log(n/\varepsilon))$ iteration bound for getting an ε-solution of the problem at hand, which coincides with the best known bound for infeasible interior-point algorithms. The algorithm constructs strictly feasible iterates for a sequence of perturbations of the given problem and its dual problem. A special feature of the algorithm is that it uses only full-Newton steps. Two types of full-Newton steps are used, so-called feasibility steps and usual (centering) steps. Starting at strictly feasible iterates of a perturbed pair, (very) close to its central path, feasibility steps serve to generate strictly feasible iterates for the next perturbed pair. By accomplishing a few centering steps for the new perturbed pair we obtain strictly feasible iterates close enough to the central path of the new perturbed pair. The algorithm finds an optimal solution or detects infeasibility or unboundedness of the given problem.

Key words. linear optimization, interior-point method, infeasible method, primal-dual method, polynomial complexity

AMS subject classifications. 90C05, 90C51

DOI. 10.1137/050623917

1. Introduction.

Interior-point methods (IPMs) for solving linear optimization (LO) problems were initiated by Karmarkar [8]. They not only have polynomial complexity but are also highly efficient in practice. One may distinguish between *feasible* IPMs and *infeasible* IPMs (IIPMs). Feasible IPMs start with a strictly feasible interior point and maintain feasibility during the solution process. An elegant and theoretically sound method to find a strictly feasible starting point is to use a self-dual embedding model, by introducing artificial variables. This technique was presented first by Ye, Todd, and Mizuno [33]. Subsequent references are [1, 20, 31]. Well-known commercial software packages are based on this approach; for example, MOSEK [2][1] and SeDuMi [23][2] are based on the use of the self-dual model. Also, the leading commercial linear optimization package CPLEX[3] includes the self-dual embedding model as a possible option.

Most of the existing software packages use an IIPM. IIPMs start with an arbitrary positive point, and feasibility is reached as optimality is approached. The first IIPMs were proposed by Lustig [11] and Tanabe [24]. Global convergence was shown by Kojima, Megiddo, and Mizuno [9], whereas Zhang [34] proved an $O(n^2L)$ iteration bound for IIPMs under certain conditions. Mizuno [12] introduced a primal-dual IIPM and proved global convergence of the algorithm. Other relevant references are [4, 5, 6, 10, 13, 15, 16, 18, 19, 22, 25, 28, 29]. A detailed discussion and analysis of

*Received by the editors February 7, 2005; accepted for publication (in revised form) November 3, 2005; published electronically February 15, 2006.
http://www.siam.org/journals/siopt/16-4/62391.html

†Faculty of Electrical Engineering, Computer Science and Mathematics, Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands (C.Roos@ewi.tudelft.nl).

[1] MOSEK is available from http://www.mosek.com.

[2] SeDuMi is available from http://sedumi.mcmaster.ca.

[3] CPLEX is available from http://cplex.com.

IIPMs can be found in the book by Wright [30] and, with less detail, in the books by Ye [32] and Vanderbei [26]. The performance of existing IIPMs highly depends on the choice of the starting point, which makes these methods less robust than the methods that use the self-dual embedding technique.

As usual, we consider the LO problem in the standard form

$$\text{(P)} \qquad \min\left\{c^T x : Ax = b, \quad x \geq 0\right\},$$

with its dual problem

$$\text{(D)} \qquad \max\left\{b^T y : A^T y + s = c, \quad s \geq 0\right\}.$$

Here $A \in \mathbf{R}^{m\times n}$, b, $y \in \mathbf{R}^m$, and c, x, $s \in \mathbf{R}^n$. Without loss of generality we assume that $\operatorname{rank}(A) = m$. The vectors x, y, and s are the vectors of variables.

The best know iteration bound for IIPMs, as given in [32, Theorem 5.14], is

$$O\left(n \log \frac{\max\left\{\left(x^0\right)^T s^0, \left\|b - Ax^0\right\|, \left\|c - A^T y^0 - s^0\right\|\right\}}{\varepsilon}\right). \tag{1}$$

Here $x^0 > 0$, y^0 and $s^0 > 0$ denote the starting points, and $b - Ax^0$ and $c - A^T y^0 - s^0$ are the initial primal and dual residue vectors, respectively, whereas ε is an upper bound for the duality gap and the norms of residual vectors upon termination of the algorithm. It is assumed in this result that there exists an optimal solution (x^*, y^*, s^*) such that $\|(x^*; s^*)\|_\infty \leq \zeta$, and the initial iterates are $(x^0, y^0, s^0) = \zeta(e, 0, e)$.

Until 2003, the search directions used in all primal-dual IIPMs were computed from the linear system

$$\begin{aligned} A\Delta x &= b - Ax, \\ A^T \Delta y + \Delta s &= c - A^T y - s, \\ s\Delta x + x\Delta s &= \mu e - xs, \end{aligned} \tag{2}$$

which yields tho so-called primal-dual Newton search directions Δx, Δy, and Δs. Here xs denotes the componentwise (or Hadamard) product of the vectors x and s. Recently, Salahi, Terlaky, and Zhang used a so-called self-regular proximity function to define a new search direction for IIPMs [21]. Their modification involves only the third equation in the above system. The iteration bound of their method does not improve the bound in (1).

To introduce the idea underlying the algorithm presented in this paper we make some remarks with an historical flavor. In feasible IPMs the iterates are feasible and the ultimate goal is to get iterates that are optimal. There is a well-known IPM that aims to reach optimality in one step, namely the affine-scaling method. Although variants of this method have been shown to be polynomial in [14] and [7], with complexity bounds $O(nL^2)$ and $O(nL)$, respectively, where L denotes the binary input size, these bounds are much worse than the best known bounds for IPMs. The question of whether or not the affine-scaling method is polynomial is still unsettled. The last two decades have made it very clear that to get a more efficient method one should be less greedy and work with a search direction that moves the iterates only slowly in the direction of optimality. The reason is that only then one can take full advantage of the efficiency of Newton's method, which is the workhorse in all IPMs.

In IIPMs, the iterates are not feasible, and apart from reaching optimality one needs to strive for feasibility. This is reflected by the choice of the search direction, as defined by (2) because, when moving from x to $x^+ := x + \Delta x$ the new iterate x^+ satisfies the primal feasibility constraints, except possibly the nonnegativity constraint. In fact, in general, x^+ will have negative components, and, to keep the iterate positive, one is forced to take a damped step of the form $x^+ := x + \alpha\Delta x$, where $\alpha < 1$ denotes the step size. This recalls, however, the phenomenon that occurred with the affine-scaling method in feasible IPMs. There it has become clear that the best complexity results hold for methods that are much less greedy and that use full-Newton steps (with $\alpha = 1$). Striving to reach feasibility in one step might be too greedy and may deteriorate the overall behavior of a method. One should better exercise a little patience and move slower in the direction of feasibility. Therefore, in our approach the search directions are designed in such a way that a full-Newton step reduces the sizes of the residual vectors with the same speed as the duality gap. The outcome of the analysis in this paper shows that this is a good strategy.

The paper is organized as follows. As a preparation for the rest of the paper, in section 2 we first recall some basic tools in the analysis of a feasible IPM. These tools will be used also in the analysis of the IIPM proposed in this paper. Section 3 is used to describe our algorithm in more detail. One characteristic of the algorithm is that it uses intermediate problems. The intermediate problems are suitable perturbations of the given problems (P) and (D) so that at any stage the iterates are strictly feasible for the current perturbed problems; the size of the perturbation decreases at the same speed as the barrier parameter μ. When μ changes to a smaller value, the perturbed problem corresponding to μ changes, and hence also the current central path. The algorithm keeps the iterates close to the μ-center on the central path of the current perturbed problem. To get the iterates feasible for the new perturbed problem and close to its central path, we use a so-called feasibility step. The largest, and hardest, part of the analysis, which is presented in section 4, concerns this step. It turns out that to keep control over this step, before taking the step the iterates need to be very well centered. Some concluding remarks can be found in section 5.

Some notation used throughout the paper is as follows. The 2-norm and the infinity norm are denoted by $\|\cdot\|$ and $\|\cdot\|_\infty$, respectively. If $x, s \in \mathbf{R}^n$, then xs denotes the componentwise (or Hadamard) product of the vectors x and s. Furthermore, e denotes the all-one vector of length n. If $z \in \mathbf{R}^n_+$ and $f \ : \ \mathbf{R}_+ \to \mathbf{R}_+$, then $f(z)$ denotes the vector in $\mathbf{R}^n_+$ whose ith component is $f(z_i)$, with $1 \leq i \leq n$. We write $f(x) = O(g(x))$ if $f(x) \leq \gamma\, g(x)$ for some positive constant γ.

2. Feasible full-Newton step IPMs. In preparation for dealing with IIPMs, in this section we briefly recall the classical way to obtain a polynomial-time path-following feasible IPM for solving (P) and (D). To solve these problems one needs to find a solution of the following system of equations:

$$\begin{aligned} Ax &= b, & x &\geq 0, \\ A^T y + s &= c, & s &\geq 0, \\ xs &= 0. \end{aligned}$$

In these so-called optimality conditions the first two constraints represent primal and dual feasibility, whereas the last equation is the so-called complementarity condition. The nonnegativity constraints in the feasibility conditions make the problem already nontrivial: only iterative methods can find solutions of linear systems involving inequality constraints. The complementarity condition is nonlinear, which makes it extra hard to solve this system.

2.1. The central path. IPMs replace the complementarity condition by the so-called centering condition $xs = \mu e$, where μ may be any positive number. This yields the system

$$\begin{aligned} Ax &= b, \qquad & x &\geq 0, \\ A^T y + s &= c, \qquad & s &\geq 0, \\ xs &= \mu e. \end{aligned} \tag{3}$$

Surprisingly enough, if this system has a solution for some $\mu > 0$, then a solution exists for every $\mu > 0$, and this solution is unique. This happens if and only if (P) and (D) satisfy the interior-point condition (IPC), i.e., if (P) has a feasible solution $x > 0$ and (D) has a solution (y, s) with $s > 0$ (see, e.g., [20]). If the IPC is satisfied, then the solution of (3) is denoted by $(x(\mu), y(\mu), s(\mu))$ and called the μ-center of (P) and (D). The set of all μ-centers is called the central path of (P) and (D). As μ goes to zero, $x(\mu)$, $y(\mu)$, and $s(\mu)$ converge to optimal solutions of (P) and (D). Of course, system (3) is still hard to solve, but by applying Newton's method one can easily find approximate solutions.

2.2. Definition and properties of the Newton step. We proceed by describing Newton's method for solving (3), with μ fixed. Given any primal feasible $x > 0$ and dual feasible y and $s > 0$, we want to find displacements Δx, Δy, and Δs such that

$$\begin{aligned} A(x + \Delta x) &= b, \\ A^T(y + \Delta y) + s + \Delta s &= c, \\ (x + \Delta x)(s + \Delta s) &= \mu e. \end{aligned}$$

Neglecting the quadratic term $\Delta x \Delta s$ in the left-hand side expression of the third equation, we obtain the following linear system of equations in the search directions Δx, Δy, and Δs:

$$A\Delta x = b - Ax, \tag{4}$$

$$A^T \Delta y + \Delta s = c - A^T y - s, \tag{5}$$

$$s\Delta x + x\Delta s = \mu e - xs. \tag{6}$$

Since A has full rank, and the vectors x and s are positive, one may easily verify that the coefficient matrix in the linear system (4)–(6) is nonsingular. Hence this system uniquely defines the search directions Δx, Δy, and Δs. These search directions are used in all existing primal-dual (feasible and infeasible) IPMs and are equivalent to Newton's method for solving the equations in system (3).

If x is primal feasible and (y, s) dual feasible, then $b - Ax = 0$ and $c - A^T y - s = 0$, whence the above system reduces to

$$A\Delta x = 0, \tag{7}$$

$$A^T \Delta y + \Delta s = 0, \tag{8}$$

$$s\Delta x + x\Delta s = \mu e - xs, \tag{9}$$

which gives the usual search directions for feasible primal-dual IPMs.

```
Primal-Dual Feasible IPM

Input:
  Accuracy parameter $\varepsilon > 0$;
  barrier update parameter $\theta$, $0 < \theta < 1$;
  feasible $(x^0, y^0, s^0)$ with $\left(x^0\right)^T s^0 = n\mu^0$, $\delta(x^0, s^0, \mu^0) \leq 1/2$.
begin
  $x := x^0;\ y := y^0;\ s := s^0;\ \mu := \mu^0;$
  while $x^T s \geq \varepsilon$ do
  begin
    $\mu$-update:
      $\mu := (1-\theta)\mu;$
    centering step:
      $(x,\, y,\, s) := (x,\, y,\, s) + (\Delta x,\, \Delta y,\, \Delta s);$
  end
end
```

Fig. 1. *Feasible full-Newton-step algorithm.*

The new iterates are given by

$$\begin{aligned} x^+ &= x + \Delta x, \\ y^+ &= y + \Delta y, \\ s^+ &= s + \Delta s. \end{aligned}$$

An important observation is that Δx lies in the null space of A, whereas Δs belongs to the row space of A. This implies that Δx and Δs are orthogonal, i.e.,

$$(\Delta x)^T \Delta s = 0.$$

As a consequence we have the important property that after a full-Newton step the duality gap assumes the same value as at the μ-centers, namely $n\mu$.

Lemma 2.1 (see [20, Lemma II.46]). *After a primal-dual Newton step, one has $(x^+)^T s^+ = n\mu$.*

A primal-dual feasible triplet (x, y, s) is an ε-solution of (P) and (D) if $c^T x - b^T y = x^T s \leq \varepsilon$.

Assume that a primal feasible $x^0 > 0$ and a dual feasible pair (y^0, s^0) with $s^0 > 0$ are given that are "close to" $x(\mu)$ and $(y(\mu), s(\mu))$, respectively, for some $\mu = \mu^0$. Then one can find an ε-solution in $O(\sqrt{n}\log(n/\varepsilon))$ iterations of the algorithm in Figure 1. In this algorithm $\delta(x, s; \mu)$ is a quantity that measures proximity of the feasible triple (x, y, s) to the μ-center $(x(\mu), y(\mu), s(\mu))$. Following [20], this quantity is defined as follows:

$$\delta(x, s; \mu) := \delta(v) := \frac{1}{2}\left\| v - v^{-1} \right\|, \qquad \text{where} \qquad v := \sqrt{\frac{xs}{\mu}}. \tag{10}$$

The following two lemmas are crucial in the analysis of the algorithm. We recall them without proof. They describe the effect on $\delta(x, s; \mu)$ of a μ-update and a Newton (or centering) step, respectively.

LEMMA 2.2 (see [20, Lemma II.53]). *Let (x, s) be a positive primal-dual pair and $\mu > 0$ such that $x^T s = n\mu$. Moreover, let $\delta := \delta(x, s; \mu)$ and let $\mu^+ = (1 - \theta)\mu$. Then*

$$\delta(x, s; \mu^+)^2 = (1 - \theta)\delta^2 + \frac{\theta^2 n}{4(1 - \theta)}.$$

LEMMA 2.3 (see [20, Theorem II.49]). *If $\delta := \delta(x, s; \mu) \leq 1$, then the primal-dual Newton step is feasible; i.e., x^+ and s^+ are nonnegative. Moreover, if $\delta < 1$, then x^+ and s^+ are positive and*

$$\delta(x^+, s^+; \mu) \leq \frac{\delta^2}{\sqrt{2(1 - \delta^2)}}.$$

COROLLARY 2.4. *If $\delta := \delta(x, s; \mu) \leq \frac{1}{\sqrt{2}}$, then $\delta(x^+, s^+; \mu) \leq \delta^2$.*

2.3. Complexity analysis. We have the following theorem, whose simple proof we include, because it slightly improves the complexity result in [20, Theorem II.52].

THEOREM 2.5. *If $\theta = 1/\sqrt{2n}$, then the algorithm requires at most*

$$\sqrt{2n} \log \frac{n\mu^0}{\varepsilon}$$

iterations. The output is a primal-dual pair (x, s) such that $x^T s \leq \varepsilon$.

Proof. At the start of the algorithm we have $\delta(x, s; \mu) \leq 1/2$. After the update of the barrier parameter to $\mu^+ = (1 - \theta)\mu$, with $\theta = 1/\sqrt{2n}$, we have, by Lemma 2.2, the following upper bound for $\delta(x, s; \mu^+)$:

$$\delta(x, s; \mu^+)^2 \leq \frac{1 - \theta}{4} + \frac{1}{8(1 - \theta)} \leq \frac{3}{8}.$$

Assuming $n \geq 2$, the last inequality follows since the expression on its left-hand side is a convex function of θ, whose value is $3/8$ both at $\theta = 0$ and at $\theta = 1/2$. Since $\theta \in [0, 1/2]$, the left-hand side does not exceed $3/8$. Since $3/8 < 1/2$, we obtain $\delta(x, s; \mu^+) \leq 1/\sqrt{2}$. After the primal-dual Newton step to the μ^+-center we have, by Corollary 2.4, $\delta(x^+, s^+; \mu^+) \leq 1/2$. Also, from Lemma 2.1, $(x^+)^T s^+ = n\mu^+$. Thus, after each iteration of the algorithm the properties

$$x^T s = n\mu, \quad \delta(x, s; \mu) \leq \frac{1}{2}$$

are maintained, and hence the algorithm is well defined. The iteration bound in the theorem now easily follows from the fact that in each iteration the value of $x^T s$ is reduced by the factor $1 - \theta$ (see, for example, the proof of Theorem 3.2 in [30] for such a deduction). This proves the theorem. □

3. Infeasible full-Newton step IPM. In the case of an infeasible method we call the triple (x, y, s) an ε-solution of (P) and (D) if the 2-norms of the residual vectors $b - Ax$ and $c - A^T y - s$ do not exceed ε, and also $x^T s \leq \varepsilon$. In this section we present an infeasible-start algorithm that generates an ε-solution of (P) and (D), if it exists, or establishes that no such solution exists.

3.1. The perturbed problems. We start with choosing arbitrarily $x^0 > 0$ and y^0, $s^0 > 0$ such that $x^0 s^0 = \mu^0 e$ for some (positive) number μ^0. For any ν with $0 < \nu \le 1$ we consider the perturbed problem (P$_\nu$), defined by

$$(\mathrm{P}_\nu) \qquad \min\left\{\left(c - \nu\left(c - A^T y^0 - s^0\right)\right)^T x : Ax = b - \nu\left(b - Ax^0\right), \quad x \ge 0\right\},$$

and its dual problem (D$_\nu$), which is given by

$$(\mathrm{D}_\nu) \quad \max\left\{\left(b - \nu\left(b - Ax^0\right)\right)^T y : A^T y + s = c - \nu\left(c - A^T y^0 - s^0\right), \quad s \ge 0\right\}.$$

Note that if $\nu = 1$, then $x = x^0$ yields a strictly feasible solution of (P$_\nu$) and $(y, s) = (y^0, s^0)$ a strictly feasible solution of (D$_\nu$). We conclude that if $\nu = 1$, then (P$_\nu$) and (D$_\nu$) satisfy the IPC.

Lemma 3.1 (cf. [32, Theorem 5.13]). *The original problems,* (P) *and* (D)*, are feasible if and only if for each ν satisfying $0 < \nu \le 1$ the perturbed problems* (P$_\nu$) *and* (D$_\nu$) *satisfy the IPC.*

Proof. Suppose that (P) and (D) are feasible. Let $\bar{x}$ be a feasible solution of (P) and $(\bar{y}, \bar{s})$ a feasible solution of (D). Then $A\bar{x} = b$ and $A^T\bar{y} + \bar{s} = c$, with $\bar{x} \ge 0$ and $\bar{s} \ge 0$. Now let $0 < \nu \le 1$, and consider

$$x = (1-\nu)\,\bar{x} + \nu\, x^0, \quad y = (1-\nu)\,\bar{y} + \nu\, y^0, \quad s = (1-\nu)\,\bar{s} + \nu\, s^0.$$

One has

$$Ax = A\left((1-\nu)\,\bar{x} + \nu\, x^0\right) = (1-\nu)\, A\bar{x} + \nu A x^0 = (1-\nu)\, b + \nu A x^0 = b - \nu\left(b - Ax^0\right),$$

showing that x is feasible for (P$_\nu$). Similarly,

$$\begin{aligned} A^T y + s &= (1-\nu)\left(A^T\bar{y} + \bar{s}\right) + \nu\left(A^T y^0 + s^0\right) \\ &= (1-\nu)\, c + \nu\left(A^T y^0 + s^0\right) = c - \nu\left(c - A^T y^0 - s^0\right), \end{aligned}$$

showing that (y, s) is feasible for (D$_\nu$). Since $\nu > 0$, x and s are positive, thus proving that (P$_\nu$) and (D$_\nu$) satisfy the IPC.

To prove the inverse implication, suppose that (P$_\nu$) and (D$_\nu$) satisfy the IPC for each ν satisfying $0 < \nu \le 1$. Obviously, then (P$_\nu$) and (D$_\nu$) are feasible for these values of ν. Letting ν go to zero it follows that (P) and (D) are feasible. □

In the sections to follow we assume that (P) and (D) are feasible. Only in section 4.7 will we discuss how our algorithm can be used to detect infeasibility or unboundedness of (P) and (D). It may be worth noting that if (P) and (D) satisfy the IPC and x^0 and (y^0, s^0) are feasible, then (P$_\nu$) $\equiv$ (P) and (D$_\nu$) $\equiv$ (D) for each $\nu \in (0, 1]$.

3.2. The central path of the perturbed problems. Let (P) and (D) be feasible and $0 < \nu \le 1$. Then Lemma 3.1 implies that the problems (P$_\nu$) and (D$_\nu$) satisfy the IPC, and hence their central paths exist. This means that the system

$$b - Ax = \nu(b - Ax^0), \qquad x \ge 0, \tag{11}$$

$$c - A^T y - s = \nu(c - A^T y^0 - s^0), \qquad s \ge 0, \tag{12}$$

$$xs = \mu e \tag{13}$$

has a unique solution for every $\mu > 0$. In what follows this unique solution is denoted by $(x(\mu, \nu), y(\mu, \nu), s(\mu, \nu))$. These are the μ-centers of the perturbed problems (P$_\nu$) and (D$_\nu$).

Note that since $x^0 s^0 = \mu^0 e$, x^0 is the μ^0-center of the perturbed problem (P_1) and (y^0, s^0) the μ^0-center of (D_1). In other words, $(x(\mu^0, 1),\, y(\mu^0, 1),\, s(\mu^0, 1)) = (x^0, y^0, s^0)$. In what follows the parameters μ and ν always satisfy the relation $\mu = \nu\,\mu^0$.

3.3. An iteration of our algorithm. We just established that if $\nu = 1$ and $\mu = \mu^0$, then $x = x^0$ is the μ-center of the perturbed problem (P_ν) and $(y, s) = (y^0, s^0)$ the μ-center of (D_ν). These are our initial iterates.

We measure proximity to the μ-center of the perturbed problems by the quantity $\delta(x, s; \mu)$ as defined in (10). Initially we thus have $\delta(x, s; \mu) = 0$. In what follows we assume that at the start of each iteration, just before the μ-update, $\delta(x, s; \mu)$ is smaller than or equal to a (small) threshold value $\tau > 0$. So this is certainly true at the start of the first iteration.

Now we describe one (main) iteration of our algorithm. Suppose that for some $\mu \in (0, \mu^0]$ we have x, y, and s satisfying the feasibility conditions (11) and (12) for $\nu = \mu/\mu^0$ and such that $x^T s = n\mu$ and $\delta(x, s; \mu) \le \tau$. We reduce μ to $\mu^+ = (1 - \theta)\mu$, with $\theta \in (0, 1)$, and find new iterates x^+, y^+, and s^+ that satisfy (11) and (12), with μ replaced by μ^+ and ν by $\nu^+ = \mu^+/\mu^0$, and such that $x^T s = n\mu^+$ and $\delta(x^+, s^+; \mu^+) \le \tau$. Note that $\nu^+ = (1 - \theta)\nu$.

To be more precise, this is achieved as follows. Each main iteration consists of a feasibility step and a few centering steps. The feasibility step serves to get iterates (x^f, y^f, s^f) that are strictly feasible for (P_{ν^+}) and (D_{ν^+}) and close to their μ-centers $(x(\mu^+, \nu^+),\, y(\mu^+, \nu^+),\, s(\mu^+, \nu^+))$. In fact, the feasibility step is designed in such a way that $\delta(x^f, s^f; \mu^+) \le 1/\sqrt{2}$. Since the triple (x^f, y^f, s^f) is strictly feasible for (P_{ν^+}) and (D_{ν^+}), we then can easily get iterates (x^+, y^+, s^+) that are strictly feasible for (P_{ν^+}) and (D_{ν^+}) and such that $\delta(x, s; \mu^+) \le \tau$, just by performing a few centering steps starting at (x^f, y^f, s^f) and targeting at the μ^+-centers of (P_{ν^+}) and (D_{ν^+}).

Before describing the feasibility step it will be convenient to introduce some new notation. We denote the initial values of the primal and dual residuals r_b^0 and r_c^0, respectively, as

$$\begin{aligned} r_b^0 &= b - Ax^0, \\ r_c^0 &= c - A^T y^0 - s^0. \end{aligned}$$

Then the feasibility equations for (P_ν) and (D_ν) are given by

$$Ax = b - \nu r_b^0, \qquad x \ge 0, \tag{14}$$

$$A^T y + s = c - \nu r_c^0, \qquad s \ge 0, \tag{15}$$

and those of (P_{ν^+}) and (D_{ν^+}) by

$$Ax = b - \nu^+ r_b^0, \qquad x \ge 0, \tag{16}$$

$$A^T y + s = c - \nu^+ r_c^0, \qquad s \ge 0. \tag{17}$$

Now suppose that the triplet (x, y, s) is feasible for (P_ν) and (D_ν). To get iterates that are feasible for (P_{ν^+}) and (D_{ν^+}) we need search directions $\Delta^f x$, $\Delta^f y$, and $\Delta^f s$ such that

$$\begin{aligned} A(x + \Delta^f x) &= b - \nu^+ r_b^0, \\ A^T(y + \Delta^f y) + (s + \Delta^f s) &= c - \nu^+ r_c^0. \end{aligned}$$

Since x is feasible for (P_ν) and (y,s) is feasible for (D_ν), it follows that $\Delta^f x$, $\Delta^f y$, and $\Delta^f s$ should satisfy

$$\begin{aligned} A\Delta^f x &= (b - Ax) - \nu^+ r_b^0 = \nu r_b^0 - \nu^+ r_b^0 = \theta\nu r_b^0, \\ A^T\Delta^f y + \Delta^f s &= (c - A^T y - s) - \nu^+ r_c^0 = \nu r_c^0 - \nu^+ r_c^0 = \theta\nu r_c^0. \end{aligned}$$

Therefore, the following system is used to define $\Delta^f x$, $\Delta^f y$, and $\Delta^f s$:

$$A\Delta^f x = \theta\nu r_b^0, \tag{18}$$

$$A^T\Delta^f y + \Delta^f s = \theta\nu r_c^0, \tag{19}$$

$$s\Delta^f x + x\Delta^f s = \mu e - xs, \tag{20}$$

and after the feasibility step the iterates are given by

$$x^f = x + \Delta^f x, \tag{21}$$

$$y^f = y + \Delta^f y, \tag{22}$$

$$s^f = s + \Delta^f s. \tag{23}$$

We conclude that after the feasibility step the iterates satisfy the affine equations in (11) and (12), with $\nu = \nu^+$. The hard part in the analysis will be to guarantee that x^f and s^f are positive and satisfy $\delta(x^f, s^f; \mu^+) \le 1/\sqrt{2}$.

After the feasibility step we perform centering steps in order to get iterates (x^+, y^+, s^+) that satisfy ${x^+}^T s^+ = n\mu^+$ and $\delta(x^+, s^+; \mu^+) \le \tau$. By using Corollary 2.4, the required number of centering steps can easily be obtained. Indeed, assuming $\delta = \delta(x^f, s^f; \mu^+) \le 1/\sqrt{2}$, after k centering steps we will have iterates (x^+, y^+, s^+) that are still feasible for (P_{ν^+}) and (D_{ν^+}) and that satisfy

$$\delta(x^+, s^+; \mu^+) \le \left(\frac{1}{\sqrt{2}}\right)^{2^k}.$$

Thus, $\delta(x^+, s^+; \mu^+) \le \tau$ if k satisfies

$$\left(\frac{1}{\sqrt{2}}\right)^{2^k} \le \tau,$$

which gives

$$k \ge \log_2\left(\log_2 \frac{1}{\tau^2}\right). \tag{24}$$

3.4. The algorithm. A more formal description of the algorithm is given in Figure 2. Note that after each iteration the residuals and the duality gap are reduced by a factor $1 - \theta$. The algorithm stops if the norms of the residuals and the duality gap are less than the accuracy parameter ε.

4. An analysis of the algorithm.

Let x, y, and s denote the iterates at the start of an iteration, and assume $\delta(x, s; \mu) \le \tau$. Recall that at the start of the first iteration this is certainly true, because then $\delta(x, s; \mu) = 0$.

```
Primal-Dual Infeasible IPM

Input:
  Accuracy parameter $\varepsilon > 0$;
  barrier update parameter $\theta$, $0 < \theta < 1$;
  threshold parameter $\tau > 0$.
begin
  $x := x^0 > 0;\ y := y^0;\ s := s^0 > 0;\ x^0 s^0 = \mu^0 e;\ \mu = \mu^0;\ \nu = 1;$
  while $\max\left(x^T s,\, \|b - Ax\|,\, \left\|c - A^T y - s\right\|\right) \geq \varepsilon$ do
  begin
    feasibility step:
      $(x,\, y,\, s) := (x,\, y,\, s) + (\Delta^f x,\, \Delta^f y,\, \Delta^f s);$
    update of $\mu$ and $\nu$:
      $\mu := (1 - \theta)\mu;$
      $\nu := (1 - \theta)\nu;$
    centering steps:
    while $\delta(x, s; \mu) \geq \tau$ do
      $(x,\, y,\, s) := (x,\, y,\, s) + (\Delta x,\, \Delta y,\, \Delta s);$
    endwhile
  end
end
```

Fig. 2. *The algorithm.*

4.1. The effect of the feasibility step and the choice of θ.

As we established in section 3.3, the feasibility step generates new iterates $\left(x^f,\, y^f,\, s^f\right)$ that satisfy the feasibility conditions for (P$_{\nu^+}$) and (D$_{\nu^+}$), except possibly the nonnegativity constraints. A crucial element in the analysis is to show that after the feasibility step $\delta(x^f, s^f; \mu^+) \leq 1/\sqrt{2}$, i.e., that the iterates $\left(x^f,\, y^f,\, s^f\right)$ are within the region where the Newton process targeting at the μ^+-centers of (P$_{\nu^+}$) and (D$_{\nu^+}$) is quadratically convergent.

Defining

$$v = \sqrt{\frac{xs}{\mu}}, \quad d_x := \frac{v\Delta^f x}{x}, \quad d_s := \frac{v\Delta^f s}{s}, \tag{25}$$

we have, using (20) and (25),

$$x^f s^f = xs + \left(s\Delta^f x + x\Delta^f s\right) + \Delta^f x\Delta^f s = \mu e + \Delta^f x\Delta^f s = \mu e + \frac{xs}{v^2}\, d_x d_s = \mu\left(e + d_x d_s\right). \tag{26}$$

Lemma 4.1 (cf. [20, Lemma II.45]). *The iterates $\left(x^f, y^f, s^f\right)$ are strictly feasible if and only if $e + d_x d_s > 0$.*

Proof. Note that if x^f and s^f are positive, then (26) makes clear that $e + d_x d_s > 0$, proving the "only if" part of the statement in the lemma. For the proof of the converse implication we introduce a step length $\alpha \in [0,1]$, and we define

$$x^\alpha = x + \alpha\Delta^f x, \quad y^\alpha = y + \alpha\Delta^f y, \quad s^\alpha = s + \alpha\Delta^f s.$$

We then have $x^0 = x, x^1 = x^+$ and similar relations for y and s. Hence we have $x^0 s^0 = xs > 0$. We write

$$x^\alpha s^\alpha = (x + \alpha\Delta^f x)(s + \alpha\Delta^f s) = xs + \alpha\left(s\Delta^f x + x\Delta^f s\right) + \alpha^2 \Delta^f x \Delta^f s.$$

Using $s\Delta^f x + x\Delta^f s = \mu e - xs$ gives

$$x^\alpha s^\alpha = xs + \alpha\left(\mu e - xs\right) + \alpha^2 \Delta^f x \Delta^f s.$$

Now suppose $e + d_x d_s > 0$. From the definitions of d_x and d_s in (25) we deduce $\mu d_x d_s = \Delta^f x \Delta^f s$. Hence $\mu e + \Delta^f x \Delta^f s > 0$, or, equivalently, $\Delta^f x \Delta^f s > -\mu e$. Substitution gives

$$x^\alpha s^\alpha > xs + \alpha\left(\mu e - xs\right) - \alpha^2 \mu e = (1-\alpha)\left(xs + \alpha\mu e\right), \quad \alpha \in [0,1].$$

Since $(1-\alpha)\left(xs + \alpha\mu e\right) \geq 0$ it follows that $x^\alpha s^\alpha > 0$ for $0 \leq \alpha \leq 1$. Hence, none of the entries of x^α and s^α vanishes for $0 \leq \alpha \leq 1$. Since x^0 and s^0 are positive, and x^α and s^α depend linearly on α, this implies that $x^\alpha > 0$ and $s^\alpha > 0$ for $0 \leq \alpha \leq 1$. Hence, x^1 and s^1 must be positive, proving the "if" part of the statement in the lemma. □

COROLLARY 4.2. *The iterates $\left(x^f, y^f, s^f\right)$ are strictly feasible if $\left\|d_x d_s\right\|_\infty < 1$.*

Proof. By Lemma 4.1, x^f and s^f are strictly feasible if and only if $e + d_x d_s > 0$. Since the last inequality holds if $\left\|d_x d_s\right\|_\infty < 1$, the corollary follows. □

In what follows we denote

$$\omega(v) := \tfrac{1}{2}\sqrt{\left\|d_x\right\|^2 + \left\|d_s\right\|^2}. \tag{27}$$

This implies $\|d_x\| \leq 2\omega(v)$ and $\|d_s\| \leq 2\omega(v)$, and, moreover,

$$d_x^T d_s \leq \|d_x\|\,\|d_s\| \leq \tfrac{1}{2}\left(\left\|d_x\right\|^2 + \left\|d_s\right\|^2\right) \leq 2\omega(v)^2, \tag{28}$$

$$\left\|d_x d_s\right\|_\infty \leq \|d_x\|\,\|d_s\| \leq 2\omega(v)^2. \tag{29}$$

LEMMA 4.3. *If $\omega(v) < 1/\sqrt{2}$, then the iterates $\left(x^f, y^f, s^f\right)$ are strictly feasible.*

Proof. Let $\omega(v) < 1/\sqrt{2}$. Then (29) implies that $\left\|d_x d_s\right\|_\infty < 1$. By Corollary 4.2 this implies the statement in the lemma. □

Assuming $\omega(v) < 1/\sqrt{2}$, which guarantees strict feasibility of the iterates $\left(x^f, y^f, s^f\right)$, we proceed by deriving an upper bound for $\delta(x^f, s^f; \mu^+)$. By definition (10) we have

$$\delta(x^f, s^f; \mu^+) = \frac{1}{2}\left\|v^f - \frac{e}{v^f}\right\|, \text{ where } v^f = \sqrt{\frac{x^f s^f}{\mu^+}}.$$

In what follows we denote $\delta(x^f, s^f; \mu^+)$ also shortly by $\delta(v^f)$.

LEMMA 4.4. *Let $\omega(v) < 1/\sqrt{2}$. Then one has*

$$4\delta(v^f)^2 \leq \frac{\theta^2 n}{1-\theta} + \frac{2\omega(v)^2}{1-\theta} + (1-\theta)\,\frac{2\omega(v)^2}{1-2\omega(v)^2}.$$

Proof. Using (26), after division of both sides by $\mu^+ = (1-\theta)\mu$, we get

$$\left(v^f\right)^2 = \frac{\mu(e + d_x d_s)}{\mu^+} = \frac{e + d_x d_s}{1-\theta}.$$

Due to Lemma 4.3 and Corollary 4.2 we have $\|d_x d_s\|_\infty < 1$, whence $e + d_x d_s > 0$. By defining for the moment $u = \sqrt{e + d_x d_s}$, we have $v^f = u/\sqrt{1-\theta}$. Hence we may write

$$2\delta(v^f) = \left\| \frac{u}{\sqrt{1-\theta}} - \sqrt{1-\theta}\, u^{-1} \right\| = \left\| \frac{\theta u}{\sqrt{1-\theta}} + \sqrt{1-\theta}\left(u - u^{-1}\right) \right\|.$$

Therefore we have

$$\begin{aligned} 4\delta(v^f)^2 &= \frac{\theta^2}{1-\theta}\|u\|^2 + (1-\theta)\left\|u - u^{-1}\right\|^2 + 2\theta u^T\left(u - u^{-1}\right) \\ &= \left(\frac{\theta^2}{1-\theta} + 2\theta\right)\|u\|^2 + (1-\theta)\left\|u - u^{-1}\right\|^2 - 2\theta u^T u^{-1} \\ &= \left(\frac{\theta^2}{1-\theta} + 2\theta\right) e^T\left(e + d_x d_s\right) + (1-\theta)\left\|u - u^{-1}\right\|^2 - 2\theta n \\ &= \left(\frac{\theta^2}{1-\theta} + 2\theta\right)\left(n + d_x^T d_s\right) + (1-\theta)\left\|u - u^{-1}\right\|^2 - 2\theta n \\ &= \frac{\theta^2 n}{1-\theta} + \left(\frac{\theta^2}{1-\theta} + 2\theta\right) d_x^T d_s + (1-\theta)\left\|u^{-1} - u\right\|^2. \end{aligned}$$

The last term can be reduced as follows:

$$\begin{aligned} \left\|u^{-1} - u\right\|^2 &= e^T\left(e + d_x d_s + \frac{e}{e + d_x d_s} - 2e\right) = d_x^T d_s + \sum_{i=1}^n \frac{1}{1 + d_{xi} d_{si}} - n \\ &= d_x^T d_s + \sum_{i=1}^n \left(\frac{1}{1 + d_{xi} d_{si}} - 1\right) = d_x^T d_s - \sum_{i=1}^n \frac{d_{xi} d_{si}}{1 + d_{xi} d_{si}}. \end{aligned}$$

Substitution gives

$$\begin{aligned} 4\delta(v^f)^2 &= \frac{\theta^2 n}{1-\theta} + \left(\frac{\theta^2}{1-\theta} + 2\theta\right) d_x^T d_s + (1-\theta)\left(d_x^T d_s - \sum_{i=1}^n \frac{d_{xi} d_{si}}{1 + d_{xi} d_{si}}\right) \\ &= \frac{\theta^2 n}{1-\theta} + \frac{d_x^T d_s}{1-\theta} - (1-\theta)\sum_{i=1}^n \frac{d_{xi} d_{si}}{1 + d_{xi} d_{si}}. \end{aligned}$$

Hence, using (28) and (29), we arrive at

$$\begin{aligned} 4\delta(v^f)^2 &\leq \frac{\theta^2 n}{1-\theta} + \frac{2\omega(v)^2}{1-\theta} + (1-\theta)\sum_{i=1}^n \frac{|d_{xi} d_{si}|}{1 - 2\omega(v)^2} \\ &\leq \frac{\theta^2 n}{1-\theta} + \frac{2\omega(v)^2}{1-\theta} + \frac{1-\theta}{2}\sum_{i=1}^n \frac{{d_x}_i^2 + {d_s}_i^2}{1 - 2\omega(v)^2} \\ &= \frac{\theta^2 n}{1-\theta} + \frac{2\omega(v)^2}{1-\theta} + (1-\theta)\,\frac{2\omega(v)^2}{1 - 2\omega(v)^2}, \end{aligned}$$

which completes the proof. □

Because we need to have $\delta(v^f) \leq 1/\sqrt{2}$, it follows from Lemma 4.4 that it suffices to have

$$\frac{\theta^2 n}{1-\theta} + \frac{2\omega(v)^2}{1-\theta} + (1-\theta)\,\frac{2\omega(v)^2}{1-2\omega(v)^2} \leq 2. \tag{30}$$

We conclude this section by presenting a value that we do not allow $\omega(v)$ to exceed and by choosing a value for θ so that inequality (30) is satisfied.

LEMMA 4.5. *Let $\omega(v) \leq \frac{1}{2}$ and*

$$\theta = \frac{\alpha}{\sqrt{2n}}, \quad 0 \leq \alpha \leq \sqrt{\frac{n}{n+1}}. \tag{31}$$

Then the iterates $\left(x^f, y^f, s^f\right)$ are strictly feasible and $\delta(v^f) \leq \frac{1}{\sqrt{2}}$.

Proof. Since $\omega(v) \leq \frac{1}{2}$, the iterates $\left(x^f, y^f, s^f\right)$ are certainly strictly feasible, due to Lemma 4.3. As we just established, $\delta(v^f) \leq 1/\sqrt{2}$ holds if inequality (30) is satisfied. The left-hand side in this inequality is monotonically increasing in $\omega(v)$. By substituting $\omega(v) = \frac{1}{2}$, the inequality reduces to

$$\frac{\theta^2 n}{1-\theta} + \frac{1}{2\,(1-\theta)} + (1-\theta) \leq 2.$$

Using that $\theta \geq 0$, one easily verifies that this is equivalent to

$$\theta \leq \frac{1}{\sqrt{2n+2}},$$

which is in agreement with (31). Thus the lemma has been proved. □

We proceed by considering the vectors d_x and d_s in more detail in order to obtain an upper bound for $\omega(v)$.

4.2. The scaled search directions d_x and d_s.

One may easily check that the system (18)–(20), which defines the search directions $\Delta^f x$, $\Delta^f y$, and $\Delta^f s$, can be expressed in terms of the scaled search directions d_x and d_s as follows:

$$\bar{A} d_x = \theta \nu r_b^0, \tag{32}$$

$$\bar{A}^T \frac{\Delta^f y}{\mu} + d_s = \theta \nu v s^{-1} r_c^0, \tag{33}$$

$$d_x + d_s = v^{-1} - v, \tag{34}$$

where

$$\bar{A} = A V^{-1} X, \quad V = \operatorname{diag}(v), \quad X = \operatorname{diag}(x). \tag{35}$$

If r_b^0 and r_c^0 are zero, i.e., if the initial iterates are feasible, then d_x and d_s are orthogonal vectors, since then the vector d_x belongs to the null space and d_s to the row space of the matrix $\bar{A}$. It follows that d_x and d_s form an orthogonal decomposition of the vector $v^{-1} - v$. As a consequence we then have obvious upper bounds for the norms of d_x and d_s, namely $\|d_x\| \leq 2\delta(v)$ and $\|d_s\| \leq 2\delta(v)$, and, moreover, $\omega(v) = \delta(v)$, with $\omega(v)$ as defined in (27).

In the infeasible case, orthogonality of d_x and d_s may not be assumed, however, and the situation may be quite different. This is illustrated in Figure 3. As a consequence, it becomes much harder to get upper bounds for $\|d_x\|$ and $\|d_s\|$, thus complicating the analysis of the algorithm in comparison with feasible IPMs. To obtain an upper bound for $\omega(v)$ is the subject of several subsections to follow.

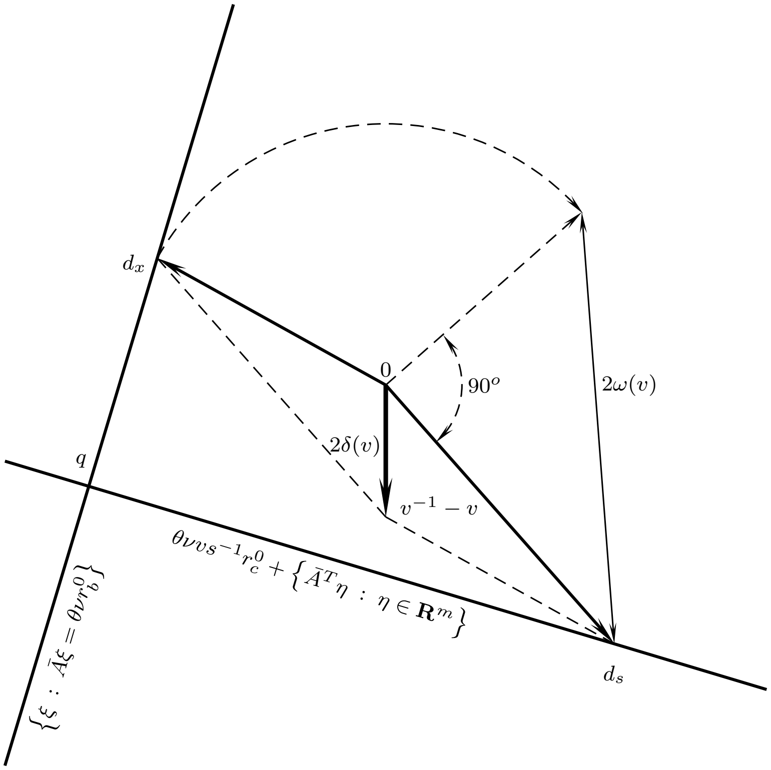

d_x
0
90^o
$2\omega(v)$
$2\delta(v)$
q
$v^{-1} - v$
$\theta\nu v s^{-1} r_c^0 + \left\{\bar{A}^T\eta \,:\, \eta \in \mathbf{R}^m\right\}$
d_s
$\left\{\xi \,:\, \bar{A}\xi = \theta\nu r_b^0\right\}$

FIG. 3. *Geometric interpretation of* $\omega(v)$.

4.3. An upper bound for $\omega(v)$. Let us denote the null space of the matrix $\bar{A}$ as $\mathcal{L}$. So,

$$\mathcal{L} := \left\{\xi \in \mathbf{R}^n \;:\; \bar{A}\xi = 0\right\}.$$

Then the affine space $\left\{\xi \in \mathbf{R}^n \;:\; \bar{A}\xi = \theta\nu r_b^0\right\}$ equals $d_x + \mathcal{L}$. Note that due to a well-known result from linear algebra the row space of $\bar{A}$ equals the orthogonal complement $\mathcal{L}^\perp$ of $\mathcal{L}$. Obviously, $d_s \in \theta\nu v s^{-1} r_c^0 + \mathcal{L}^\perp$. Also note that $\mathcal{L} \cap \mathcal{L}^\perp = \{0\}$, and as a consequence the affine spaces $d_x + \mathcal{L}$ and $d_s + \mathcal{L}^\perp$ meet in a unique point. This point is denoted by q.

LEMMA 4.6. *Let q be the (unique) point in the intersection of the affine spaces $d_x + \mathcal{L}$ and $d_s + \mathcal{L}^\perp$. Then*

$$2\omega(v) \le \sqrt{\|q\|^2 + (\|q\| + 2\delta(v))^2}.$$

Proof. To simplify the notation in this proof we denote $r = v^{-1} - v$. Since $\mathcal{L} + \mathcal{L}^\perp = \mathbf{R}^n$, there exist $q_1, r_1 \in \mathcal{L}$ and $q_2, r_2 \in \mathcal{L}^\perp$ such that

$$q = q_1 + q_2, \quad r = r_1 + r_2.$$

On the other hand, since $d_x - q \in \mathcal{L}$ and $d_s - q \in \mathcal{L}^\perp$ there must exist $\ell_1 \in \mathcal{L}$ and $\ell_2 \in \mathcal{L}^\perp$ such that

$$d_x = q + \ell_1, \quad d_s = q + \ell_2.$$

Due to (34) it follows that $r = 2q + \ell_1 + \ell_2$, which implies

$$(2q_1 + \ell_1) + (2q_2 + \ell_2) = r_1 + r_2.$$

Since the decomposition $\mathcal{L} + \mathcal{L}^\perp = \mathbf{R}^n$ is unique, we conclude that

$$\ell_1 = r_1 - 2q_1, \quad \ell_2 = r_2 - 2q_2.$$

Hence we obtain

$$\begin{aligned} d_x &= q + r_1 - 2q_1 = (r_1 - q_1) + q_2, \\ d_s &= q + r_2 - 2q_2 = q_1 + (r_2 - q_2). \end{aligned}$$

Since the spaces $\mathcal{L}$ and $\mathcal{L}^\perp$ are orthogonal we conclude from this that

$$4\omega(v)^2 = \|d_x\|^2 + \|d_s\|^2 = \|r_1 - q_1\|^2 + \|q_2\|^2 + \|q_1\|^2 + \|r_2 - q_2\|^2 = \|q - r\|^2 + \|q\|^2 .$$

Assuming $q \neq 0$, since $\|r\| = 2\delta(v)$ the right-hand side is maximal if $r = -2\delta(v)q / \|q\|$, and thus we obtain

$$4\omega(v)^2 \leq \left\| \left(1 + \frac{2\delta(v)}{\|q\|}\right) q \right\|^2 + \|q\|^2 = \|q\|^2 + (\|q\| + 2\delta(v))^2 ,$$

which implies the inequality in the lemma if $q \neq 0$. Since the inequality in the lemma holds with equality if $q = 0$, this completes the proof. □

In what follows we denote $\delta(v)$ simply as δ. Recall from Lemma 4.5 that in order to guarantee that $\delta(v^f) \leq \frac{1}{\sqrt{2}}$ we want to have $\omega(v) \leq \frac{1}{2}$. Due to Lemma 4.6 this will certainly hold if $\|q\|$ satisfies

$$\|q\|^2 + (\|q\| + 2\delta)^2 \leq 1. \tag{36}$$

4.4. An upper bound for $\|q\|$. Recall from Lemma 4.6 that q is the (unique) solution of the system

$$\begin{aligned} \bar{A}q &= \theta\nu r_b^0, \\ \bar{A}^T\xi + q &= \theta\nu v s^{-1} r_c^0. \end{aligned}$$

We proceed by deriving an upper bound for $\|q\|$. Before doing this we have to specify our initial iterates (x^0, y^0, s^0). These are chosen in the usual way. So, we assume that $\zeta > 0$ is such that $\|x^* + s^*\|_\infty \leq \zeta$ for some optimal solutions x^* of (P) and (y^*, s^*) of (D), and we start the algorithm with

$$x^0 = s^0 = \zeta e, \quad y^0 = 0, \quad \mu^0 = \zeta^2.$$

We have the following result.

Lemma 4.7. *One has*

$$\sqrt{\mu}\, \|q\| \leq \theta\nu\, \zeta \sqrt{e^T \left(\frac{x}{s} + \frac{s}{x}\right)}. \tag{37}$$

Proof. From the definition (35) of $\bar{A}$ we deduce that $\bar{A} = \sqrt{\mu}\, AD$, where

$$D = \operatorname{diag}\left(\frac{xv^{-1}}{\sqrt{\mu}}\right) = \operatorname{diag}\left(\sqrt{\frac{x}{s}}\right) = \operatorname{diag}\left(\sqrt{\mu}\, vs^{-1}\right).$$

For the moment, let us write

$$r_b = \theta\nu r_b^0, \quad r_c = \theta\nu r_c^0.$$

Then the system defining q is equivalent to

$$\begin{aligned} \sqrt{\mu}\, AD\, q &= r_b, \\ \sqrt{\mu}\, DA^T\xi + q &= \frac{1}{\sqrt{\mu}}\, Dr_c. \end{aligned}$$

This implies

$$\mu\, AD^2A^T\xi = AD^2r_c - r_b,$$

whence

$$\xi = \frac{1}{\mu}\left(AD^2A^T\right)^{-1}\left(AD^2r_c - r_b\right).$$

Substitution gives

$$q = \frac{1}{\sqrt{\mu}}\left(Dr_c - DA^T\left(AD^2A^T\right)^{-1}\left(AD^2r_c - r_b\right)\right).$$

Observe that

$$q_1 := Dr_c - DA^T\left(AD^2A^T\right)^{-1}AD^2r_c = \left(I - DA^T\left(AD^2A^T\right)^{-1}AD\right)Dr_c$$

is the orthogonal projection of Dr_c onto the null space of AD. Let $(\bar{y}, \bar{s})$ be such that $A^T\bar{y} + \bar{s} = c$. Then we may write

$$r_c = \theta\nu r_c^0 = \theta\nu\left(c - A^Ty^0 - s^0\right) = \theta\nu\left(A^T(\bar{y} - y^0) + \bar{s} - s^0\right).$$

Since $DA^T(\bar{y} - y^0)$ belongs to the row space of AD, which is orthogonal to the null space of AD, we obtain

$$\|q_1\| \leq \theta\nu\left\|D\left(\bar{s} - s^0\right)\right\|.$$

On the other hand, let $\bar{x}$ be such that $A\bar{x} = b$. Then

$$r_b = \theta\nu r_b^0 = \theta\nu(b - Ax^0) = \theta\nu A(\bar{x} - x^0),$$

and the vector

$$q_2 := DA^T\left(AD^2A^T\right)^{-1}r_b = \theta\nu DA^T\left(AD^2A^T\right)^{-1}AD\left(D^{-1}\left(\bar{x} - x^0\right)\right)$$

is the orthogonal projection of $\theta\nu D^{-1}\left(\bar{x} - x^0\right)$ onto the row space of AD. Hence it follows that

$$\|q_2\| \leq \theta\nu\left\|D^{-1}\left(\bar{x} - x^0\right)\right\|.$$

Since $\sqrt{\mu}\, q = q_1 + q_2$ and q_1 and q_2 are orthogonal, we may conclude that

$$\sqrt{\mu}\,\|q\| = \sqrt{\|q_1\|^2 + \|q_2\|^2} \leq \theta\nu\sqrt{\left\|D\left(\bar{s} - s^0\right)\right\|^2 + \left\|D^{-1}\left(\bar{x} - x^0\right)\right\|^2}, \tag{38}$$

where, as always, $\mu = \mu^0\nu$.

We are still free to choose $\bar{x}$ and $\bar{s}$, subject to the constraints $A\bar{x} = b$ and $A^T\bar{y}+\bar{s} = c$. Taking $\bar{x} = x^*$ and $\bar{s} = s^*$ the entries of the vectors $x^0 - \bar{x}$ and $s^0 - \bar{s}$ satisfy

$$0 \leq x^0 - \bar{x} \leq \zeta e, \quad 0 \leq s^0 - \bar{s} \leq \zeta e.$$

Thus it follows that

$$\sqrt{\left\|D\left(\bar{s}-s^0\right)\right\|^2 + \left\|D^{-1}\left(\bar{x}-x^0\right)\right\|^2} \leq \zeta\sqrt{\|De\|^2 + \|D^{-1}e\|^2} = \zeta\sqrt{e^T\left(\frac{x}{s}+\frac{s}{x}\right)}.$$

Substitution into (38) gives

$$\sqrt{\mu}\ \|q\| \leq \theta\nu\,\zeta\sqrt{e^T\left(\frac{x}{s}+\frac{s}{x}\right)},$$

proving the lemma. □

To proceed we need upper bounds for the elements of the vectors x/s and s/x.

4.5. Some bounds for $\boldsymbol{x}$ and $\boldsymbol{s}$. The choice of $\boldsymbol{\tau}$ and $\boldsymbol{\alpha}$. Recall that x is feasible for (P_ν) and (y,s) for (D_ν) and, moreover, $\delta(x,s;\mu) \leq \tau$; i.e., these iterates are close to the μ-centers of (P_ν) and (D_ν). Based on this information we need to estimate the sizes of the entries of the vectors x/s and s/x. Since the concerning analysis does not belong to the mainstream of the paper we have moved this analysis to the appendix. Based on this analysis we choose

$$\tau = \frac{1}{8}. \tag{39}$$

Then, by Corollary A.10, we have

$$\sqrt{\frac{x}{s}} \leq \sqrt{2}\,\frac{x(\mu,\nu)}{\sqrt{\mu}}, \qquad \sqrt{\frac{s}{x}} \leq \sqrt{2}\,\frac{s(\mu,\nu)}{\sqrt{\mu}}.$$

Substitution into (37) gives

$$\sqrt{\mu}\ \|q\| \leq \theta\nu\,\zeta\sqrt{2e^T\left(\frac{x(\mu,\nu)^2}{\mu}+\frac{s(\mu,\nu)^2}{\mu}\right)}.$$

This implies

$$\mu\ \|q\| \leq \theta\nu\zeta\sqrt{2}\,\sqrt{\|x(\mu,\nu)\|^2 + \|s(\mu,\nu)\|^2}.$$

Therefore, also using $\mu = \mu^0\nu = \zeta^2\nu$ and $\theta = \frac{\alpha}{\sqrt{2n}}$, we obtain the following upper bound for the norm of q:

$$\|q\| \leq \frac{\alpha}{\zeta\sqrt{n}}\,\sqrt{\|x(\mu,\nu)\|^2 + \|s(\mu,\nu)\|^2}.$$

At this stage we define

$$\kappa(\zeta,\nu) := \frac{\sqrt{\|x(\mu,\nu)\|^2 + \|s(\mu,\nu)\|^2}}{\zeta\sqrt{2n}}, \quad 0 < \nu \leq 1,\ \mu = \mu^0\nu, \tag{40}$$

and

$$\bar{\kappa}(\zeta) = \max_{0<\nu\leq 1} \kappa(\zeta,\nu). \tag{41}$$

Then $\|q\|$ can be bounded above as follows:

$$\|q\| \leq \alpha\bar{\kappa}(\zeta)\sqrt{2}.$$

We found in (36) that in order to have $\delta(v^f) \leq 1/\sqrt{2}$, we should have $\|q\|^2 + (\|q\| + 2\delta)^2 \leq 1$. Therefore, since $\delta \leq \tau = \frac{1}{8}$, it suffices if q satisfies $\|q\|^2 + \left(\|q\| + \frac{1}{4}\right)^2 \leq 1$. So we certainly have $\delta(v^f) \leq 1/\sqrt{2}$ if $\|q\| \leq \frac{1}{2}$. Since $\|q\| \leq \alpha\bar{\kappa}(\zeta)\sqrt{2}$, the latter inequality is satisfied if we take

$$\alpha = \frac{1}{2\sqrt{2}\,\bar{\kappa}(\zeta)}. \tag{42}$$

Note that since $x(\zeta^2,1) = s(\zeta^2,1) = \zeta e$, we have $\kappa(\zeta,1) = 1$. As a consequence we obtain that $\bar{\kappa}(\zeta) \geq 1$. We can prove that $\bar{\kappa}(\zeta) \leq \sqrt{2n}$. This is shown in the next section.

4.6. Bound for $\bar{\kappa}(\zeta)$.

Due to the choice of the vectors $\bar{x}$, $\bar{y}$, $\bar{s}$ and the number ζ (cf. section 4.4) we have

$$\begin{aligned} A\bar{x} &= b, & 0 \leq \bar{x} \leq \zeta e,\\ A^T\bar{y} + \bar{s} &= c, & 0 \leq \bar{s} \leq \zeta e,\\ \bar{x}\bar{s} &= 0. \end{aligned}$$

To simplify notation in the rest of this section, we denote $x = x(\mu,\nu)$, $y = y(\mu,\nu)$, and $s = s(\mu,\nu)$. Then x, y, and s are uniquely determined by the system

$$\begin{aligned} b - Ax &= \nu(b - A\zeta e), & x \geq 0,\\ c - A^Ty - s &= \nu(c - \zeta e), & s \geq 0,\\ xs &= \nu\zeta^2 e. \end{aligned}$$

Hence we have

$$\begin{aligned} A\bar{x} - Ax &= \nu(A\bar{x} - A\zeta e), & x \geq 0,\\ A^T\bar{y} + \bar{s} - A^Ty - s &= \nu(A^T\bar{y} + \bar{s} - \zeta e), & s \geq 0,\\ xs &= \nu\zeta^2 e. \end{aligned}$$

We rewrite this system as

$$\begin{aligned} A\left(\bar{x} - x - \nu\bar{x} + \nu\zeta e\right) &= 0, & x \geq 0,\\ A^T\left(\bar{y} - y - \nu\bar{y}\right) &= s - \bar{s} + \nu\bar{s} - \nu\zeta e, & s \geq 0,\\ xs &= \nu\zeta^2 e. \end{aligned}$$

Using again that the row space of a matrix and its null space are orthogonal, we obtain

$$\left(\bar{x} - x - \nu\bar{x} + \nu\zeta e\right)^T\left(s - \bar{s} + \nu\bar{s} - \nu\zeta e\right) = 0.$$

Defining

$$a := (1-\nu)\bar{x} + \nu\zeta e, \quad b := (1-\nu)\bar{s} + \nu\zeta e,$$

we have $(a-x)^T(b-s) = 0$. This gives

$$a^Tb + x^Ts = a^Ts + b^Tx.$$

Since $\bar{x}^T\bar{s} = 0$, $\bar{x} + \bar{s} \le \zeta e$, and $xs = \nu\zeta^2 e$, we may write

$$\begin{aligned} a^Tb + x^Ts &= ((1-\nu)\bar{x} + \nu\zeta e)^T((1-\nu)\bar{s} + \nu\zeta e) + \nu\zeta^2 n \\ &= \nu(1-\nu)(\bar{x}+\bar{s})^T\zeta e + \nu^2\zeta^2 n + \nu\zeta^2 n \\ &\le \nu(1-\nu)(\zeta e)^T\zeta e + \nu^2\zeta^2 n + \nu\zeta^2 n \\ &= \nu(1-\nu)\zeta^2 n + \nu^2\zeta^2 n + \nu\zeta^2 n = 2\nu\zeta^2 n. \end{aligned}$$

Moreover, also using $a \ge \nu\zeta e$, $b \ge \nu\zeta e$, we get

$$\begin{aligned} a^Ts + b^Tx &= ((1-\nu)\bar{x} + \nu\zeta e)^T s + ((1-\nu)\bar{s} + \nu\zeta e)^T x \\ &= (1-\nu)\left(s^T\bar{x} + x^T\bar{s}\right) + \nu\zeta e^T(x+s) \\ &\ge \nu\zeta e^T(x+s) = \nu\zeta\left(\|s\|_1 + \|x\|_1\right). \end{aligned}$$

Hence we obtain $\|s\|_1 + \|x\|_1 \le 2\zeta n$. Since $\|x\|^2 + \|s\|^2 \le (\|s\|_1 + \|x\|_1)^2$, it follows that

$$\frac{\sqrt{\|x\|^2 + \|s\|^2}}{\zeta\sqrt{2n}} \le \frac{\|s\|_1 + \|x\|_1}{\zeta\sqrt{2n}} \le \frac{2\zeta n}{\zeta\sqrt{2n}} = \sqrt{2n},$$

thus proving

$$\bar{\kappa}(\zeta) \le \sqrt{2n}.$$

4.7. Complexity analysis. In the previous sections we have found that if at the start of an iteration the iterates satisfy $\delta(x,s;\mu) \le \tau$, with τ as defined in (39), then after the feasibility step, with θ as defined in (31), and α as in (42), the iterates satisfy $\delta(x^f, s^f; \mu^+) \le 1/\sqrt{2}$.

According to (24), at most

$$\log_2\left(\log_2\frac{1}{\tau^2}\right) = \log_2(\log_2 64) \le 3$$

centering steps suffice to get iterates (x^+, y^+, s^+) that satisfy $\delta(x^+, s^+; \mu^+) \le \tau$. So each main iteration consists of at most 4 so-called inner iterations, in each of which we need to compute a search direction (for either a feasibility step or a centering step).

It has become a custom to measure the complexity of an IPM by the required number of inner iterations. In each main iteration both the duality gap and the norms of the residual vectors are reduced by the factor $1-\theta$. Hence, using ${x^0}^Ts^0 = n\zeta^2$, the total number of main iterations is bounded above by

$$\frac{1}{\theta}\log\frac{\max\left\{n\zeta^2, \left\|r_b^0\right\|, \left\|r_c^0\right\|\right\}}{\varepsilon}.$$

Due to (31) and (42) we have

$$\theta = \frac{\alpha}{\sqrt{2n}} = \frac{1}{4\bar{\kappa}(\zeta)\sqrt{n}}.$$

Hence the total number of inner iterations is bounded above by

$$16\,\bar{\kappa}(\zeta)\sqrt{n}\,\log\frac{\max\left\{n\zeta^2,\,\left\|r_b^0\right\|,\,\left\|r_c^0\right\|\right\}}{\varepsilon}, \tag{43}$$

where $\bar{\kappa}(\zeta) \leq \sqrt{2n}$. Thus we may state without further proof the main result of the paper.

THEOREM 4.8. *If* (P) *and* (D) *are feasible and* $\zeta > 0$ *is such that* $\|x^* + s^*\|_\infty \leq \zeta$ *for some optimal solutions* x^* *of* (P) *and* (y^*, s^*) *of* (D), *then after at most*

$$16\sqrt{2}\,n\,\log\frac{\max\left\{n\zeta^2,\,\left\|r_b^0\right\|,\,\left\|r_c^0\right\|\right\}}{\varepsilon}$$

inner iterations the algorithm finds an ε*-solution of* (P) *and* (D).

A basic question is, of course, how to choose the number ζ, which determines the initial iterates and has to be fixed before starting the algorithm. A related question that we did not yet deal with is whether or not our algorithm can detect infeasibility (or unboundedness) of (P) and (D). In the case, where the entries of A, b, and c are rational numbers, these issues can be dealt with as follows. It is well known that if (P) and (D) are feasible, then there exist optimal solutions x^* and (y^*, s^*) of (P) and (D) such that $\|x^* + s^*\|_\infty \leq 2^L$, where L denotes the binary input size of (P) and (D). The number L can be computed straightforwardly from the input data A, b, and c. Thus, when starting the algorithm with $\zeta = 2^L$, after at most

$$16\sqrt{2}\,n\,\log\frac{\max\left\{n4^L,\,\left\|b - 2^L Ae\right\|,\,\left\|c - 2^L e\right\|\right\}}{\varepsilon}$$

iterations the algorithm finds an ε-solution if it exists. Otherwise we must decide that (P) and (D) are infeasible or unbounded.

Working with the number L may not be possible in practice, however, since this number can be very large. For such cases it may be worth noting that if (P) and (D) are infeasible or unbounded, then Lemma 3.1 implies that (P_ν) and (D_ν) do not satisfy the IPC for all $\nu \in (0, \bar{\nu}]$ for some $\bar{\nu} \in (0, 1)$. If the iterates after the feasibility step satisfy $\delta(x^f, s^f; \mu^+) \leq 1/\sqrt{2}$, we are sure that the perturbed problems corresponding to $\nu = \mu^+/\mu^0$ satisfy the IPC, and hence are strictly feasible. So we then have $\nu \geq \bar{\nu}$. On the other hand, if $\nu < \bar{\nu}$, the algorithm will find that $\delta(x^f, s^f; \mu^+) > 1/\sqrt{2}$, which implies that there are no optimal solutions x^* and (y^*, s^*) such that $\zeta \geq \|x^* + s^*\|_\infty$. We can then run the algorithm again with $\zeta := 2\zeta$. If necessary, this can be repeated. When starting with $\zeta = 1$, after doubling the value of ζ at most L times the algorithm must have found optimal solutions of (P) and (D) if these exist. Otherwise (P) and (D) are infeasible or unbounded.

5. Concluding remarks.

The current paper shows that the techniques that have been developed in the field of feasible full-Newton step IPMs, and which have now been known for almost 20 years, are sufficient to get a full-Newton step IIPM whose complexity is at least as good as the currently best known performance of

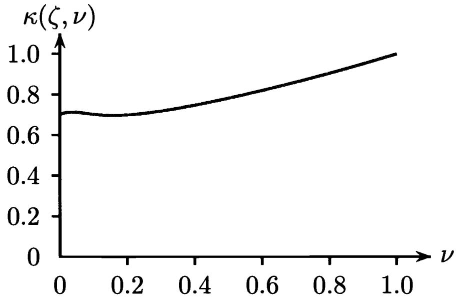


FIG. 4. *Typical behavior of $\kappa(\zeta,\nu)$ as a function of ν.*

IIPMs. Following a well-known metaphor of Isaac Newton,[4] it looks as if a "smooth pebble or pretty shell on the sea-shore of IPMs" has been overlooked for a surprisingly long time.

It is worth mentioning that we found computational evidence for the following conjecture.

CONJECTURE 5.1. *If* (P) *and* (D) *are feasible and* $\zeta \geq \|x^* + s^*\|_\infty$ *for some pair of optimal solutions* x^* *and* (y^*, s^*), *then* $\bar{\kappa}(\zeta) = 1$.

The evidence was provided by a simple MATLAB implementation of our algorithm. As input we used a primal-dual pair of randomly generated feasible problems with known optimal solutions x^* and (y^*, s^*) and ran the algorithm with $\zeta = \|x^* + s^*\|_\infty$. This was done for various sizes of the problems and for at least 10^5 instances. No counterexample for the conjecture was found. Typically, the graph of $\kappa(\zeta,\nu)$, as a function of ν, is as depicted in Figure 4. The importance of the conjecture is evident. Its trueness would reduce the currently best iteration bound for IIPMs by a factor $\sqrt{2n}$.

It may be clear that the full-Newton step method presented in this paper may not be efficient from a practical point of view, just as the feasible IPMs with the best theoretical performance are far from practical. But just as in the case of feasible IPMs, one might expect that computationally efficient large-update methods for IIPMs can be designed whose theoretical complexity is not worse than $\sqrt{n}$ times the iteration bound in this paper. Even better results for large-update methods might be obtained by changing the search direction, by using methods that are based on kernel functions, as presented in [3, 17]. This requires further investigation. Also, extensions to second-order cone optimization, semidefinite optimization, linear complementarity problems, etc. seem to be within reach.

Appendix. Some technical lemmas. Given a strictly primal feasible point x of (P) and a strictly dual feasible point (y, s) of (D), and $\mu > 0$, let

$$\Phi\left(xs;\mu\right) := \Psi(v) := \sum_{i=1}^{n} \psi(v_i), \quad v_i := \sqrt{\frac{x_i s_i}{\mu}}, \quad \psi(t) := \frac{1}{2}\left(t^2 - 1 - \log t^2\right).$$

It is well known that $\psi(t)$ is the kernel function of the primal-dual logarithmic barrier function, which, up to some constant, is the function $\Phi\left(xs;\mu\right)$ (see, e.g., [3]).

[4] "I do not know what I may appear to the world; but to myself I seem to have been only like a boy playing on the sea-shore, and diverting myself in now and then finding a smoother pebble or a prettier shell than ordinary, whilst the great ocean of truth lay all undiscovered before me" [27, p. 863].

Lemma A.1. *One has*

$$\Phi\left(xs;\mu\right)=\Phi\left(xs(\mu);\mu\right)+\Phi\left(x(\mu)s;\mu\right).$$

Proof. The equality in the lemma is equivalent to

$$\sum_{i=1}^{n}\left(\frac{x_is_i}{\mu}-1-\log\frac{x_is_i}{\mu}\right)$$
$$=\sum_{i=1}^{n}\left(\frac{x_i(\mu)s_i}{\mu}-1-\log\frac{x_i(\mu)s_i}{\mu}\right)+\sum_{i=1}^{n}\left(\frac{x_is_i(\mu)}{\mu}-1-\log\frac{x_is_i(\mu)}{\mu}\right).$$

Since

$$\log\frac{x_is_i}{\mu}=\log\frac{x_is_i}{x_i(\mu)s_i(\mu)}=\log\frac{x_i}{x_i(\mu)}+\log\frac{s_i}{s_i(\mu)}=\log\frac{x_is_i(\mu)}{\mu}+\log\frac{x_i(\mu)s_i}{\mu},$$

this equality holds if and only if

$$x^Ts-n\mu=\left(x^Ts(\mu)-n\mu\right)+\left(s^Tx(\mu)-n\mu\right).$$

Using that $x(\mu)s(\mu)=\mu e$, whence $x(\mu)^Ts(\mu)=n\mu$, this can be written as $(x-x(\mu))^T(s-s(\mu))=0$, which holds if the vectors $x-x(\mu)$ and $s-s(\mu)$ are orthogonal. This is indeed the case, because $x-x(\mu)$ belongs to the null space and $s-s(\mu)$ to the row space of A. This proves the lemma. □

Lemma A.2. *One has*

$$\psi\left(\sqrt{\frac{x_i}{x_i(\mu)}}\right)\leq\Psi(v),\quad\psi\left(\sqrt{\frac{s_i}{s_i(\mu)}}\right)\leq\Psi(v),\quad i=1,\ldots,n.$$

Proof. By Lemma A.1 we have $\Phi\left(xs;\mu\right)=\Phi\left(xs(\mu);\mu\right)+\Phi\left(x(\mu)s;\mu\right)$. Since $\Phi\left(xs;\mu\right)$ is always nonnegative, also $\Phi\left(xs(\mu);\mu\right)$ and $\Phi\left(x(\mu)s;\mu\right)$ are nonnegative. Thus it follows that $\Phi\left(xs(\mu);\mu\right)\leq\Psi(v)$ and $\Phi\left(x(\mu)s;\mu\right)\leq\Psi(v)$. The first of these two inequalities gives

$$\Phi\left(xs(\mu);\mu\right)=\sum_{i=1}^{n}\psi\left(\sqrt{\frac{x_is_i(\mu)}{\mu}}\right)\leq\Psi(v).$$

Since $\psi(t)\geq 0$, for every $t>0$, it follows that

$$\psi\left(\sqrt{\frac{x_is_i(\mu)}{\mu}}\right)\leq\Psi(v),\quad i=1,\ldots,n.$$

Due to $x(\mu)s(\mu)=\mu e$, we have

$$\frac{x_is_i(\mu)}{\mu}=\frac{x_is_i(\mu)}{x_i(\mu)s_i(\mu)}=\frac{x_i}{x_i(\mu)},$$

whence we obtain the first inequality in the lemma. The second inequality follows in the same way. □

Note that $\psi(t)$ is monotonically decreasing for $t\leq 1$ and monotonically increasing for $t\geq 1$. In what follows we denote by $\varrho:[0,\infty)\rightarrow[1,\infty)$ the inverse function of

$\psi(t)$ for $t \geq 1$ and by $\chi \ : \ [0,\infty) \to (0,1]$ the inverse function of $\psi(t)$ for $0 < t \leq 1$. So we have

$$\varrho(s) = t \Leftrightarrow s = \psi(t), \quad s \geq 0,\, t \geq 1, \tag{44}$$

and

$$\chi(s) = t \Leftrightarrow s = \psi(t), \quad s \geq 0,\, 0 < t \leq 1. \tag{45}$$

Note that $\chi(s)$ is monotonically decreasing and $\varrho(s)$ is monotonically increasing in $s \geq 0$.

Lemma A.3. *Let $t > 0$ and $\psi(t) \leq s$. Then $\chi\left(s\right) \leq t \leq \varrho\left(s\right)$.*

Proof. This is almost obvious. Since $\psi(t)$ is strictly convex and minimal at $t = 1$, with $\psi(1) = 0$, $\psi(t) \leq s$ implies that t belongs to a closed interval whose extremal points are $\chi(s)$ and $\varrho(s)$. □

Corollary A.4. *One has*

$$\chi\left(\Psi(v)\right) \leq \sqrt{\frac{x_i}{x_i(\mu)}} \leq \varrho\left(\Psi(v)\right), \quad \chi\left(\Psi(v)\right) \leq \sqrt{\frac{s_i}{s_i(\mu)}} \leq \varrho\left(\Psi(v)\right).$$

Proof. This is immediate from Lemma A.3 and Lemma A.2. □

Lemma A.5. *One has*

$$\sqrt{\frac{x}{s}} \leq \frac{\varrho\left(\Psi(v)\right)}{\chi\left(\Psi(v)\right)}\, \frac{x(\mu)}{\sqrt{\mu}}, \qquad \sqrt{\frac{s}{x}} \leq \frac{\varrho\left(\Psi(v)\right)}{\chi\left(\Psi(v)\right)}\, \frac{s(\mu)}{\sqrt{\mu}}.$$

Proof. Using that $x_i(\mu)s_i(\mu) = \mu$, for each i, Corollary A.4 implies

$$\sqrt{\frac{x_i}{s_i}} \leq \frac{\varrho\left(\Psi(v)\right)\sqrt{x_i(\mu)}}{\chi\left(\Psi(v)\right)\sqrt{s_i(\mu)}} = \frac{\varrho\left(\Psi(v)\right)}{\chi\left(\Psi(v)\right)}\sqrt{\frac{x_i(\mu)}{s_i(\mu)}} = \frac{\varrho\left(\Psi(v)\right)}{\chi\left(\Psi(v)\right)}\sqrt{\frac{x_i^2(\mu)}{\mu}} = \frac{\varrho\left(\Psi(v)\right)}{\chi\left(\Psi(v)\right)}\,\frac{x_i(\mu)}{\sqrt{\mu}},$$

which implies the first inequality. The second inequality is obtained in the same way. □

We proceed by deriving an upper bound for $\Psi(v)$ in terms of $\delta(v)$. First we deal with a simple lemma.

Lemma A.6. *Let $t \geq 1$. Then $\psi(t) - \psi\left(\frac{1}{t}\right) \geq 0$.*

Proof. Define $f(t) := \psi(t) - \psi\left(\frac{1}{t}\right)$ for $t > 0$. Then

$$f'(t) = t - \frac{1}{t} - \left(\frac{1}{t} - t\right)\frac{-1}{t^2} = \left(t - \frac{1}{t}\right)\left(1 - \frac{1}{t^2}\right) = \frac{\left(t^2-1\right)^2}{t^3} \geq 0.$$

It follows that $f(t)$ is monotonically increasing for $t > 0$. Since $f(1) = 0$ this proves that $f(t) \geq 0$ for $t \geq 1$, and hence the lemma follows. □

Theorem A.7. *Let $\delta(v)$ be as defined in* (10) *and $\rho(\delta)$ as in* (46). *Then $\Psi(v) \leq \psi\left(\rho(\delta(v))\right)$.*

Proof. The statement in the lemma is obvious if $v = e$, since then $\delta(v) = \Psi(v) = 0$ and since $\rho(0) = 1$ and $\psi(1) = 0$. Otherwise we have $\delta(v) > 0$ and $\Psi(v) > 0$. To deal with the nontrivial case we consider, for $\tau > 0$, the problem

$$z_\tau = \max_v \left\{ \Psi(v) = \sum_{i=1}^n \psi(v_i) \ : \ \delta(v)^2 = \tfrac{1}{4}\sum_{i=1}^n \psi'(v_i)^2 = \tau^2 \right\}.$$

The first-order optimality conditions are

$$\psi'(v_i) = \lambda\psi'(v_i)\psi''(v_i), \quad i = 1, \ldots, n,$$

where $\lambda \in \mathbf{R}$. From this we conclude that we have either $\psi'(v_i) = 0$ or $\lambda\psi''(v_i) = 1$ for each i. Since $\psi''(t)$ is monotonically decreasing, this implies that all v_i's for which $\lambda\psi''(v_i) = 1$ have the same value. Denoting this value as t, and observing that all other coordinates have value 1 (since $\psi'(v_i) = 0$ for these coordinates), we conclude that for some k, and after reordering the coordinates, v has the form

$$v = (\underbrace{t, \ldots, t}_{k \text{ times}}, \underbrace{1, \ldots, 1}_{n-k \text{ times}}).$$

Since $\psi'(1) = 0$, $\delta(v) = \tau$ implies $k\psi'(t)^2 = 4\tau^2$. Since $\psi'(t) = t - 1/t$, it follows that

$$t - \frac{1}{t} = \pm\frac{2\tau}{\sqrt{k}},$$

which gives $t = \rho(\tau/\sqrt{k})$ or $t = 1/\rho(\tau/\sqrt{k})$, where $\rho : \mathbf{R}_+ \to [1, \infty)$ is defined by

$$\rho(\delta) := \delta + \sqrt{1 + \delta^2}. \tag{46}$$

By Lemma A.6, the first value, which is greater than 1, gives the largest value of $\psi(t)$. Since we are maximizing $\Psi(v)$, we conclude that $t = \rho(\tau/\sqrt{k})$, whence we have

$$\Psi(v) = k\,\psi\left(\rho\left(\frac{\tau}{\sqrt{k}}\right)\right).$$

The question remains which value of k maximizes $\Psi(v)$. To investigate this we take the derivative of $\Psi(v)$ with respect to k. To simplify the notation we write

$$\Psi(v) = k\,\psi\left(t\right), \quad t = \rho\left(s\right), \quad s = \frac{\tau}{\sqrt{k}}.$$

The definition of t implies $t = s + \sqrt{1 + s^2}$. This gives $(t - s)^2 = 1 + s^2$, or $t^2 - 1 = 2st$, whence we have

$$2s = t - \frac{1}{t} = \psi'(t).$$

Some straightforward computations now yield

$$\frac{d\,\Psi(v)}{dk} = \psi\left(t\right) - \frac{s^2\rho(s)}{\sqrt{1 + s^2}} =: f(\tau).$$

We consider this derivative as a function of τ, as indicated. One may easily verify that $f(0) = 0$. We proceed by computing the derivative with respect to τ. This gives

$$f'(\tau) = -\frac{1}{\sqrt{k}} \frac{s^2}{(1+s^2)\sqrt{1+s^2}}.$$

This proves that $f'(\tau) \leq 0$. Since $f(0) = 0$, it follows that $f(\tau) \leq 0$ for each $\tau \geq 0$. Hence we conclude that $\Psi(v)$ is decreasing in k. So $\Psi(v)$ is maximal when $k = 1$, which gives the result in the theorem. □

Corollary A.8. *Let $\tau \geq 0$, $\delta(v) \leq \tau$, and $\rho(\delta)$ as defined in* (46). *Then $\Psi(v) \leq \tau'$, where*

$$\tau' := \psi\left(\rho(\tau)\right). \tag{47}$$

Proof. Since $\rho(\delta)$ is monotonically increasing in δ, and $\rho(\delta) \geq 1$ for all $\delta \geq 0$, and, moreover, $\psi(t)$ is monotonically increasing if $t \geq 1$, the function $\psi\left(\rho(\delta)\right)$ is increasing in δ for $\delta \geq 0$. Thus the result is immediate from Theorem A.7. □

Theorem A.9. *Let $\tau \geq 0$, $\delta(v) \leq \tau$, and τ' as defined in* (47). *Then*

$$\sqrt{\frac{x}{s}} \leq \frac{\varrho\left(\tau'\right)}{\chi\left(\tau'\right)} \frac{x(\mu)}{\sqrt{\mu}}, \qquad \sqrt{\frac{s}{x}} \leq \frac{\varrho\left(\tau'\right)}{\chi\left(\tau'\right)} \frac{s(\mu)}{\sqrt{\mu}}.$$

Proof. Since $\varrho(t)$ is monotonically increasing and $\chi(t)$ monotonically decreasing this is an immediate consequence of Lemma A.5 and Corollary A.8. □

Corollary A.10. *Let $\tau = \frac{1}{8}$ and $\delta(v) \leq \tau$. Then*

$$\sqrt{\frac{x}{s}} \leq \sqrt{2}\, \frac{x(\mu)}{\sqrt{\mu}}, \qquad \sqrt{\frac{s}{x}} \leq \sqrt{2}\, \frac{s(\mu)}{\sqrt{\mu}}.$$

Proof. If $\tau = \frac{1}{8}$, then $\tau' \approx 0.016921$, $\varrho\left(\tau'\right) \approx 1.13278$, and $\chi\left(\tau'\right) \approx 0.872865$, whence

$$\frac{\varrho\left(\tau'\right)}{\chi\left(\tau'\right)} \approx 1.29777 < \sqrt{2}.$$

Thus the result follows. □

Acknowledgments. Hossein Mansouri, Ph.D. student in Delft, forced the author to become more familiar with IIPMs when, in September 2004, he showed him a draft of a paper on large-update IIPMs based on kernel functions. Thank you, Hossein! I am also much indebted to Jean-Philippe Vial. He found a serious flaw in an early version of this paper. Thanks are also due Kurt Anstreicher, Florian Potra, Shinji Mizuno, and other colleagues for some useful discussions during the Oberwolfach meeting *Optimization and Applications* in January 2005. I also thank Yanqin Bai for pointing out some typos in an earlier version of the paper. Finally, I kindly acknowledge the useful comments of two anonymous referees that were very helpful in improving the readability of the paper.

REFERENCES

[1] E. D. Andersen and Y. Ye, *A computational study of the homogeneous algorithm for large-scale convex optimization*, Comput. Appl. Optim., 10 (1998), pp. 243–269.

[2] E. D. Andersen and K. D. Andersen, *The MOSEK interior-point optimizer for linear programming: An implementation of the homogeneous algorithm*, in High Performance Optimization Techniques, S. Zhang, H. Frenk, C. Roos, and T. Terlaky, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999, pp. 197–232.

[3] Y. Q. Bai, M. El Ghami, and C. Roos, *A comparative study of kernel functions for primal-dual interior-point algorithms in linear optimization*, SIAM J. Optim., 15 (2004), pp. 101–128.

[4] S. C. Billups and M. C. Ferris, *Convergence of an infeasible interior-point algorithm from arbitrary positive starting points*, SIAM J. Optim., 6 (1996), pp. 316–325.

[5] J. F. Bonnans and F. A. Potra, *On the convergence of the iteration sequence of infeasible path following algorithms for linear complementarity problems*, Math. Oper. Res., 22 (1997), pp. 378–407.

[6] R. M. Freund, *A potential-function reduction algorithm for solving a linear program directly from an infeasible "warm start"*, Math. Programming, 52 (1991), pp. 441–466.

[7] B. Jansen, C. Roos, and T. Terlaky, *A polynomial Dikin–type primal–dual algorithm for linear programming*, Math. Oper. Res., 21 (1996), pp. 341–353.

[8] N. Karmarkar, *New polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[9] M. Kojima, N. Megiddo, and S.Mizuno, *A primal-dual infeasible-interior-point algorithm for linear programming*, Math. Programming, 61 (1993), pp. 263–280.

[10] M. Kojima, T. Noma, and A. Yoshise, *Global convergence in infeasible-interior-point algorithms*, Math. Programming, 65 (1994), pp. 43–72.

[11] I. J. Lustig, *Feasible issues in a primal-dual interior-point method*, Math. Programming, 67 (1990), pp. 145–162.

[12] S. Mizuno, *Polynomiality of infeasible-interior-point algorithms for linear programming*, Math. Programming, 67 (1994), pp. 109–119.

[13] S. Mizuno, M. J. Todd, and Y. Ye, *A surface of analytic centers and infeasible- interior-point algorithms for linear programming*, Math. Oper. Res., 20 (1995), pp. 135–162.

[14] R. D. C. Monteiro, I. Adler, and M. G. C. Resende, *A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension*, Math. Oper. Res., 15 (1990), pp. 191–214.

[15] Yu. Nesterov, M. J. Todd, and Y. Ye, *Infeasible-start primal-dual methods and infeasibility detectors for nonlinear programming problems*, Math. Programming, 84 (1999), pp. 227–267.

[16] Yu. Nesterov and M. J. Todd, *On the Riemannian geometry defined by self-concordant barriers and interior-point methods*, Found. Comput. Math., 2 (2002), pp. 333–361.

[17] J. Peng, C. Roos, and T. Terlaky, *Self-Regularity. A New Paradigm for Primal-Dual Interior-Point Algorithms*, Princeton University Press, Princeton, NJ, 2002.

[18] F. A. Potra, *A quadratically convergent predictor-corrector method for solving linear programs from infeasible starting points*, Math. Programming, 67 (1994), pp. 383–406.

[19] F. A. Potra, *An infeasible-interior-point predictor-corrector algorithm for linear programming*, SIAM J. Optim., 6 (1996), pp. 19–32.

[20] C. Roos, T. Terlaky, and J.-Ph. Vial, *Theory and Algorithms for Linear Optimization. An Interior-Point Approach*, John Wiley & Sons, Chichester, UK, 1997. Revised edition: *Interior-Point Methods for Linear Optimization*. Springer, New York, 2005.

[21] M. Salahi, T. Terlaky, and G. Zhang, *The complexity of self-regular proximity based infeasible IPMs*, Comput. Optim. Appl., to appear.

[22] R. Sheng and F. A. Potra, *A quadratically convergent infeasible-interior-point algorithm for LCP with polynomial complexity*, SIAM J. Optim., 7 (1997), pp. 304–317.

[23] J. F. Sturm, *Using SeDuMi* 1.02*, a MATLAB toolbox for optimization over symmetric cones*, Optim. Methods Softw., 11 (1999), pp. 625–653.

[24] K. Tanabe, *Centered Newton method for linear programming: Interior and "exterior" point method*, in New Methods for Linear Programming, K. Tone, ed., The Institute of Statistical Mathematics, Tokyo, Japan, 1990, pp. 98–100 (in Japanese).

[25] M. J. Todd and Y. Ye, *Approximate Farkas lemmas and stopping rules for iterative infeasible-point algorithms for linear programming*, Math. Programming, 81 (1998), pp. 1–21.

[26] R. J. Vanderbei, *Linear Programming: Foundations and Extensions*, Kluwer Academic Publishers, Boston, 1996.

[27] R. S. Westfall, *Never at Rest. A Biography of Isaac Newton*, Cambridge University Press, Cambridge, UK, 1964.

[28] S. J. Wright, *An infeasible-interior-point algorithm for linear complementarity problems*, Math. Programming, 67 (1994), pp. 29–52.

[29] S. J. Wright, *A path-following infeasible-interior-point algorithm for linear complementarity problems*, Optim. Methods Softw., 2 (1993), pp. 79–106.

[30] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1996.

[31] F. Wu, S. Wu, and Y. Ye, *On quadratic convergence of the $O(\sqrt{n}L)$-iteration homogeneous*

and self-dual linear programming algorithm, Ann. Oper. Res., 87 (1999), pp. 393–406.

[32] Y. Ye, *Interior Point Algorithms, Theory and Analysis*, John Wiley & Sons, Chichester, UK, 1997.

[33] Y. Ye, M. J. Todd, and S. Mizuno, *An $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm*, Math. Oper. Res., 19 (1994), pp. 53–67.

[34] Y. Zhang, *On the convergence of a class of infeasible interior-point methods for the horizontal linear complementarity problem*, SIAM J. Optim., 4 (1994), pp. 208–227.

SIAM J. OPTIM.
Vol. 16, No. 4, pp. 1137–1154

A TRUNCATED PROJECTED NEWTON-TYPE ALGORITHM FOR LARGE-SCALE SEMI-INFINITE PROGRAMMING[*]

QIN NI[†], CHEN LING[‡], LIQUN QI[§], AND KOK LAY TEO[¶]

Abstract. In this paper, a truncated projected Newton-type algorithm is presented for solving large-scale semi-infinite programming problems. This is a hybrid method of a truncated projected Newton direction and a modified projected gradient direction. The truncated projected Newton method is used to solve the constrained nonlinear system. In order to guarantee global convergence, a robust loss function is chosen as the merit function, and the projected gradient method inserted is used to decrease the merit function. This algorithm is suitable for handling large-scale problems and possesses superlinear convergence rate. The global convergence of this algorithm is proved and the convergence rate is analyzed. The detailed implementation is discussed, and some numerical tests for solving large-scale semi-infinite programming problems, with examples up to 2000 decision variables, are reported.

Key words. semi-infinite programming, Karush–Kuhn–Tucker system, large-scale problem

AMS subject classifications. 90C34, 90C06, 90C90, 65K05, 49M05

DOI. 10.1137/040619867

1. Introduction.

We consider the semi-infinite programming (SIP) problem

$$\min\{f(x),\ x \in X\}, \tag{1.1}$$

where $X = \{x \in \Re^n :\ g(x,v) \leq 0 \text{ for all } v \in \Omega\}$, $f(x) : \Re^n \to \Re$, and $g : \Re^n \times \Re^m \to \Re$ are twice continuously differentiable functions. In this paper, we assume that Ω is a nonempty compact box with

$$\Omega = \{v \in \Re^m :\ a \leq v \leq b\},$$

where $a \in \Re^m$, $b \in \Re^m$, and $a < b$.

Such an SIP problem has wide applications such as approximation theory, optimal control, eigenvalue computation, mechanical stress of materials, and statistical design. Many methods have been proposed for the SIP problem. We refer readers to [4, 6, 11, 12, 15, 17] for details.

Some large-scale SIP problems arise from the modeling of optimal control and approximation (see [5, 16, 19]). In order to increase the control precision in an optimal control problem, one should increase the number of switching points. That is, the larger the number of switching points set, the higher the control precision. If one sets

[*] Received by the editors November 30, 2004; accepted for publication (in revised form) November 7, 2005; published electronically February 15, 2006.
http://www.siam.org/journals/siopt/16-4/61986.html

[†] Department of Mathematics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, People's Republic of China (niqfs@nuaa.edu.cn). This author's work was supported by the National Natural Science Foundation of China (grant 10471062).

[‡] School of Information, Zhejiang University of Finance and Economics, Hangzhou, 310012, China (linghz@hzcnc.com). Current address: Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong (01902146r@polyu.edu.hk).

[§] Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong (maqilq@polyu.edu.hk). This author's work was supported by the Hong Kong Research Grant Council.

[¶] Department of Mathematics and Statistics, Curtin University of Technology, Bentley, WA 6102, Australia (K.L.Teo@curtin.edu.au).

a large number of switching points, the discretization of the control space will lead to large-scale SIP problems. In approximation theory, if a function $f(v)$ is approximated on the interval $[a, b]$ by a polynomial

$$f_N(v) = \sum_{j=1}^{N} x_j v^{j-1}$$

and the approximation is in the Chebyshev norm, then we get a SIP problem. It is clear that the larger the order of the polynomial, the higher the approximation precision. When a very high-order polynomial is used to approximate f on $[a, b]$, a large-scale SIP problem is generated. However, some efficient algorithms for small-scale SIP problems do not directly translate into algorithms for large-scale SIP problems.

In this paper, we extend a smoothing projected Newton-type algorithm proposed in [14] to solve large-scale SIP problems. The smoothing projected Newton-type algorithm proposed in [14] enjoys global and locally superlinear convergence. However, it is not suitable for large-scale SIP problems. We modify this algorithm in two aspects. First, a truncated solution of the system is determined by an iterative method, in which the computation of the matrix-vector product, instead of the matrix factorization, is used such that the implementation at each iteration is relatively simple and time-economic. Second, in order to guarantee the global convergence, a robust loss function [7] is chosen as the merit function and the projected gradient method inserted is used to decrease the merit function. This loss function uses a measure which does not weigh very large components of the variable heavily. Numerical results show that this loss function is a good merit function. This modified algorithm is called a truncated projected Newton-type algorithm and is suitable for handling large-scale problems. The global convergence of this algorithm is proved and the superlinear convergence rate is analyzed. The detailed implementation is discussed, and some numerical tests for solving large-scale SIP problems, with examples up to 2000 decision variables, are reported.

This paper is organized as follows. We present a truncated projected Newton-type algorithm in section 2. The convergence of the algorithm is analyzed in section 3 and numerical tests are given in section 4. We propose some comments in section 5.

For convenience, we denote $\nabla_x^T = (\frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_n})$ and $\nabla_x = (\frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_n})^T$ for $x \in \Re^n$. For a smoothing function $\Phi : \Re^n \to \Re^m$, we denote $\nabla_x^T \Phi = (\frac{\partial \Phi}{\partial x_1}, \dots, \frac{\partial \Phi}{\partial x_n})$ and $\nabla_x \Phi \equiv \nabla_x \Phi^T = (\nabla_x \Phi_1, \dots, \nabla_x \Phi_m)$. For $u \in \Re^n$ and $v \in \Re^m$, we denote by (u, v) the column vector $(u^T, v^T)^T$ in $\Re^{n+m}$.

2. A truncated projected Newton-type algorithm.

In order to describe our algorithm, we recall some notation and definitions in [14].

Let

$$V(x) = \{v \in \Omega : \ g(x, v) = 0\}.$$

If there exists a vector $d \in \Re^n$ such that

$$\nabla_x g(x, v)^T d < 0$$

for all $v \in V(x)$, then we say that an extended Mangasarian–Fromovitz constraint qualification (EMFCQ) holds at x. It is well known that if x is a local minimizer of the SIP problem (1.1), and EMFCQ holds at x, then the KKT system of the SIP problem (1.1) is as follows:

$$\begin{aligned}&\nabla f(x)+\sum_{j=1}^{p}u_j\nabla_x g(x,v^j)=0,\\ &g(x,v)\le 0\ \ \forall\ v\in\Omega,\\ &u_j>0,\ \ g(x,v^j)=0,\quad j=1,\dots,p,\end{aligned}\tag{2.1}$$

where $v^j\in V(x)$ for $j=1,\dots,p$ and $p\le n$.

By the definition of $V(x)$ and the first inequality of (2.1), $v^j\in V(x)$ $(j=1,\dots,p)$ imply that v^j $(j=1,\dots,p)$ are global minimizers of the following minimization problem:

$$\begin{aligned}\min\quad & -g(x,v)\\ \text{subject to}\quad & v\in\Omega.\end{aligned}\tag{2.2}$$

The KKT system of (2.2) can be written as

$$(v'-v)^T(-\nabla_v g(x,v))\ge 0\quad \forall\ v'\in\Omega,$$

and it can be reformulated as a system of nonsmooth equations as follows:

$$v-\mathrm{mid}(a,b,v+\nabla_v g(x,v))=0,\tag{2.3}$$

where the mid function is defined for all $i=1,\dots,m$ as

$$(\mathrm{mid}(c,d,w))_i=\begin{cases}c_i & \text{if } w_i<c_i,\\ w_i & \text{if } c_i\le w_i\le d_i,\\ d_i & \text{if } d_i<w_i.\end{cases}$$

The system of nonsmooth equations (2.3) can be approximated by

$$(\phi(t,x,v))_i=v_i-\varphi(t,a_i,b_i,v_i+(\nabla_v g(x,v))_i),\qquad i=1,\dots,m,\tag{2.4}$$

where

$$\varphi(t,c,d,w)=\frac{c+\sqrt{(c-w)^2+4t^2}}{2}+\frac{d-\sqrt{(d-w)^2+4t^2}}{2}$$

is the Chen–Harker–Kanzow–Smale smoothing function for $\mathrm{mid}(c,d,w)$. It is clear that ϕ is smooth for $t\ne 0$. In order to handle the first constrained condition of (2.1), Teo, Rehbock, and Jennings [20] used a nonsmooth function

$$G(x)=\int_V [g(x,v)]_+dv,\tag{2.5}$$

where $[x]_+=\max\{0,x\}$. This is approximated by the smoothing function

$$G(t,x)=\int_\Omega \frac{\sqrt{g^2(x,v)+4t^2}+g(x,v)}{2}dv,\tag{2.6}$$

which is defined in [14]. Hence (2.1) can be approximated by

$$\begin{aligned} &\nabla f(x) + \sum_{j=1}^{p} u_i \nabla_x g(x, v^j) = 0, \\ &u_j > 0, \quad g(x, v^j) = 0, \quad j = 1, \ldots, p, \\ &\phi(t, x, v^j) = 0, \qquad j = 1, \ldots, p, \\ &G(t, x) = 0. \end{aligned} \tag{2.7}$$

Now we define

$$L(x, u, V) = f(x) + \sum_{j=1}^{p} u_j g(x, v^j), \tag{2.8}$$

where $V = (v^1, v^2, \ldots, v^p) \in \Re^{mp}$ and $u = (u_1, \ldots, u_p)^T \in \Re^p$, and denote

$$\mathbf{g}(x, V) = \begin{bmatrix} g(x, v^1) \\ g(x, v^2) \\ \cdots \\ g(x, v^p) \end{bmatrix}, \quad \bar{\phi}(t, x, V) = \begin{bmatrix} \phi(t, x, v^1) \\ \phi(t, x, v^2) \\ \cdots \\ \phi(t, x, v^p) \end{bmatrix}. \tag{2.9}$$

In order to balance the number between equations and variables, we add an artificial variable y. By simple analysis, we can know that the KKT system of the SIP problem (1.1) can be reformulated as a equivalent system of constrained equations in the following:

$$\begin{aligned} &\Phi(w) = 0, \\ &u \geq 0, \qquad y \geq 0, \end{aligned} \tag{2.10}$$

where $w = (t, z) = (t, x, u, V, y) \in \Re \times \Re^n \times \Re^p \times \Re^{mp} \times \Re$, and

$$\Phi(w) = \begin{bmatrix} t \\ H(w) \end{bmatrix}, \quad H(w) = \begin{bmatrix} \nabla_x L(x, u, V) \\ \mathbf{g}(x, V) \\ \bar{\phi}(t, x, V) \\ G(t, x) + y \end{bmatrix}. \tag{2.11}$$

For convenience, we denote $w = (t, x, u, V, y) \in \Re^{\tilde{n}}$, $\tilde{n} = n+2+(m+1)p$. The function $\Phi(w)$ has the following property.

Lemma 2.1 (see [14]). *$\Phi(w) = \Phi(t, z)$ is smooth at (t, z) with $t \neq 0$ and semismooth at $(0, z)$.*

For the meaning of semismoothness we refer readers to [10, 13].

The problem (2.10) was established and a smoothing projected Newton-type algorithm was proposed for solving this problem in [14]. In the smoothing projected Newton-type algorithm in [14], the Newton direction is obtained by solving the following linear system:

$$\Phi(w_k) + \nabla^T \Phi(w_k) \Delta w_k = \beta_k \bar{w}, \tag{2.12}$$

where $\Delta w_k = (\Delta t_k, \Delta x_k, \Delta u_k, \Delta V_k, \Delta y_k) \in \Re^{\tilde{n}}$, $\bar{w} = (\bar{t}, 0)$, $\bar{t} > 0$, and

$$\tag{2.13}
\begin{aligned}
&\nabla^T \Phi(w_k)\\
&= \begin{bmatrix}
1 & 0_{1\times n} & 0_{1\times p} & 0_{1\times m} & \cdots & 0_{1\times m} & 0 \\
0_{n\times 1} & \nabla_x^2 L(x,u,V) & \nabla_x \mathbf{g}^T(x,V) & u_1 \nabla_{v^1}^T(\nabla_x g(x,v^1)) & \cdots & u_p \nabla_{v^p}^T(\nabla_x g(x,v^p)) & 0_{n\times 1} \\
0 & \nabla_x^T g(x,v^1) & 0_{1\times p} & \nabla_{v^1}^T g(x,v^1) & \cdots & 0_{1\times m} & 0 \\
0 & \nabla_x^T g(x,v^2) & 0_{1\times p} & 0_{1\times m} & \cdots & 0_{1\times m} & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & \nabla_x^T g(x,v^p) & 0_{1\times p} & 0_{1\times m} & \cdots & \nabla_{v^p}^T g(x,v^p) & 0 \\
\nabla_t \phi(t,x,v^1) & \nabla_x^T \phi(t,x,v^1) & 0_{p\times p} & \nabla_{v^1}^T \phi(t,x,v^1) & \cdots & 0_{1\times m} & 0_{p\times 1} \\
\nabla_t \phi(t,x,v^2) & \nabla_x^T \phi(t,x,v^2) & 0_{p\times p} & 0_{1\times m} & \cdots & 0_{1\times m} & 0_{p\times 1} \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\nabla_t \phi(t,x,v^p) & \nabla_x^T \phi(t,x,v^p) & 0_{p\times p} & 0_{1\times m} & \cdots & \nabla_{v^p}^T \phi(t,x,v^p) & 0_{p\times 1} \\
\nabla_t G(t,x) & \nabla_x^T G(t,x) & 0_{1\times p} & 0_{1\times m} & \cdots & 0_{1\times m} & 1
\end{bmatrix}.
\end{aligned}$$

We remark that from (2.13) we see that the last row of the matrix $\nabla^T \Phi(w_k)$ is independent of other rows, so the introduction of artificial variable y can reduce the possible degeneration generated by the function $G(t,x)$. In order to solve large-scale problems, we determine an inexact solution of (2.12) by using a restarted generalized minimum residual algorithm (GMRES($\tilde{m}$)) [3]. Here we use $\tilde{m}$ other than usual m because there is an another meaning for m in this paper. The vector Δw_k is called a truncated solution of (2.12) if

$$\tag{2.14} \|\Phi(w_k) + \nabla^T \Phi(w_k)\Delta w_k - \beta_k \bar{w}\| \le r_k$$

for $r_k > 0$.

In [14], a simple merit function

$$\tag{2.15} \Psi(w) = \frac{1}{2}\sum_{j=1}^{\tilde{n}} \Phi_j^2(w)$$

is chosen and its gradient is

$$\tag{2.16} \nabla \Psi(w) = \nabla \Phi(w) \Phi(w).$$

In order to solve the large-scale SIP problem, we consider the following function:

$$\tag{2.17} \Psi_h(w) = \sum_{j=1}^{\tilde{n}} \rho_{h_j}(\Phi_j(w)),$$

where

$$\rho_{h_j}(\xi) = \begin{cases} \xi^2/2 & \text{if } |\xi| \le h_j \\ h_j|\xi| - h_j^2/2 & \text{otherwise,} \end{cases}$$

h_j, $j = 1, \ldots, \tilde{n}$, are positive constants, and $\rho_{h_j}(\xi)$ is linear in ξ for $|\xi| > h_j$. This function was proposed by Huber and Dutter (see [7] and [2]) for solving the least squares problem. The measure $\rho(\xi)$ in this function does not weigh very large components of ξ heavily.

We use the function (2.17) as the merit function. The gradient of this function $\Psi_h(w)$ is

$$\nabla\Psi_h(w) = \nabla\Phi(w)\Phi_h(w), \tag{2.18}$$

where

$$\Phi_h(w) = \sum_{j\in J_h} \Phi_j(w)e_j + \sum_{j\in K_h} \mathrm{sign}(\Phi_j(w))h_j e_j, \tag{2.19}$$

$$J_h = \{j :\ 1 \le j \le \tilde{n},\ |\Phi_j(w)| \le h_j\}, \quad K_h = \{1,\dots,\tilde{n}\}/J_h. \tag{2.20}$$

The problem (2.10) is equivalent to finding a global solution of the following minimization problem:

$$\begin{array}{cc} \min & \Psi_h(w) \\ \text{subject to} & u \ge 0,\ y \ge 0. \end{array} \tag{2.21}$$

We call w a stationary point of (2.21) if it satisfies

$$\|\bar{d}_G(1)\| = 0, \tag{2.22}$$

where

$$\bar{d}_G(1) = \Pi_W(w - \gamma\nabla\Psi_h(w)) - w = \left[\begin{array}{c} -\gamma\nabla_t\Psi_h(w) \\ \Pi_Z(z - \gamma\nabla_z\Psi_h(w)) - z \end{array}\right], \tag{2.23}$$

$\gamma > 0$ is a constant, and $\Pi_W(\cdot)$ is an orthogonal projection operator onto W,

$$\begin{aligned} W &= \{w = (t, x, u, V, y) \in \Re^{\tilde{n}} :\ u \ge 0, y \ge 0\}, \\ Z &= \{z = (x, u, V, y) \in \Re^{\tilde{n}-1} :\ u \ge 0, y \ge 0\}. \end{aligned}$$

Let $\alpha \in (0, 1)$ be a constant. For a sequence $\{w_k\}_{k=0}^{\infty}$, we define

$$\beta_0 = \beta(w^0) = \alpha \min\{1, \|\bar{d}_G^0(1)\|^2\},$$

and

$$\beta_k = \beta(w^k) := \begin{cases} \beta_{k-1} & \text{if } \alpha\min\{1, \|\bar{d}_G^k(1)\|^2\} > \beta_{k-1}, \\ \alpha\min\{1, \|\bar{d}_G^k(1)\|^2\} & \text{otherwise.} \end{cases} \tag{2.24}$$

Now we state our truncated projected Newton-type algorithm for solving (2.10) below.

Algorithm 2.1.

Step 0. (Initialization)

Choose constants $\eta, \rho, \sigma \in (0, 1)$ with $\sigma\eta < 1$, $p_1 > 0$, $p_2 > 2$ and $\alpha \in (0, 1)$, $\bar{t} > 0$ with $\alpha\bar{t} < 1$, $h_j \ge 1$, $j = 1, \dots, \tilde{n}$. Let $\bar{w} = (\bar{t}, 0, 0, 0, 0)$, $t_0 = \bar{t}$, and $w^0 = (t_0, x^0, u^0, V^0, y^0)$ with $u_i^0 \ge 0$ $(i = 1, \dots, p)$, $y^0 \ge 0$. Set $k = 0$.

Step 1. (Termination Test)

Compute

$$\xi_k = \min\left\{1, \frac{t_k}{|t_k + \nabla_t H(w_k)H_h(w_k)|}, \frac{\eta\|\Phi(w_k)\|}{\|\nabla\Psi_h(w_k)\|}\right\}, \tag{2.25}$$

where $H_h(w_k)$ is obtained by removing just the first element of $\Phi_h(w)$ (see (2.19)).

$$\gamma_k = \begin{cases} \min\left\{\xi_k, \frac{\eta\Psi_h(w_k)}{\|\nabla\Psi_h(w_k)\|^2}\right\} & \text{if } |\Phi_j(w_k)| \le h_j,\ j = 1, 2, \dots, \tilde{n}, \\ \xi_k & \text{otherwise.} \end{cases} \tag{2.26}$$

If $\bar{d}_G^k(1) = 0$, then stop; otherwise compute β_k by (2.24) and go to Step 2.

Step 2. (Search Directions)

2.1. Compute the negative gradient direction. Compute

$$d_G^k = -\gamma_k \nabla\Psi_h(w_k) + \beta_k \bar{w}. \tag{2.27}$$

If

$$|\Phi_j(w_k)| \le h_j, \quad j = 1, 2, \dots, \tilde{n}, \tag{2.28}$$

then go to Step 2.2; otherwise set $d_{tN}^k = d_G^k$ and go to Step 3.

2.2. Compute the truncated Newton direction. Determine Δw_k, which satisfies

$$\|\Phi(w_k) + \nabla^T\Phi(w_k)\Delta w_k - \beta_k\bar{w}\| = o(\Psi_h(w^k)), \tag{2.29}$$

and set $d_{tN}^k = \Delta w_k$.

Step 3. (Line Search)

Let m_k be the smallest nonnegative integer m satisfying

$$\Psi_h(w^k + \bar{d}^k((\rho)^m)) \le \Psi_h(w^k) + \sigma\nabla\Psi_h(w^k)^T\tilde{d}_G^k((\rho)^m), \tag{2.30}$$

where for any $\lambda \in [0, 1]$,

$$\bar{d}^k(\lambda) = \tau^*(\lambda)\tilde{d}_G^k(\lambda) + (1 - \tau^*(\lambda))\tilde{d}_{tN}^k(\lambda). \tag{2.31}$$

Here

$$\tilde{d}_G^k(\lambda) := \Pi_W(w_k + \lambda d_G^k) - w_k, \quad \tilde{d}_{tN}^k(\lambda) := \Pi_W(w_k + \lambda d_{tN}^k) - w_k, \tag{2.32}$$

and $\tau^*(\lambda)$ is a solution of the following minimization problem:

$$\min_{\tau\in[0,1]} \frac{1}{2}\|\Phi(w_k) + \Phi'(w_k)[\tau\tilde{d}_G^k(\lambda) + (1-\tau)\tilde{d}_{tN}^k(\lambda)]\|^2.$$

Let $\lambda_k = (\rho)^{m_k}$ and $w^{k+1} = w^k + \bar{d}^k(\lambda_k)$.

Step 4. Set $k = k + 1$ and go to Step 1.

Remarks. (1) Algorithm 2.1 is able to handle sparse large-scale SIP problems. In Step 2.2, a truncated solution of the problem (2.12) is determined by using GMRES($\tilde{m}$) method. Hence, the matrix factorizations are avoided, because this iterative algorithm requires computing only matrix-vector products. If the SIP problem possesses the sparse data structure, the computation of the matrix, $\nabla^T\Phi(w_k)$, can take advantage of the sparsity of $\nabla_x^2 L(x_k, u_k, V_k)$. Therefore Algorithm 2.1 is applicable to the sparse large-scale SIP problem.

(2) If the condition (2.28) is not satisfied, then only the projected negative gradient direction is generated in the iteration; otherwise Step 2.2 is carried out and

mixed projected directions are generated. In addition, if (2.28) is satisfied, then $\Psi_h(w) = \Psi(w)$ holds.

(3) The condition (2.29) guarantees the convergence of Algorithm 2.1, which is discussed in the next section. In the implementation of the algorithm, one kind of choice of the right side in (2.29) is $\frac{1}{k+1}\min\{1, \Psi_h(w_k)\}$.

(4) $\tau^*(\lambda)$ is easily obtained, and we refer readers to [14].

(5) Another line search technique in Step 3 can be used if only the projected negative gradient is the search direction. Although it does not affect the convergence and its proof, it can decrease the number of inner iterations. In section 4 we give a detailed description.

(6) We remark that $G(t, x)$ in (2.11) and its derivative are not evaluated exactly. We use Newton–Cotes formulas (Simpson's rule) for approximating the integral and choose n_i equally spaced points in the interval $[a_i, b_i]$ such that $(b_i - a_i)/n_i \leq 0.05$ is satisfied, where $i = 1, \ldots, m$. Numerical results show that this choice is proper.

3. Convergence analysis.

In this section we discuss the convergence property of Algorithm 2.1. From the definition of β_k, the following lemma is obvious.

LEMMA 3.1. *$\{\beta_k\}$ defined in (2.24) has the following properties:*

(i) *$\{\beta_k\}$ is a nonincreasing sequence.*

(ii) *For all k, β_k satisfies*

$$\beta_k \leq \alpha \min\{1, \|\bar{d}_G^k(1)\|^2\}.$$

In the following we give the descent property of $\tilde{d}_G^k(\lambda)$ in Algorithm 2.1.

LEMMA 3.2. *Suppose that $w_k = (t^k, z^k) \in W$ with $t^k > 0$ is not a stationary point of (2.21). Then for any $\lambda \in (0, 1]$, it holds that*

$$\nabla\Psi_h(w_k)^T \tilde{d}_G^k(\lambda) \leq -\frac{\lambda}{\xi_k}(1 - \alpha\bar{t}\,)\|\bar{d}_G^k(1)\|^2 < 0. \tag{3.1}$$

Proof. In this proof, for simplicity, we drop the superscript k. For any $w = (t, z) \in W$ with $t > 0$, suppose that w is not a stationary point of (2.21). Then

$$\nabla\Psi_h(w) = \nabla\Phi(w)\Phi_h(w) = \left[\begin{array}{c} \tilde{t} + \nabla_t H^T(w) H_h(w) \\ \nabla_z H(w) H_h(w) \end{array}\right] \equiv \left[\begin{array}{c} \nabla_t \Psi_h(w) \\ \nabla_z \Psi_h(w) \end{array}\right],$$

where $\tilde{t} = \min(t, h_1)$, $\nabla_t H^T(w)$ is the first row of $\nabla H(w)$, $\nabla_z H(w)$ is the submatrix of $\nabla H(w)$ obtained by removing just the first row of $\nabla H(w)$, and $H_h(w)$ is obtained by removing just the first element of $\Phi_h(w)$ (see (2.19)). From (2.32) and the definition of projection in (2.23), $\tilde{d}_G(\lambda)$ can be written as

$$\begin{aligned} \tilde{d}_G(\lambda) \equiv \left[\begin{array}{c} (\tilde{d}_G(\lambda))_t \\ (\tilde{d}_G(\lambda))_z \end{array}\right] &= \Pi_W(w - \lambda\gamma\nabla\Psi_h(w) + \lambda\beta\bar{w}) - w \\ &= \left[\begin{array}{c} -\lambda\gamma(\tilde{t} + \nabla_t H^T(w) H_h(w)) + \lambda\beta\bar{t} \\ \Pi_Z(z - \lambda\gamma\nabla_z\Psi_h(w)) - z \end{array}\right]. \end{aligned}$$

Then we have

$$\begin{aligned}
&(\tilde t + \nabla_t H^T(w)H_h(w))[-\lambda\gamma(\tilde t + \nabla_t H(w)H_h(w)) + \lambda\beta\bar t]\\
&= -\lambda\gamma\|\tilde t + \nabla_t H(w)H_h(w)\|^2 + \lambda(\tilde t + \nabla_t H(w)H_h(w))\beta\bar t\\
&\le -\frac{\lambda}{\gamma}\|\gamma\nabla_t\Psi_h(w)\|^2 + \frac{\lambda}{\gamma}\|\gamma\nabla_t\Psi_h(w)\|\beta\bar t\\
&\le -\frac{\lambda}{\gamma}\|\gamma\nabla_t\Psi_h(w)\|^2 + \frac{\lambda}{\gamma}\|\gamma\nabla_t\Psi_h(w)\|(\alpha\bar t)\|\bar d_G(1)\|\\
&\le -\frac{\lambda}{\gamma}\|\gamma\nabla_t\Psi_h(w)\|^2 + \alpha\bar t\frac{\lambda}{\gamma}\|\bar d_G(1)\|^2,
\end{aligned}\tag{3.2}$$

where the second inequality comes from Lemma 3.1(ii) and the fact that $\beta \le \alpha\|\bar d_G(1)\|$, the last inequality, is due to $\|\gamma\nabla_t\Psi_h(w)\| \le \|\bar d_G(1)\|$ (see (2.23)). In addition,

$$\begin{aligned}
&\nabla_z\Psi_h(w)^T[\Pi_Z(z - \lambda\gamma\nabla_z\Psi_h(w)) - z]\\
&= -\frac{1}{\lambda\gamma}[z - \lambda\gamma\nabla_z\Psi_h(w) - z]^T[\Pi_Z(z - \lambda\gamma\nabla_z\Psi_h(w)) - z]\\
&= \frac{1}{\lambda\gamma}[\Pi_Z(z - \lambda\gamma\nabla_z\Psi_h(w)) - (z - \lambda\gamma\nabla_z\Psi_h(w))]^T[\Pi_Z(z - \lambda\gamma\nabla_z\Psi_h(w)) - z]\\
&\quad -\frac{1}{\lambda\gamma}\|\Pi_Z(z - \lambda\gamma\nabla_z\Psi_h(w)) - z\|^2\\
&\le -\frac{1}{\lambda\gamma}\|\Pi_Z(z - \lambda\gamma\nabla_z\Psi_h(w)) - z\|^2\\
&\le -\frac{\lambda}{\gamma}\|\Pi_Z(z - \gamma\nabla_z\Psi_h(w)) - z\|^2,
\end{aligned}\tag{3.3}$$

where the first and second inequalities come from the property of projector (see [1]). It follows from (3.2) and (3.3) that

$$\begin{aligned}
&\nabla\Psi_h(w)^T\tilde d_G(\lambda)\\
&= (\tilde t + \nabla_t H^T(w)H_h(w))[-\lambda\gamma(\tilde t + \nabla_t H(w)H_h(w)) + \lambda\beta\bar t]\\
&\quad + \nabla_z\Psi(w)^T[\Pi_Z(z - \lambda\gamma\nabla_z\Psi_h(w)) - z]\\
&\le -\frac{\lambda}{\gamma}\left[\|\gamma\nabla_t\Psi_h(w)\|^2 + \|\Pi_z(z - \gamma\nabla_z\Psi_h(w)) - z\|^2\right] + \alpha\bar t\frac{\lambda}{\gamma}\|\bar d_G(1)\|^2\\
&= -\frac{\lambda}{\gamma}(1 - \alpha\bar t)\|\bar d_G(1)\|^2 < 0.
\end{aligned}$$

The proof is complete. □

Remark. If (2.28) is not satisfied and $\bar d_G^k(1) \ne 0$, then from Step 2.1 of Algorithm 2.1 we know that only the projected negative gradient is chosen as the search direction. Hence, Lemma 3.2 shows that this is a descent direction, which implies that after a finite number of iterations, (2.28) will be satisfied.

In order to establish the global convergence of Algorithm 2.1, we need the following lemma, which shows that Algorithm 2.1 can keep $t^k > 0$ at each iteration.

LEMMA 3.3. *Let* $\{w^k\}$ *be a sequence generated by Algorithm* 2.1. *Then for each* k, $k = 0, 1, \dots$, $w^k = (t^k, z^k)$ *satisfies*

$$t^k \ge \beta_k\bar t. \tag{3.4}$$

Furthermore, if w^k is not a stationary point of (2.21)*, then*

$$t^k > 0.$$

Proof. We prove this lemma by induction. From the choices of t^0 and β_0 in Algorithm 2.1, it is obvious that (3.4) holds for $k = 0$. Suppose that for any integer l, $w^l = (t^l, z^l)$ satisfies (3.4). Now we prove that $w^{l+1} = (t^{l+1}, z^{l+1})$ satisfies (3.4) as well.

If the condition (2.28) is not satisfied for $k = l$, we have

$$\bar{d}^l(\lambda_l) = \tilde{d}^l_G(\lambda_l) = \Pi_W(w^l + \lambda_l d^l_G) - w^l, \quad d^l_G = -\xi_l \nabla \Psi_h(w^l) + \beta_l \bar{w},$$

where λ_l is the accepted steplength at the lth iteration. It follows from Algorithm 2.1 that

$$\begin{aligned}(\bar{d}^l(\lambda_l))_t &= \lambda_l[-\xi_l(t^l + \nabla_t H(w) H_h(w)) + \beta(w^l)\bar{t}] \\ &\geq -\lambda_l t^l + \lambda_l \beta(w^l)\bar{t} \qquad (\text{ see } (2.25) \text{ }),\end{aligned}$$

where $(\bar{d}^l(\lambda_l))_t$ is the first element of $\bar{d}^l(\lambda_l)$. Then we have

$$\begin{aligned}t^{l+1} - \beta(w^{l+1})\bar{t} &= t^l + (\bar{d}^l(\lambda_l))_t - \beta(w^{l+1})\bar{t} \\ &\geq (1-\lambda_l)t^l + \lambda_l\beta(w^l)\bar{t} - \beta(w^{l+1})\bar{t} \\ &\geq (1-\lambda_l)t^l + \lambda_l\beta(w^l)\bar{t} - \beta(w^l)\bar{t} \\ &= (1-\lambda_l)(t^l - \beta(w^l)\bar{t}) \geq 0,\end{aligned}$$

where the second and third inequalities are due to the monotonicity property of $\beta(w^l)$ in Lemma 3.1 and $t^l \geq \beta(w^l)\bar{t}$.

If the condition (2.28) is satisfied for $k = l$, then we have

$$(\bar{d}^l(\lambda_l))_t = (\tau^*(\lambda_l)\tilde{d}^l_G(\lambda_l) + (1 - \tau^*(\lambda_l))\tilde{d}^l_{tN}(\lambda_l))_t.$$

By a similar way, we can obtain that $t^{l+1} - \beta(w^{l+1})\bar{t} \geq 0$.

Therefore (3.4) holds for any nonnegative integer k. Furthermore, from (3.4) and the fact that w^k is not a stationary point of (2.21), $t^k > 0$ holds. We complete the proof. □

LEMMA 3.4. *Let $\{w^k\}$ be a sequence generated by Algorithm* 2.1*. Then any accumulation point of $\{w^k\}$ is a stationary point of* (2.21)*.*

Proof. Lemma 3.3 shows that if Algorithm 2.1 does not stop at a stationary point of (2.21), then $t^k > 0$ for any k. This implies that Ψ and Ψ_h are continuously differentiable at w_k. The remark after the proof of Lemma 3.2 implies that for k sufficiently large, the condition (2.28) is always satisfied and $\Psi_h(w) = \Psi(w)$ (see Remark (2) after Algorithm 2.1). Hence, by using a similar way to the proof of Theorem 4.1 in [18], we can prove that this theorem holds. □

In order to analyze the local convergence of Algorithm 2.1, we make the following standard assumption.

(A1) Let $w^* = (t^*, z^*) = (0, z^*)$ be an accumulation point of the sequence $\{w^k\}$ generated by Algorithm 2.1. Suppose $\lim_{k\in K} w_k = w^*$ for some subset $K \subset \{1, 2, \dots\}$, w^* is a solution of the system of equations (2.10), and Φ is BD-regular at w^* where the definition of BD-regularity refers to [13].

BD-regularity can be satisfied without special difficulty. Before giving a sufficient condition for BD-regularity to hold, we need the following assumptions:

(A2) The vectors $\nabla_x g(x, v^j), j = 1, \ldots, p$, are linearly independent.

(A3) The matrix $\nabla_x^2 L(x, u, V)$ is positive definite, and for every $j = 1, 2, \ldots, p$, the matrix $(\nabla_v^2 g(x, v^j))_M$ is negative definite whenever $J_M(x, v^j) \neq \emptyset$, where

$$J_M(x, v) = \{i \mid a_i < v_i + (\nabla_v g(x, v))_i < b_i\},$$

and $(\nabla_v^2 g(x, v^j))_M$ is a principal square submatrix of $\nabla_v^2 g(x, v)$, which is determined by the columns and rows with the index $i \in J_M(x, v)$.

(A4) For every $j = 1, 2, \ldots, p$, $\{i \mid v_i + (\nabla_v g(x, v))_i = a_i$ or $v_i + (\nabla_v g(x, v))_i = b_i\}$ is an empty set.

In addition, for any $(x, v) \in \Re^n \times \Re^m$, we denote

$$J_L(x, v) = \{i \mid v_i + (\nabla_v g(x, v))_i < a_i\}, \quad J_R(x, v) = \{i \mid b_i < v_i + (\nabla_v g(x, v))_i\}$$

and state a simple lemma without proof in the following.

Lemma 3.5. *Let*

$$T = \begin{bmatrix} A & B & DC \\ B^T & 0 & 0 \\ D^T & 0 & F \end{bmatrix},$$

where $A \in \Re^{p \times p}$, $B \in \Re^{p \times q}$, $C \in \Re^{r \times r}$, $D \in \Re^{p \times r}$, *and* $F \in \Re^{r \times r}$. *Suppose that* A *and* $C^T F$ *are positive definite and negative semidefinite, respectively. If the column rank of* B *and* F *are* q *and* r, *respectively, then* T *is nonsingular.*

Proof. Let $Td = 0$, where $d = (d_1, d_2, d_3)$ is a suitable partitioned vector. Then

$$Ad_1 + Bd_2 + DCd_3 = 0, \tag{3.5}$$

$$B^T d_1 = 0, \tag{3.6}$$

$$D^T d_1 + Fd_3 = 0. \tag{3.7}$$

Multiplication (3.5) with d_1^T yields

$$d_1^T Ad_1 + d_1^T Bd_2 + d_1^T DCd_3 = 0,$$

which, together with (3.6) and (3.7), implies

$$d_1^T Ad_1 + d_3^T(-C^T F)d_3 = 0.$$

From the property of A and $C^T F$, we have that $d_1 = 0$. Then it follows from (3.7) and the property of F that $d_3 = 0$. Because of (3.5) and the property of B, $d_2 = 0$ holds. The proof is complete. □

Theorem 3.6. *Suppose that* $w^* = (t^*, z^*) = (t^*, x^*, u^*, V^*, y^*)$ *is a solution of* (2.10) *and satisfies* (A2)–(A4). *Then* Φ *is BD-regular at* w^*.

Proof. Without loss of generality, by (A4), we assume that

$$\begin{aligned} J_L(x^*, v^{j*}) &= \{1, 2, \ldots, k_1^j\}, \\ J_M(x^*, v^{j*}) &= \{k_1^j + 1, \ldots, k_2^j\}, \\ J_R(x^*, v^{j*}) &= \{k_2^j + 1, \ldots, m\}, \end{aligned}$$

where $1 \leq k_1^j \leq k_2^j \leq m$. Because $w^* = (t^*, z^*)$ is a solution of (2.10), $t^* = 0$. Moreover, we have, by $\phi(0, x^*, v^{j*}) = 0$, that

$$v^{j*} - \text{mid}(a, b, v^{j*} + \nabla_v g(x^*, v^{j*})) = 0, \quad j = 1, \ldots, p. \tag{3.8}$$

By (3.8) and the definition of the mid function, we have that for $j = 1, \dots, p$ and $i \in J_M(x^*, v^{j*})$,

$$(\nabla_{v^j} g(x^*, v^{j*}))_i = 0. \tag{3.9}$$

By direct computation, we obtain that for any $Q \in \partial_B \Phi(w^*)$,

$$\begin{aligned} &Q \\ &= \left[\begin{array}{ccccccc} 1 & 0_{1\times n} & 0_{1\times p} & 0_{1\times m} & \cdots & 0_{1\times m} & 0 \\ 0_{n\times 1} & \nabla_x^2 L(x^*, u^*, V^*) & \nabla_x \mathbf{g}^T(x^*, V^*) & u_1^* D_1 & \cdots & u_p^* D_p & 0_{n\times 1} \\ 0 & \nabla_x^T g(x^*, v^{1*}) & 0_{1\times p} & \nabla_{v^1}^T g(x^*, v^{1*}) & \cdots & 0_{1\times m} & 0 \\ 0 & \nabla_x^T g(x^*, v^{2*}) & 0_{1\times p} & 0_{1\times m} & \cdots & 0_{1\times m} & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \nabla_x^T g(x^*, v^{p*}) & 0_{1\times p} & 0_{1\times m} & \cdots & \nabla_{v^p}^T g(x^*, v^{p*}) & 0 \\ Q_1 & C_1 D_1^T & 0_{m\times p} & E_1 + C_1 F_1 & \cdots & 0_{m\times m} & 0_{p\times 1} \\ Q_2 & C_2 D_2^T & 0_{m\times p} & 0_{m\times m} & \cdots & 0_{m\times m} & 0_{p\times 1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ Q_p & C_p D_p^T & 0_{m\times p} & 0_{m\times m} & \cdots & E_p + C_p F_p & 0_{p\times 1} \\ U_1 & U_2 & 0_{1\times p} & 0_{1\times m} & \cdots & 0_{1\times m} & 1 \end{array}\right], \end{aligned} \tag{3.10}$$

where $U_1 \in \partial_t G(0, x^*)$, $U_2 \in \partial_x G(0, x^*)$, and for $j = 1, \dots, p$,

$$Q_j \in \partial_t \phi(0, x^*, v^{j*}),\ D_j = \nabla_{v^j}^T(\nabla_x g(x^*, v^{j*})),\ F_j = \nabla_{v^j}^T(\nabla_{v^j} g(x^*, v^{j*})),$$

$$C_j = \operatorname{diag}(0_{j_1}, -I_{j_2}, 0_{j_3}), \quad E_j = \operatorname{diag}(I_{j_1}, 0_{j_2}, I_{j_3}), \tag{3.11}$$

where $0_{j_1}, 0_{j_2}$, and 0_{j_3} are zero square matrices with k_1^j, $(k_2^j - k_1^j)$, and $(m - k_2^j)$ order, respectively, and I_{j_1}, I_{j_2}, and I_{j_3} are identity matrices with k_1^j, $(k_2^j - k_1^j)$, and $(m - k_2^j)$ order, respectively. By (3.10), it is easy to see that the matrix Q is also nonsingular as the matrix

$$\tilde{Q} = \left[\begin{array}{ccccc} \nabla_x^2 L(x^*, u^*, V^*) & \nabla_x \mathbf{g}^T(x^*, V^*) & u_1^* D_1 & \cdots & u_p^* D_p \\ \nabla_x^T g(x^*, v^{1*}) & 0_{1\times p} & \nabla_{v^1}^T g(x^*, v^{1*}) & \cdots & 0_{1\times m} \\ \nabla_x^T g(x^*, v^{2*}) & 0_{1\times p} & 0_{1\times m} & \cdots & 0_{1\times m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \nabla_x^T g(x^*, v^{p*}) & 0_{1\times p} & 0_{1\times m} & \cdots & \nabla_{v^p}^T g(x^*, v^{p*}) \\ C_1 D_1^T & 0_{m\times p} & E_1 + C_1 F_1 & \cdots & 0_{m\times m} \\ C_2 D_2^T & 0_{m\times p} & 0_{m\times m} & \cdots & 0_{m\times m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_p D_p^T & 0_{m\times p} & 0_{m\times m} & \cdots & E_p + C_p F_p \end{array}\right].$$

We denote by $(D_j)_{ML}$ a submatrix of D_j constituted by the columns with the index $i \in J_M(x, v^j)$ and by $(F_j)_M$ a principal square submatrix of F_j, which is determined by the columns and rows with the index $i \in J_M(x, v^j)$. Then from special forms of C_j and E_j we have

$$C_j D_j^T = \left[\begin{array}{c} 0 \\ -(D_j)_{ML}^T \\ 0 \end{array}\right], \quad E_j + C_j F_j = \left[\begin{array}{ccc} I_{j_1} & 0 & 0 \\ * & -(F_j)_M & * \\ 0 & 0 & I_{j_2} \end{array}\right],$$

where two $*$ are some proper partitioned matrices. Hence the nonzero elements of $\nabla_{v^j}^T g(x^*, v^{j*})$ and the matrix $*$ are deleted by the some proper row transformations. Hence it is not difficult to know that the matrix $\tilde{Q}$ is also nonsingular as the matrix

$$Q^* = \begin{bmatrix} \nabla_x^2 L(x^*, u^*, V^*) & \nabla_x \mathbf{g}^T(x^*, V^*) & DU \\ \nabla_x^T \mathbf{g}(x^*, V^*) & 0 & 0 \\ D^T & 0 & F \end{bmatrix}, \tag{3.12}$$

where $D = ((D_1)_{ML}, \ldots, (D_p)_{ML})$, $F = \operatorname{diag}((F_1)_M, \ldots, (F_p)_M)$, $U = \operatorname{diag}(u_1^* I_1, \ldots, u_p^* I_p)$, and I_j, $j = 1, \ldots, p$, are some proper identity matrices. It is clear that $U^T F$ is negative definite, and from (A2) and (A3) it follows that all other conditions in Lemma 3.5 are satisfied. Hence from Lemma 3.5 we know that Q^* is nonsingular and we complete the proof. □

The following lemma is the same as Lemma 4.1 in [14]; its proof is omitted.

LEMMA 3.7. *There exist positive constants κ and ϵ such that for every w_k satisfying $\|w_k - w^*\| \leq \epsilon$,*

(i) *$\nabla^T \Phi(w_k)$ is nonsingular and satisfies*

$$\|\nabla^T \Phi(w_k)\| \leq \kappa,$$

(ii)

$$\|\Phi(w_k)\| = \sqrt{2}\Psi(w_k)^{\frac{1}{2}} = O(\|w_k - w^*\|).$$

LEMMA 3.8. *Let $\{w^k\}$ be a sequence generated by Algorithm 2.1. Then for all $k \in K$ sufficiently large, we have*

$$\beta(w^k) = O(\Psi(w^k)) = O(\|w^k - w^*\|^2); \tag{3.13}$$

and

$$w^k + \lambda d_{tN}^k = (1 - \lambda) w^k + \lambda w^* + \lambda o(\Psi(w^k)^{\frac{1}{2}}) \tag{3.14}$$

for any $\lambda \in (0, 1]$.

Proof. From the definition of $\beta(w_k)$ (see (2.24)), the choice of γ_k (see (2.26)), the projection property, and Lemma 3.7, it follows that for w^k sufficiently close to w^*, $\Psi_h(w_k) = \Psi(w_k)$,

$$\beta(w^k) \leq \alpha \|\bar{d}_G^k(1)\|^2 \leq \alpha \gamma_k^2 \|\nabla \Psi(w^k)\|^2 \leq \alpha \eta \Psi(w^k) = \frac{\alpha \eta}{2} \|\Phi(w^k)\|^2 = O(\|w^k - w^*\|^2).$$

This shows that (i) holds. Let

$$\theta_k = \Phi(w_k) - \beta_k \bar{w} + \nabla^T \Phi(w_k) d_{tN}^k.$$

Then from (2.29), we have that for w^k sufficiently close to w^*,

$$\|\theta_k\| = o(\Psi_h(w_k)) = o(\Psi(w_k)), \tag{3.15}$$

which implies that

$$\begin{aligned} w^k + \lambda d_{tN}^k &= w^k + \lambda \nabla^T \Phi(w^k)^{-1} [-\Phi(w^k) + \beta(w^k) \bar{w} + \theta_k] \\ &= w^k - \lambda \nabla^T \Phi(w^k)^{-1} [\Phi(w^k) - \Phi(w^*) - \nabla^T \Phi(w^k)(w^k - w^*)] \\ &\quad - \lambda (w^k - w^*) + \lambda \nabla^T \Phi(w^k)^{-1} (\beta(w^k) \bar{w} + \theta_k) \\ &= (1 - \lambda) w^k + \lambda w^* + \lambda o(\|w^k - w^*\|) + \lambda o(\Psi(w^k)) \\ &= (1 - \lambda) w^k + \lambda w^* + \lambda o(\Psi(w^k)^{\frac{1}{2}}), \end{aligned}$$

where the third equality is due to the semismoothness of Φ, (i), and (3.15). The proof is complete. □

Now we obtain the following convergence theorem.

THEOREM 3.9. *Suppose that $\{w^k\}$ is a sequence generated by Algorithm* 2.1 *and w^* is a point satisfying* (A1). *Then the whole sequence $\{w^k\}$ superlinearly converges to w^*.*

Proof. At first we know that for w^k sufficiently close to w^*, $\Psi_h(w_k) = \Psi(w_k)$.

In a similar way to the proof of Theorem 3.2 in [18] and Lemma 3.8, we have that for sufficiently large $k \in K$,

$$\|w^k + \bar{d}^k(1) - w^*\| = o(\Psi(w^k)^{\frac{1}{2}}) = o(\|\Phi(w^k)\|) = o(\|w^k - w^*\|), \tag{3.16}$$

and

$$\begin{aligned} \Psi(w^k + \bar{d}^k(1)) &= \frac{1}{2}\|\Phi(w^k + \bar{d}^k(1))\|^2 \\ &= \frac{1}{2}\|\Phi(w^k + \bar{d}^k(1)) - \Phi(w^*)\|^2 \\ &= O(\|w^k + \bar{d}^k(1) - w^*\|^2) \\ &= o(\Psi(w^k)), \end{aligned} \tag{3.17}$$

where the last equality is due to (3.16). Thus,

$$\begin{aligned} -\nabla\Psi(w^k)^T\tilde{d}_G^k(1) &\le \|\nabla\Psi(w^k)\|\|\tilde{d}_G^k(1)\| \\ &= \|\nabla\Psi(w^k)\|\|\Pi_W(w^k - \gamma_k\nabla\Psi(w^k) + \beta(w^k)\bar{w}) - w^k\| \\ &\le \|\nabla\Psi(w^k)\|[\|\gamma_k\nabla\Psi(w^k)\| + O(\Psi(w^k))] \\ &\le \eta\Psi(w^k) + o(\Psi(w^k)), \end{aligned} \tag{3.18}$$

where the second inequality is due to the property of $\beta(w^k)$ and the projection property, and the last inequality comes from the choice of γ_k. It follows from (3.17) and (3.18) that

$$\begin{aligned} \Psi(w^k) + \sigma\nabla\Psi(w^k)^T\tilde{d}_G^k(1) &\ge (1 - \sigma\eta)\Psi(w^k) + o(\Psi(w^k)) \\ &\ge o(\Psi(w^k)) = \Psi(w^k + \bar{d}^k(1)), \end{aligned} \tag{3.19}$$

which implies that

$$w^{k+1} = w^k + \bar{d}^k(1),$$

for k sufficiently large. Moreover, from (3.16) we conclude that w^k converges to w^* superlinearly. We complete the proof. □

4. Implementation and numerical tests. In this section, we discuss some detailed implementation of Algorithm 2.1 and give some numerical results for medium-sized and large-scale SIP problems.

4.1. Implementation of Algorithm 2.1. In order to decrease the number of inner iterations, we use another line search technique if only the projected gradient direction is the search direction. In this case, the initial value of λ is set to

$$\min\left\{1, \frac{1}{\|d_G^k\|}, \frac{0.2\Psi_h(w_k)}{-\nabla\Psi_h(w_k)^T d_G^k}, \frac{t_k}{|t_k + \nabla_t H(w_k)H_h(w_k)|}\right\},$$

and λ is updated by quadratic interpolation technique.

In Algorithm 2.1, we choose the values of parameters (see Step 0) as

$$\eta = 0.9,\ \rho = 0.5,\ \sigma = 0.0005,\ \alpha = 0.5,\ \bar{t} = 0.9,\ p_1 = 10^{-10},\ p_2 = 2.1,$$

and

$$h_j = \max\{2.5, 10^{-3} * |\Psi(w^0)|,\ j = 1, 2, \ldots, \tilde{n}\},$$

where the choice of h_j, $j = 1, 2, \ldots, \tilde{n}$, is similar to that in [9]. The starting points u^0 and y^0 for all problems are set to $t^0 = \bar{t}$, $u^0 = 0.05e$, $y^0 = 0.5$, where e is the vector of ones. For GMRES($\tilde{m}$), we choose $\tilde{m} = 10$ when $n < 100$ and $\tilde{m} = 20$ when $n \geq 100$.

In some test problems, we have tried using a simple left preconditioning matrix

$$M = \mathrm{diag}(1, L_{ssor}, I_{(m+1)p+1}),$$

where L_{ssor} is an SSOR preconditioning matrix defined by

$$L_{ssor} = (D - \omega E)D^{-1}(D - \omega F),$$

D is the diagonal part of $\nabla^2_{xx} L(x, u, V)$, and $-F$ and $-E$ are the strict upper and lower parts of $\nabla^2_{xx} L(x, u, V)$. Numerical results show that GMRES without preconditioning is better than that with preconditioning for $\omega = 1$ and $\omega = 0.5$. Hence we give the numerical results without preconditioning in the following.

4.2. Numerical results. Now we discuss the implementation of Algorithm 2.1, which has been implemented in FORTRAN 77. All calculation within the driving programs, test problems, and optimization code are carried out in double precision. The problem is solved on a personal computer (Pentium III 1133 MHz, 256 MB memory).

Although a lot of large SIP-type problems arise from optimal control and approximation theory, it is difficult to find large-scale SIP problems in the literature suitable for using as test problems. In order to evaluate Algorithm 2.1 for large-scale SIP problems, we enlarge three test problems, where two problems are from [14] and [8] and another is generated from an optimal control problem. We list the three SIP problems in the following.

Problem I.

$$f(x) = \frac{1}{2} x^T x, \quad g(x, v) = 3 + 4.5 \sin\left(\frac{4.7\pi(v - 1.23)}{8}\right) - \sum_{i=1}^{n} x_i v^{i-1},$$
$$V = [1, b], p = 1 \text{ if } n \leq 60, b = 100;\ \text{otherwise } b = 1.$$

Problem II.

$$f(x) = \int_0^1 \left(\sum_{i=1}^{n} x_i t^{i-1} - \tan t\right)^2 dt,\ g(x, v) = \tan v - \sum_{i=1}^{n} x_i v^{i-1},\ V = [0, 1], p = 1.$$

Problem III.

$$\begin{array}{rl} \min & p(g) h^T h \\ \text{subject to} & g^T A(v_1, v_2) h \leq r(v_1, v_2), \end{array}$$

Table 1
Test results of Problem I.

n	ITK	iITK	$\|\bar{d}_G^k(1)\|$	$\Psi(w_k)$	$f(x_k)$
10	8	65	5.52E-12	1.60E-19	0.07412
20	8	50	4.96E-12	4.29E-19	0.08319
40	67	393	9.86E-7	2.06E-11	3.1788
60	98	489	8.84E-7	2.91E-11	4.8860
100	60	286	9.66E-7	5.10E-8	2.3862
400	67	580	2.70E-7	3.58E-13	4.580
1000	78	630	8.29E-6	2.97E-10	8.519
2000	52	603	6.96E-6	6.69E-7	18.07

where $v_1 \in [-\pi, \pi]$, $v_2 \in [0, 2\pi]$, $p(g) = g^T B g$, $h \in \Re^{n_1}$, $g \in \Re^{n_2}$, $B \in \Re^{n_2 \times n_2}$; $A(v_1, v_2) \in \Re^{n_2 \times n_1}$, $n_2 = n_1$, and

$$B = \begin{bmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{bmatrix},$$

$$A(v_1, v_2)$$
$$= \begin{bmatrix} 1 & \sin(bv_2) & \cos(cv_1) & & & & & & \\ \sin(av_1) & 1 & \sin(bv_2) & \cos(cv_1) & & & & & \\ \cos(dv_2) & \sin(av_1) & 1 & \sin(bv_2) & \cos(cv_1) & & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & & \cos(dv_2) & \sin(av_1) & 1 & \sin(bv_2) & \cos(cv_1) \\ & & & & & \cos(dv_2) & \sin(av_1) & 1 & \sin(bv_2) \\ & & & & & & \cos(dv_2) & \sin(av_1) & 1 \end{bmatrix}.$$

We use Algorithm 2.1 to solve these problems. The dimensions (n) of these problems are chosen by 10, 20, 40, 60, 80, 100, 200, 400, 1000, and 2000. The termination condition is that the l_2 norm of the projected gradient, $\|\bar{d}_G^k(1)\|$, is reduced below 10^{-6} when $n < 100$ (10^{-5} when $n \geq 100$). The results of the test are given in Tables 1, 2, and 3. The number of outer iterations (ITK), the total number of inner iterations for solving subproblems (iITK), the norm of projected gradient ($\|\bar{d}_G^k(1)\|$), the merit function value $\Psi(w_k)$, and the objective function value $f(x_k)$ are shown in these tables.

Table 1 shows that Algorithm 2.1 performs very well for solving Problem I with the different dimensions. There is some difference among different dimensions. When $n \geq 40$, there is a slight increase in the iteration number.

Problem II is dense; i.e., its Hessian of Lagrangian function $\nabla_x^2 L(x, u, V)$ is not sparse. Although the Hessian can be stored according to its special structure, the computation in each iteration cannot be decreased. Here Algorithm 2.1 is used for solving Problem II, whose dimensions range from 10 to 200. Table 2 shows that Algorithm 2.1 performs well for solving some medium dense SIP problems.

TABLE 2
Test results of Problem II.

n	ITK	iITK	$\|\bar{d}_G^k(1)\|$	$\Psi(w_k)$	$f(x_k)$
10	31	140	1.14E-7	4.86E-12	0.3147
20	46	235	9.68E-7	7.98E-10	0.6717
40	50	306	1.48E-7	3.48E-12	0. 5803
80	65	476	4.48E-7	3.93E-11	1.424
100	58	528	2.87E-7	1.86E-11	1.069
200	71	768	2.86E-6	1.46E-9	1.323

TABLE 3
Test results of Problem III.

n	ITK	iITK	$\|\bar{d}_G^k(1)\|$	$\Psi(w_k)$	$f(x_k)$
20	26	694	9.04E-7	7.80E-12	18.23
60	28	973	5.72E-7	3.03E-12	21.82
100	27	963	9.63E-6	8.98E-12	20.36
200	24	605	6.97E-6	4.56E-10	16.84
600	20	509	9.21E-6	7.75E-10	13.72
1000	25	494	9.32E-6	7.59E-10	13.82
2000	22	488	8.39E-6	8.06E-10	13.81

Problem III is a somewhat complicated SIP problem which often arises from the optimal control field. In this problem, $\Omega \subset \Re^2$, while in Problems I and II, $\Omega \subset \Re$. Its Hessian of the Lagrangian function is sparse; however, the computation of elements is not simple due to some trigonometric functions. Numerical results of this problem are given in Table 3, which shows that Algorithm 2.1 can solve some large-scale sparse SIP problems. It is interesting that the outer iteration number does not increase and inner iteration numbers decrease as the dimensions increase.

5. Comments.

Although the development of the code for Algorithm 2.1 is still at its primary stage, the numerical results have indicated that Algorithm 2.1 is capable of processing large-scale SIP problems. However, there are some issues which may be addressed in further research.

Because "large scale" here refers only to the decision variables, it is hoped that an improved version of Algorithm 2.1 may also be capable of handling high-dimensional index sets. In addition, our method works on the KKT system of SIP; i.e., it does not minimize the original objective function f. Sometimes this may limit the applicability of this method to a special class of SIP problems.

By Algorithm 2.1 we can obtain stationary points of (2.21). It is possible that some of them may not be stationary points of (1.1). If Ω in (1.1) is a nonpolyhedral index set, then our method cannot be used directly.

We hope that with further research more efficient methods can be obtained for solving general SIP problem with many decision variables and high-dimensional index sets.

Acknowledgments. We are grateful to Professor Fukushima and two anonymous referees for their detailed comments. The approach for solving nonsymmetric linear system suggested by a referee improved our method.

REFERENCES

[1] P. H. Calamai and J. J. Moré, *Projected gradient methods for linear constrained problems*, Math. Programming, 39 (1987), pp. 93–116.

[2] J. E. Dennis, *Nonlinear least squares and equations*, in The State of the Art in Numerical Analysis, D. A. H. Jacobs, ed., Academic Press, New York, 1975, pp. 269–312.

[3] V. Fraysse, L. Giraud, S. Gratton, and J. Langou, *A Set of GMRES Routines for Real and Complex Arithmetics on High Performance Computers*, CERFACS Technical report TR/PA/03/3, Toulouse Cedex, France, 2003.

[4] M. A. Goberna and M. A. López, *Semi-infinite Programming: Recent Advances*, Kluwer Academic Publishers, Boston, 2001.

[5] P. R. Gribik, *Selected applications of semi-infinite programming*, in Constructive Approaches to Mathematical Models, Academic Press, New York, 1979, pp. 171–187.

[6] R. Hettich and K. O. Kortanek, *Semi-infinite programming: Theory, methods, and applications*, SIAM Rev., 35 (1993), pp. 380–429.

[7] P. J. Huber, *Robust regression: Asympotics, conjectures, and Monte Carlo*, Ann. Statist., 1 (1973), pp. 799–821.

[8] S. Ito, Y. Liu, and K. L. Teo, *A dual parametrization method for convex semi-infinite programming*, Ann. Oper. Res., 98 (2000), pp. 189–214.

[9] Q. Ni, *Global convergence and implementation of NGTN method for solving large scale nonlinear sparse nonlinear programming problems*, J. Comput. Math., 19 (2001), pp. 337–346.

[10] J.-S. Pang and L. Qi, *Nonsmooth equations: Motivation and algorithms*, SIAM J. Optim., 3 (1993), pp. 443–465.

[11] E. Polak, *On the mathematical foundations of nondifferentiable optimization in engineering design*, SIAM Rev., 29 (1987), pp. 21–89.

[12] E. Polak, *Optimization: Algorithms and Consistent Approximation*, Springer-Verlag, New York, 1997.

[13] L. Qi, *Convergence analysis of some algorithms for solving nonsmooth equations*, Math. Oper. Res., 18 (1993), pp. 227–244.

[14] L. Qi, C. Ling, X. J. Tong, and G. L. Zhou, *A Smoothing Projected Newton-Type Algorithm for Semi-infinite Programming*, Technical report, Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hum, Kowloon, Hong Kong, 2004.

[15] R. Reemtsen and S. Görner, *Numerical methods for semi-infinite programming: A survey*, in Semi-infinite Programming, R. Reemtsen and J. Rükmann, eds., Kluwer Academic Publishers, Boston, 1998, pp. 195–275.

[16] E. W. Sachs, *Semi-infinite programming in control*, in Semi-infinite Programming, R. Reemtsen and J. Rükmann, eds., Kluwer Academic Publishers, Boston, 1998, pp. 389–411.

[17] G. Still, *Discretization in semi-infinite programming: The rate of convergence*, Math. Program., 91 (2001), pp. 53–69.

[18] D. Sun, R. S. Womersley, and H. Qi, *A feasible semismooth asymptotically Newton method for mixed complementarity problems*, Math. Program., 94 (2002), pp. 167–187.

[19] K. L. Teo, C. J. Goh, and K. H. Wong, *A Unified Computational Approach to Optimal Control Problems*, Longman Scientific and Technical, New York, 1991.

[20] K. L. Teo, V. Rehbock, and L. S. Jennings, *A new computational algorithm for functional inequality constrained optimization problems*, Automatica, 29 (1993), pp. 789–792.

SIAM J. OPTIM.
Vol. 16, No. 4, pp. 1155–1176

COMBINATORIAL AND CONTINUOUS MODELS FOR THE OPTIMIZATION OF TRAFFIC FLOWS ON NETWORKS*

A. FÜGENSCHUH[†], M. HERTY[‡], A. KLAR[‡], AND A. MARTIN[†]

Abstract. A hierachy of simplified models for traffic flow on networks is derived from continuous traffic flow models based on partial differential equations. The hierachy contains nonlinear and linear combinatorial models with and without dynamics. Optimization problems are treated for all models and numerical results and algorithms are compared.

Key words. traffic networks, partial differential equations, combinatorial methods

AMS subject classifications. 90B20, 35L50

DOI. 10.1137/040605503

1. Introduction.

Modeling and simulation of traffic flow on highways has been investigated intensively in the last few years. On the one hand, models describing detailed traffic dynamics on single roads have been constantly developed and improved; see [33, 32, 28, 15, 31, 1, 23, 14, 11, 8, 12]. In these papers models based on ordinary differential equations, partial differential equations, and kinetic equations have been derived. To describe traffic flow on networks, detailed dynamic models based on partial differential equations have been used in [18, 7], where theoretical results on existence of solutions for the network problem were obtained. However, the number of roads which can be treated with such an approach is restricted, in particular, if optimization problems have to be solved. On the other hand, large traffic networks with strongly simplified dynamics or even static description of the flow have been widely investigated [6, 10, 21, 26, 29, 24]. In particular, optimal control problems for traffic flow on networks arising from traffic management (see, for example, [27, 3]) are a major focus of research in this field. The purpose of the present investigation is to derive and develop a hierachy of simplified dynamical models based on the correct dynamics described by partial differential equations. These models should include reasonable dynamics and, at the same time, they should be solvable for large scale networks. Special focus is on optimal control problems and optimization techniques. We start with macroscopic models based on partial differential equations. Two such models were introduced by Holden and Risebro [18] and Coclite and Piccoli [7], respectively. In particular, dealing with optimal control questions for such large-scale networks where the flow is described by partial differential equations is very expensive from a computational point of view; see [16, 17]. Therefore, we concentrate in the present paper on the derivation of simplified dynamic models derived from the models based on partial differential equations. The resulting models are network models which are based on nonlinear algebraic equations or combinatorial models based on

*Received by the editors March 23, 2004; accepted for publication (in revised form) August 16, 2005; published electronically March 3, 2006. This work was partially supported by Deutsche Forschungsgemeinschaft (DFG), KL 1105/5.

http://www.siam.org/journals/siopt/16-4/60550.html

[†]Technische Universitat Darmstadt, Fachbereich Mathematik, Schlossgartenstr. 7, Building 2D, 64289 Darmstadt, Germany (fuegenschuh@mathematik.tu-darmstadt.de, martin@mathematik.tu-darmstadt.de).

[‡]TU Kaiserslautern, Department of Mathematics, Postfach 3049, 67653 Kaiserslautern, Germany (herty@mathematik.uni-kl.de, klar@itwm.fhg.de).

linear equations. In the simplest case well-known static combinatorial problems like min cost flow models are obtained. For the different models we study optimal control problems and various optimization methods, i.e., combinatorial and continuous optimization techniques. Using strongly simplified models, large-scale networks can be optimized with combinatorial approaches in real time. However, including more complex dynamics reduces the advantage of the combinatorial algorithms compared to continuous optimization procedures.

2. Continuous traffic flow models.

The starting point is a macroscopic traffic flow model on networks. We give a brief review of the model. Further details and more general situations are treated in [7, 16]. We consider a network of roads as follows.

DEFINITION 2.1. *For some traffic road map we introduce a finite, connected directed graph $G = (V, A)$ where the arcs $j \in A$ correspond to the roads and the vertices $v \in V$ to the junctions. G is also called traffic flow network. With each arc $j \in A$ we associate an interval $[a_j, b_j]$ representing the location x at the corresponding road, with the interpretation $x = a_j$ if we are at the tail of arc j and $x = b_j$ if we are at the head of arc j. At a single junction v the set of indices of ingoing roads is denoted by δ_v^- and the set of outgoing roads by δ_v^+.*

On each arc j, the traffic dynamics are described by a model based on partial differential equations for the density $\rho_j(x,t)$, $x \in [a_j, b_j]$, $t \geq 0$. We use the well-known Lighthill–Whitham equations [30] to model the evolution of the density. Hence the following equations are assumed to hold on the network away from junctions:

$$\partial_t \rho_j(x,t) + \partial_x f_j(\rho_j(x,t)) = 0 \ \forall j \in A, x \in [a_j, b_j], t \geq 0, \tag{2.1}$$

$$\rho_j(x,0) = \bar{\rho}_j(x) \ \forall x \in [a_j, b_j], \tag{2.2}$$

where $f_j(\rho) = \rho u_j^e(\rho)$ and $u_j^e(\rho)$ is the fundamental diagram and $\bar{\rho}_j(x)$ are given initial values. A solution $(\rho_j)_{j\in A}$ to the network problem should satisfy flux conservation through junctions, i.e., for all $v \in V$ we have

$$\sum_{j\in\delta_v^-} f_j(\rho_j(b_j,t)) = \sum_{j\in\delta_v^+} f_j(\rho_j(a_j,t)) \ \forall t \in \,]0,\infty[\,. \tag{2.3}$$

To obtain a well-defined problem we have to impose further boundary conditions in the sense of [2], since (2.3) is not sufficient to obtain unique solutions $(\rho_j)_{j\in A}$. An overview of possible models for junctions and the corresponding boundary values for the equations (2.1) can be found in [16]. For the following derivations we restrict ourselves to the Coclite–Piccoli model of junctions [7]. For nonconstant initial data $\bar{\rho}_j(x)$ the obtained boundary conditions cannot be given explicitly. They are well defined in the case of constant initial data and obtained as the limit of an approximation; see [7, 9, 19]. However, in certain cases and under some restrictions one can derive explicit formulas for the boundary values. We briefly describe the Coclite–Piccoli coupling conditions for special geometries. We restrict ourselves for simplicity to networks with only two types of junctions with a total of three incident roads; see Figure 2.1. We assume that f_j is defined on $[0, \rho_{j,\max}]$ and assume that f_j is smooth, concave with single maximum. Define

$$M_j = \max f_j(\rho), \ \sigma_j = \text{argmax} f_j(\rho). \tag{2.4}$$

We consider the case of a single junction v and constant initial data $\bar{\rho}_j$. Coclite and Piccoli introduced additional conditions for a junction by considering Riemann

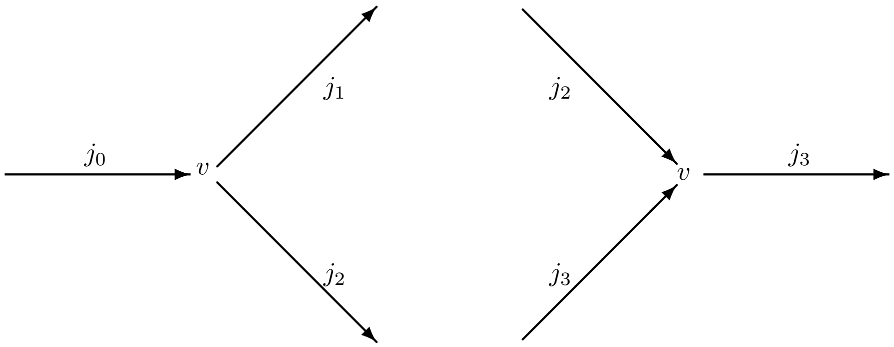


FIG. 2.1. *Considered types for a junction v. The used notation is $\delta_v^- = \{j_0\}$, $\delta_v^+ = \{j_1, j_2\}$ (left) and $\delta_v^- = \{j_1, j_2\}$, $\delta_v^+ = \{j_3\}$ (right), respectively.*

problems on the in- and outgoing roads. To be more precise, assume we have given values $\tilde{p}_j \in \mathbb{R}^+$ and the initial value $\bar{\rho}_j \in \mathbb{R}^+$. We consider the following problem:

$$\begin{aligned}&\partial_t \rho_j + \partial_x f_j(\rho_j) = 0,\\ &j \in \delta_v^+ : \rho_j(x,0) = \begin{bmatrix}\bar{\rho}_j & x > a_j\\ \tilde{p}_j & x \leq a_j\end{bmatrix}, \text{ resp.,}\\ &j \in \delta_v^- : \rho_j(x,0) = \begin{bmatrix}\tilde{p}_j & x \geq b_j\\ \bar{\rho}_j & x \leq b_j\end{bmatrix}.\end{aligned} \tag{2.5}$$

A solution $(\rho_j)_{j\in A}$ of (2.5) also satisfies (2.1). We have a degree of freedom in choosing the values $\tilde{p}_j$. Since the conservation of flux (2.3) holds, there are certain restriction on $\tilde{p}_j$. They can be given explicitly by (2.7) using the following definition of the function $\rho \mapsto \tau(\rho)$:

$$\text{for given } \rho \text{ define } \tau = \tau_j(\rho) \text{ to be } \tau \neq \rho, f_j(\tau) = f_j(\rho). \tag{2.6}$$

The restrictions are

$$\begin{aligned}&j \in \delta_v^- : \quad \tilde{p}_j \in \begin{bmatrix}\{\bar{\rho}_j\}\cup]\tau_j(\bar{\rho}_j), \rho_{j,\max}] & \text{if } \bar{\rho}_j < \sigma_j\\ [\sigma, \rho_{j,\max}] & \text{if } \bar{\rho}_j \geq \sigma_j\end{bmatrix},\\ &j \in \delta_v^+ : \quad \tilde{p}_j \in \begin{bmatrix}[0, \sigma_j) & \text{if } \bar{\rho}_j < \sigma_j\\ \{\bar{\rho}_j\} \cup [0, \tau_j(\bar{\rho}_j)[& \text{if } \bar{\rho}_j \geq \sigma_j\end{bmatrix}.\end{aligned} \tag{2.7}$$

Depending on to which interval $\tilde{p}_j$ belongs, the wave generated by the Riemann problem (2.5) is either a shock wave or a rarefaction wave. But still, there are many possible choices for $\tilde{p}_j$ depending on the values of $\bar{\rho}_j$. Therefore Coclite and Piccoli introduced more constraints.

Case 1. Consider a single junction v where road j_0 disperse in two roads j_1 and j_2. A value $\alpha_v \in \mathbb{R}$ with $0 < \alpha_v < 1$ specifying the percentage of drivers coming from road j_0 and driving to j_1 is introduced. (2.3) reads

$$\begin{aligned}&f_{j_1}(\rho_{j_1}(a_{j_1}+,\cdot)) = \alpha_v f_{j_0}(\rho_{j_0}(b_{j_0}-,\cdot)),\\ &f_{j_2}(\rho_{j_2}(a_{j_2}+,\cdot)) = (1-\alpha_v) f_{j_0}(\rho_{j_0}(b_{j_0}-,\cdot)).\end{aligned} \tag{2.8}$$

Unique values $\tilde{p}_j$, $j = j_0, j_1, j_2$, can be found by solving the maximization problem

$$\max f_{j_0}(\tilde{p}_j) \text{ such that (s.t.) } (2.7), (2.8), (2.5). \tag{2.9}$$

Expression (2.9) does not allow an explicit representation of the boundary conditions. If we neglect the possibility of shock waves, especially backward-going shock waves on the incoming street j_0, the situation is much simpler. Therefore we assume in the following

$$\bar{\rho}_j, \rho_j(x,t) \le \sigma_j \ \forall j \in A. \tag{2.10}$$

Since we omit shock waves on j_0 we obtain instead of the maximization problem (2.9) an explicit formula for calculating $\tilde{p}_j$:

$$\tilde{p}_{j_0} = \bar{\rho}_{j_0}, \ \ f_{j_1}(\tilde{p}_{j_1}) = \alpha_v f_{j_0}(\tilde{p}_{j_0}), \ \ f_{j_2}(\tilde{p}_{j_2}) = (1-\alpha_v) f_{j_0}(\tilde{p}_{j_0}). \tag{2.11}$$

Equation (2.11) is well defined due to (2.10) and yields unique values $\tilde{p}_{j_1}, \tilde{p}_{j_2}$.

Remark 2.1. Coclite and Piccoli proved the existence and uniqueness of admissible solutions satisfying (2.9) for a single junction with constant initial data. They generalized this result to prove existence for networks where for each junction v we have $|\delta_v^-| + |\delta_v^+| \le 4$ and the initial data $\bar{\rho}_j$ has bounded total variation. For more details see [7].

Case 2. Consider a single junction v where roads j_1 and j_2 merge to j_3. Flux conservation through the junction implies

$$f_{j_3}(\rho_{j_3}(a_{j_3}+,\cdot)) = f_{j_1}(\rho_{j_1}(b_{j_1}-,\cdot)) + f_{j_2}(\rho_{j_2}(b_{j_2}-,\cdot)). \tag{2.12}$$

In the same spirit as above we define unique values $\tilde{p}_j$, $j = j_i$, $i = 1,2,3$, by a procedure suggested in [16]: define maximal possible fluxes by

$$j \in \delta_v^- = \{j_1, j_2\}: \quad \gamma_j = \begin{bmatrix} f_j(\bar{\rho}_j) & \text{if } \bar{\rho}_j < \sigma_j \\ M_j & \text{if } \bar{\rho}_j \ge \sigma_j \end{bmatrix},$$

$$j \in \delta_v^+ = \{j_3\}: \quad \gamma_j = \begin{bmatrix} M_j & \text{if } \bar{\rho}_j < \sigma_j \\ f_j(\bar{\rho}_j) & \text{if } \bar{\rho}_j \ge \sigma_j \end{bmatrix}$$

and solve the maximization problem:

$$\text{If } \gamma_{j_1} + \gamma_{j_2} > \gamma_{j_3} \max \sum_{j \in \delta_v^-} f_j(\tilde{p}_j) \text{ s.t. } (2.12), (2.7), \text{ and } f_{j_1}(\tilde{p}_{j_1}) = f_{j_2}(\tilde{p}_{j_2}),$$
$$\text{If } \gamma_{j_1} + \gamma_{j_2} \le \gamma_{j_3} \max \sum_{j \in \delta_v^-} f_j(\tilde{p}_j) \text{ s.t. } (2.12), (2.7). \tag{2.13}$$

Again we obtain an explicit representation of the boundary conditions when assuming (2.10).

$$\begin{aligned} &\tilde{p}_j = \bar{\rho}_j, \ j = j_1 \text{ and } j = j_2, \\ &f_{j_3}(\tilde{p}_{j_3}) = \sum_{j \in \delta_v^-} f_j(\tilde{p}_j). \end{aligned} \tag{2.14}$$

Thus, in this simple case of no backward-traveling wave, the coupling conditions at the junctions are essentially given by the drivers' wishes for a diverging junction. For a converging junction they are given by the equality of fluxes together with the requirement that the fluxes from the two ingoing roads are equal in the dense case.

Now, optimal control problems can be investigated. Typically, the average time spent by the drivers in the network is minimized. This means we consider the objective function

$$J((\alpha_v)_{v\in V}) = \int_0^T \sum_{j\in A} \int_{a_j}^{b_j} \rho_j(x,t)dxdt. \tag{2.15}$$

This function has to be minimized with respect to the control variables α_v for $v \in V$. We solve the problem:

$$\min_{0<\alpha_v<1, v\in V} J((\alpha_v)_{v\in V}) \tag{2.16}$$

$$\text{subject to } (\rho_j)_{j\in A} \text{ is solution of (2.1) with coupling conditions}$$
$$\text{at the junctions given by (2.9) and (2.13).}$$

A solution to this problem yields an optimal distribution of a traffic flow in a network including all dynamics, like jam propagation, etc. Alternatively we can optimize the above function in the case of no backward-going shock wave. This implies replacing conditions (2.9) and (2.13) by (2.11) and (2.14). However, even in this case optimization of networks with a large number of roads in reasonable time is beyond any computational possibility.

3. Simplified nonlinear model.

In this section the traffic flow model based on partial differential equations is reduced to a system of algebraic equations; cf. [17]. This is achieved by considering a simplified situation concerning the inflow into the network and tracking single waves running through the network. In contrast to the static network models often used by traffic engineers, the present approach still contains simplified dynamics, being at the same time not much more complicated and expensive from a computational point of view. For the following we assume that no backward-going shock waves appear; this means that the traffic is optimized in such a way that no traffic jam occurs. We start with an initially empty network and refer to the end of the section for the case of partially filled networks. Moreover, for simplicity, we restrict to constant inflow $\rho_{j,0}$ applied as boundary condition at the incoming road to the network. For the geometry of the network we use the same assumptions as in the previous section, i.e., we assume to have only junctions connecting at most three roads, like in Figure 2.1.

For simplicity of presentation, we assume to have an initially empty network, i.e., $\rho_{j,0}(x) = 0$. Extensions of the procedure to other cases are straightforward and discussed in Remarks 3.2 and 3.3.

The assumption of no backward-going shock waves is imposed as in (2.10), i.e.,

$$\rho_j(x,t) \le \sigma_j \ \forall j \in A.$$

We assign two values $p_j \in \mathbb{R}$ and $t_j \in \mathbb{R}^+$ to each road j of the network. The value p_j is an approximation of the density $\rho_j(x,t)$, while t_j denotes the arrival time of a wave at road j. The following bounds are obvious:

$$0 \le p_j \le \sigma_j, \ 0 \le t_j \le T. \tag{3.1}$$

Due to (2.10) we can express the coupling conditions (2.9) and (2.13) in the form (2.11) and (2.14), respectively. We translate them in terms of p_j and obtain

Case 1.

$$\delta_v^- = \{j_0\}, \delta_v^+ = \{j_1, j_2\},$$
$$p_{j_1} = f_{j_1}^{-1}(\alpha_v f_{j_0}(p_{j_0})), \;\; p_{j_2} = f_{j_2}^{-1}((1-\alpha_v) f_{j_0}(p_{j_0})).$$

Case 2.

$$\delta_v^- = \{j_1, j_2\}, \delta_v^+ = \{j_3\},$$
$$p_{j_3} = f_{j_3}^{-1}(f_{j_1}(p_{j_1}) + f_{j_2}^{-1}(f_j(p_{j_2}))).$$

For the ingoing roads to the network we set $p_j = \rho_{j,0}$. In Case 1 the parameters $0 < \alpha_v < 1$ distribute traffic at junction v in the direction of road i. Hence, p_j is determined solely by fulfilling the coupling conditions at the junctions.

Remark 3.1. As an example note that for a flux function of the type $f_j(x) = 4x(1 - x/M_j)$ the conditions read $2p_{j_1} = M_{j_1} - \sqrt{M_{j_1}^2 - \alpha_v M_{j_1} f_{j_0}(p_{j_0})}$ and similar for p_{j_2}. For the other junction type we obtain $2p_{j_3} = M_{j_3} - \sqrt{M_{j_3}^2 - M_{j_3}\chi}$ with $\chi = f_{j_1}(p_{j_1}) + f_{j_2}(p_{j_2})$.

We model the dynamics in the following way: the times t_j describe an approximation of the arrival times of the waves (generated as solutions to the hyperbolic equation (2.1)) at $x = a_j$ of road j. Since we cannot track the whole wave we use a discontinuity as approximation. Then the times t_j are defined by (3.2) and (3.3). In more detail, we have initially a Riemann problem

$$\partial_t \rho_j + \partial_x f_j(\rho_j) = 0, \quad \rho_j(x,0) = \begin{bmatrix} p_j & x \le a_j \\ 0 & x > a_j \end{bmatrix}$$

with concave flux function f_j. The rarefaction wave is the correct solution of the above problem which we approximated by a single discontinuity. The speed of the wave is approximated with the so-called Rankine–Hugoniot speed

$$s_j = \frac{f_j(p_j)}{p_j}.$$

The arrival times of the rarefaction wave are approximated as follows. For the ingoing road j_0 we set $t_{j_0} = 0$. In the case of a junction, where one road j_0 disperse in two others j_1, j_2, we set

$$t_{j_1} = t_{j_2} = t_{j_0} + \frac{b-a}{s_{j_0}}. \tag{3.2}$$

In the case of a junction with two incoming roads j_1, j_2 and one outgoing road j_3 the situation is more complicated. We set

$$t_{j_3} = \left(t_{j_1} + \frac{b-a}{s_{j_1}}\right)\frac{p_{j_1}}{p_{j_1} + p_{j_2}} + \left(t_{j_2} + \frac{b-a}{s_{j_2}}\right)\frac{p_{j_2}}{p_{j_1} + p_{j_2}}. \tag{3.3}$$

This choice is motivated by the following calculations. Let $t^{(1)} < t^{(2)}$ denote the time when the dicontinuity from road j_1 and j_2 reach the beginning of road j_3 with the values p_{j_1} and p_{j_2}. We again assume to have one discontinuity on road j_3 instead of rarefaction waves. The traveling speeds are given by $s^{(1)} = \frac{f_{j_3}(p_{j_1})}{p_{j_1}}$ and

$s^{(2)} = \frac{f_{j_3}(p_{j_3})-f_{j_3}(p_{j_1})}{p_{j_3}-p_{j_1}}$. The values p_{j_3} are determined by the coupling condition, i.e., $f(p_{j_3}) = f(p_{j_1}) + f(p_{j_2})$. Then we have

$$\int_0^T \int_a^b \rho_{j_3}(x,t)dxdt = (T - t^{(1)})(b-a)p_{j_1} - \frac{p_{j_1}}{2s^{(1)}}(b-a)^2 \\ + (T - t^{(2)})(b-a)(p_{j_3} - p_{j_1}) - \frac{p_{j_3} - p_{j_1}}{2s^{(2)}}(b-a)^2.$$

The idea is to approximate the above integral by $(T - t_{j_3})(b-a)p_{j_3} - \frac{p_{j_3}}{2s_{j_3}}(b-a)^2$ with $s_{j_3} = f_{j_3}(p_{j_3})/p_{j_3}$. If f is linear, then the correct choice for t_{j_3} is

$$t_{j_3} = t^{(1)}\frac{p_{j_1}}{p_{j_1}+p_{j_2}} + t^{(2)}\frac{p_{j_2}}{p_{j_1}+p_{j_2}}. \tag{3.4}$$

This is used as approximation in the nonlinear case. Finally, to obtain formula (3.3) we use (3.4) together with $t^{(1)} = (t_{j_1} + \frac{b-a}{s_{j_1}})$ and $t^{(2)} = (t_{j_2} + \frac{b-a}{s_{j_2}})$. Thus we have defined a purely algebraic dynamic model for traffic flow on road networks without backward-going shock waves. The model essentially describes how a wave of vehicles travels through the network. The speed of this wave is derived from the macroscopic model. It remains to reformulate the objective function (2.15) in terms of the simplified nonlinear model. We assumed to have a discontinuity that arrives at road j at time t_j and travels with speed s_j. Evaluating the integral $\int_0^T \int_{a_j}^{b_j} \rho_j(x,t)dxdt$ under this assumptions yields

$$\int_0^T \int_{a_j}^{b_j} \rho_j(x,t)dxdt = (T - t_j)(b_j - a_j)p_j - \frac{p_j}{2s_j}(b_j - a_j)^2.$$

Finally, the full simplified nonlinear model reads with $L_j = b_j - a_j$ and $s_j = f_j(p_j)/p_j$:

$$\left.\begin{array}{r} \text{For junctions of merging type} \\ t_k = \left(t_i + \dfrac{L_i}{s_i}\right)\dfrac{p_i}{p_i+p_j} + \left(t_j + \dfrac{L_j}{s_j}\right)\dfrac{p_j}{p_i+p_j} \\ p_k = f_k^{-1}(f_i(p_i) + f_j(p_j)) \\ \text{For junctions of dispersing type} \\ t_i = t_j = t_k + \dfrac{L_k}{s_k} \\ p_i = f_i^{-1}(\alpha_v f_k(p_k)), \quad p_j = f_j^{-1}((1-\alpha_v)f_k(p_k)) \\ \text{For the road entering the network} \\ p_j = \rho_0, t_j = 0 \end{array}\right\}. \tag{3.5}$$

The objective function reads

$$J((\alpha_v)_{v\in V}; T, \rho_0) = \sum_{j\in A}(T - t_j)L_j p_j - \frac{p_j}{2s_j}L_j^2. \tag{3.6}$$

Herein T is a fixed time and ρ_0 is the inflow to the network. It turns out that also for this simplification the minimization problem $\min J$ subject to the constraints above still needs large computation times for very large networks due to the nonlinearities in the coupling conditions for p_j and t_j. For numerical results we refer to the subsequent sections.

Remark 3.2. The treatment of a partially filled network is also possible. Assume we have the initial densities $\bar{p}_j$ on road j given, where all values are consistent with the conditions at the junctions. $\bar{p}_j$ is constant for the whole road such that $\bar{p}_j < \sigma_j$. We start, as before, with an inflow $\rho_0 < \sigma$. Then similar considerations yield the following expression for an integral on j with $(L := b - a)$:

$$\int_0^T \int_a^b \rho(x,t)dxdt = Lt_j\bar{p}_j + Lp_j(T - t_j) - \frac{p_j - \bar{p}_j}{2s_j}L^2, \tag{3.7}$$

where now s_j is given by

$$s_j = \frac{f_j(p_j) - f_j(\bar{p}_j)}{p_j - \bar{p}_j}. \tag{3.8}$$

Using this definition of s_j one can approximate the arrival times of the ingoing wave t_j in the same way as before with s_j as in (3.8).

Remark 3.3. Nonconstant initial data, for example, piecewise constant initial data, can be treated in the same way. We have to assume that the waves do not interact to track the arrival times of each wave.

4. Linear models. In this section the previously introduced model is further simplified obtaining a linear model accessible to discrete optimization techniques. The basic idea is the reformulation of the above model in terms of the flux $q_j := p_j u^e(p_j)$. We introduce the notation

$$\tau_j(q_j) := \frac{1}{u^e(f_j^{-1}(q_j))} \tag{4.1}$$

and obtain $p_j = q_j\tau_j(q_j)$. The coupling conditions at the junctions read

$$\left.\begin{array}{r} \text{For junctions of merging type} \\ q_k = q_i + q_j \\ \text{For junctions of dispersing type} \\ q_i + q_j = q_k \\ \text{For all roads} \\ M_j \geq q_j \geq 0 \end{array}\right\}. \tag{4.2}$$

Note that the control variable α_v does not appear in the above formulation. Therefore the values of q_i, q_j are not solely defined by q_k. The function J is given in terms of q_j by

$$J((q_j)_{j\in A}; T, \rho_0) = \sum_{j\in A}\Big(TL_j - t_jL_j - \frac{\tau_j(q_j)L_j^2}{2}\Big)\tau(q_j)q_j. \tag{4.3}$$

Then the complete model and the optimization problem reads

$$
\left.\begin{aligned}
\min_{(q_j)_{j\in A}} J((q_j)_{j\in A}; T, \rho_0)\\
\text{where for junctions of dispersing type}\\
t_i = t_j = t_k + L_k\tau_k(q_k)\\
q_i + q_j = q_k\\
\text{where for junctions of merging type}\\
t_k = \frac{(t_i + L_i\tau_i(q_i))q_i\tau_i(q_i)}{q_i\tau_i(q_i) + q_j\tau_j(q_j)} + \frac{(t_j + L_j\tau_j(q_j))q_j\tau_j(q_j)}{q_i\tau_i(q_i) + q_j\tau_j(q_j)}\\
q_k = q_i + q_j\\
\text{where for roads ingoing to the network}\\
q_j = f_0(\rho_0), t_j = 0\\
\text{where for all roads}\\
M_j \geq q_j \geq 0
\end{aligned}\right\}. \tag{4.4}
$$

This model is still equivalent to the nonlinear model described above. We derive different (linear!) models from this formulation and refer to the subsequent sections for numerical results.

4.1. Linear models with dynamics. The coupling conditions at the junctions are linear in q_j but nonlinear in t_j. We use different possibilites to linearize the coupling t_j. In the numerical tests it turns out that the crucial point is the proper discretization of the weight w appearing in the case of merging junctions, i.e.,

$$w_i(q_i, q_j) := \frac{q_i\tau_i(q_i)}{q_i\tau_i(q_i) + q_j\tau_j(q_j)}, \qquad w_j(q_i, q_j) := \frac{q_j\tau_j(q_j)}{q_i\tau_i(q_i) + q_j\tau_j(q_j)}.$$

We propose two different approaches and compare the results numerically in section 5.

A. We approximate

$$w_i, w_j \sim \tilde{w} = \frac{1}{2}$$

and calculate the first order Taylor expansion $\tilde{\tau}_j(q) = \tau_j(0) + q\tau_j'(0)$ as an approximation for $\tau_j(q)$. That means we linearize the model globally around 0. Neglecting higher order terms, we obtain the following linear equations:

$$
\begin{aligned}
&\text{Dispersing junctions}\\
&t_i = t_j = t_k + L_k\tilde{\tau}_k(q_k),\\
&\text{Merging junctions}\\
&t_k = \big(t_i + L_i\tilde{\tau}_i(q_i)\big)\cdot\tilde{w} + \big(t_j + L_j\tilde{\tau}_j(q_j)\big)\cdot\tilde{w}.
\end{aligned} \tag{4.5}
$$

B. Instead of linearizing the functions globally, we discretize the problem using piecewise linear approximations. The junctions of merging type are now approximated by piecewise linear functions on triangles, a more refined approximation as in case A. For each junction k of the merging type consider

$$a_k(q_i, q_j) := \frac{L_i q_i(\tau_i(q_i))^2}{q_i\tau_i(q_i) + q_j\tau_j(q_j)} + \frac{L_j q_j(\tau_j(q_j))^2}{q_i\tau_i(q_i) + q_j\tau_j(q_j)}. \tag{4.6}$$

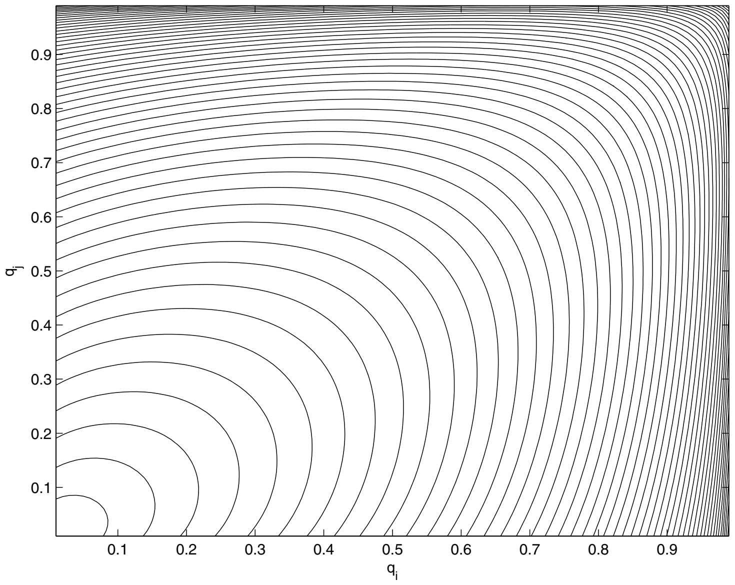


FIG. 4.1. *Contour lines of the nonlinear weight function $a_k(q_i, q_j)$ for $q_i, q_j \in [0, 1]$.*

As an example note that for $f(\rho) = 4\rho(1 - \rho/M_i)$ and $M_i = M_j = 1$ the contour lines of a_k are given in Figure 4.1.
We introduce $N_i \cdot N_j$ discretization points (ξ_v^k, η_w^k) with $0 = \xi_1^k < \xi_2^k < \cdots < \xi_{N_i-1}^k < \xi_{N_i}^k = M_i$ and $0 = \eta_1^k < \eta_2^k < \cdots < \eta_{N_j-1}^k < \eta_{N_j}^k = M_j$. Denote Δ a partition of the grid of discretization points into triangles and introduce a binary variable $y_{(p_1,p_2,p_3)}^k \in \{0, 1\}$ for each triangle $(p_1, p_2, p_3) \in \Delta$. The identification of the proper triangle corresponding to the incoming fluxes q_i, q_j is done by the next equations. Exactly one triangle has to be selected:

$$\sum_{(p_1,p_2,p_3)\in\Delta} y_{(p_1,p_2,p_3)}^k = 1. \tag{4.7}$$

Once one triangle is selected, the values of q_i, q_j can be encoded as convex combination of its corners. For this, introduce a continuous variable $\lambda_{v,w}^k \geq 0$ for each discretization point (ξ_v^k, η_w^k), which are coupled to q_i and q_j as follows:

$$q_i = \sum_{v=1}^{N_i} \sum_{w=1}^{N_j} \xi_v^k \cdot \lambda_{v,w}, \qquad q_j = \sum_{v=1}^{N_i} \sum_{w=1}^{N_j} \eta_w^k \cdot \lambda_{v,w}. \tag{4.8}$$

The convex combination condition is

$$\sum_{v=1}^{N_i} \sum_{w=1}^{N_j} \lambda_{v,w} = 1. \tag{4.9}$$

Only those three values $\lambda_{p_1}, \lambda_{p_2}, \lambda_{p_3}$ may be nonzero that correspond to the selected triangle by (4.7):

$$y_{(p_1,p_2,p_3)}^k \leq \lambda_{p_1} + \lambda_{p_2} + \lambda_{p_3} \ \forall \ (p_1, p_2, p_3) \in \Delta. \tag{4.10}$$

To introduce $\tilde{a}_k$ as a piecewise linear approximation of $a_k(q_i, q_j)$, we add the following equation to the model:

$$\tilde{a}_k = \sum_{v=1}^{N_i} \sum_{w=1}^{N_j} a_k(\xi_v, \eta_w) \cdot \lambda_{v,w}. \tag{4.11}$$

The junctions of dispersing type are approximated as in case A, whereas for the junctions of merging type, we use a blending of $\tilde{a}$ as above and $\tilde{w}$ as in case A:

$$\begin{aligned} &\text{Dispersing junctions} \\ &t_i = t_j = t_k + L_k \tilde{\tau}_k(q_k), \\ &\text{Merging junctions} \\ &t_k = (t_i + t_j) \cdot \tilde{w} + \tilde{a}_k. \end{aligned} \tag{4.12}$$

For any linearization A or B we linearize the objective function (4.3) as follows. For every $j \in A$ we introduce D_q variables $0 \leq y_i^j \leq \frac{M_j}{D_q}$ and let the flux be represented by

$$q_j = \sum_{i=1}^{D_q} y_i^j. \tag{4.13}$$

Functional J is approximated by

$$\tilde{J}((q_j)_{j\in A}; T, \rho_0) := \sum_{j \in A} z_j, \tag{4.14}$$

where we introduce for every arc $j \in A$ and every $k = 1, \dots, D_t$ the inequality

$$\begin{aligned} &\sum_{i=1}^{D_q} \left(G\left(\frac{(i+1)\cdot M_j}{D_q}, T \cdot 2^{k-D_t} \right) - G\left(\frac{i \cdot M_j}{D_q}, T \cdot 2^{k-D_t} \right) \right) \cdot \frac{D_q}{M_j} \cdot y_i^j \\ &\leq z_j + M \cdot (1 - u_{jk}), \end{aligned} \tag{4.15}$$

where M is a sufficiently big value and G is defined by

$$G(\xi, \zeta) := \left(T - \zeta - \frac{\tau(\xi) L_j}{2} \right) \cdot L_j \tau(\xi) \xi, \tag{4.16}$$

and we assume that $G(\cdot, \zeta)$ is convex for every $\zeta \in [0, T]$. Moreover, u_{jk} is a binary variable for every $j \in A$ and $k = 1, \dots, D_t$, where $u_{jk} = 1$ if $t_j \leq T \cdot 2^{k-D_t}$. Thus we add the following inequalities to the model:

$$t_j \geq T \cdot 2^{k-D_t} (1 - u_{jk}) \tag{4.17}$$

for all $j \in A$ and $k = 1, \dots, D_t$. Summarizing, we obtain a linear mixed-integer model with dynamics given by

$$\left.\begin{array}{r} \min\limits_{z_j, y_i^j, u_{jk}, \lambda^k_{v,w}, q_j, t_j} \tilde{J} \\ \text{where for junctions of dispersing type} \\ t_i = t_j = t_k + L_k \tilde{\tau}_k(q_k) \\ q_i + q_j = q_k \\ \text{where for junctions of merging type} \\ \text{case A: } t_k = \big(t_i + L_i \tilde{\tau}_i(q_i)\big) \cdot \tilde{w} + \big(t_j + L_j \tilde{\tau}_j(q_j)\big) \cdot \tilde{w} \\ \text{case B: } t_k = (t_i + t_j) \cdot \tilde{w} + \tilde{a}_k \\ q_k = q_i + q_j \\ \text{where for roads ingoing to the network} \\ q_j = f_0(\rho_0), t_j = 0 \\ \text{where for all roads} \\ M_j \geq q_j \geq 0 \\ \text{and where } u_{jk}, z_j, \lambda^k_{v,w} \text{ are coupled as introduced} \end{array}\right\} . \tag{4.18}$$

Remark 4.1. In the above modelling we set the discretization points as $T 2^{k-D_t}$ for $k = 1, \dots, D_t$. This produces a log-scale distribution of discretization points in $[0, T]$. Other distributions are also possible. For example, if we identically distribute we obtain

$$t_j \geq T \frac{k-1}{D_t - 1}(1 - u_{jk}) \tag{4.19}$$

instead of (4.17). The proper choice depends on the size of the network geometry and the scaling of T.

4.2. Linear model without dynamics. We assume $t_j = 0$ which models a static traffic flow network. We obtain a linear function $\tilde{J}$ from (4.3) by a piecewise linear approximation of J. For this, we introduce D variables $0 \leq y_i^j \leq \frac{M_j}{D}$ for every arc $j \in A$. Then the flux q_j is represented by

$$q_j = \sum_{i=1}^{D} y_i^j. \tag{4.20}$$

Now J is approximated by

$$\tilde{J}(q_j; T, \rho_0) := \sum_{j \in A} \sum_{i=1}^{D} \left(G\left(\frac{(i+1) \cdot M_j}{D} \right) - G\left(\frac{i \cdot M_j}{D} \right) \right) \cdot \frac{D}{M_j} \cdot y_i^j, \tag{4.21}$$

where G is defined by

$$G(\xi) := \left(T L_j - \frac{\tau(\xi) L_j^2}{2} \right) \cdot \tau(\xi) \cdot \xi. \tag{4.22}$$

Again, we assume $G(\cdot)$ to be convex. Summarizing, we have the following model:

$$\left.\begin{array}{r} \min\limits_{y_i^j, q_j} \tilde{J} \\ \text{where for roads connected to a junction } v \\ \sum\limits_{j \in \delta_v^+} q_j = \sum\limits_{j \in \delta_v^-} q_j \\ \text{where for roads ingoing to the network} \\ q_j = f_0(\rho_0) \\ \text{where for all roads} \\ M_j \geq q_j \geq 0 \\ \text{and } y_i^j \text{ satisfies (4.20)} \end{array}\right\} . \tag{4.23}$$

4.3. Linearization of monotone, nonconvex functions. In both linearizations above we assumed $G(\cdot, \zeta)$ and $G(\cdot)$, respectively, to be convex functions. The convexity depends on $\tau(\cdot)$ and is satisfied for those functions τ we study within this article. However, other choices for τ are possible, where G is nonconvex. But if G happens to be monotone (and nonconvex), it is still possible to obtain a linearization. We present the necessary changes to the model only in the case without dynamics. In the dynamic case they are similar.

We introduce D variables $y_i^j \geq 0$ for every arc $j \in A$. Since G is nonconvex, they cannot be coupled to the flux q_j as simple as in (4.20). Instead we use the following inequalities:

$$q_j \leq y_i^j + \frac{M_j}{D} \cdot i \;\forall i = 1, \dots, D, j \in A. \tag{4.24}$$

Now J is approximated by

$$\begin{aligned} \tilde{J}((q_j)_{j \in A}; T, \rho_0) := \sum_{j \in A} \frac{D}{M_j} \Bigg(& G\left(\frac{M_j}{D}\right) \cdot y_0^j \\ & + \sum_{i=2}^{D} \left(G\left(\frac{(i+1) \cdot M_j}{D}\right) - 2 \cdot G\left(\frac{i \cdot M_j}{D}\right) + G\left(\frac{(i-1) \cdot M_j}{D}\right)\right) \cdot \frac{D}{M_j} \cdot y_i^j \Bigg), \end{aligned} \tag{4.25}$$

where G is as in (4.22).

4.4. Min cost flow model. We again assume $t_j = 0$, i.e., the static network case. Instead of a piecewise linear approximation of our objective function (4.3) we additionally assume a simplified dynamic: if the function

$$u_j^e(\rho) = c_j$$

is constant for all j, then by definition

$$\tau(q_j) = \frac{1}{c_j}.$$

The function (4.3) reads

$$\bar{J}((q_j)_{j \in A}; T, \rho_0) = \sum_{j \in A} \omega_j q_j, \tag{4.26}$$

where ω_j are constants given by

$$\omega_j = T\frac{L_j}{c_j} - \frac{L_j^2}{2c_j^2}.$$

Together with the linear coupling conditions and the lower bounds for q_j we obtain the classical min cost flow problem:

$$\left.\begin{array}{r} \min\limits_{q_j} \sum\limits_{j\in A} \omega_j q_j \\ \text{where for roads connected to a junction } v \\ \sum\limits_{j\in\delta_v^+} q_j = \sum\limits_{j\in\delta_v^-} q_j \\ \text{where for roads ingoing to the network} \\ q_j = f_0(\rho_0) \\ \text{and where for all roads} \\ M_j \geq q_j \geq 0 \end{array}\right\}. \tag{4.27}$$

Remark 4.2. Unfortunately, the assumption $u_j^e = \text{constant}_j$ is not a realistic approximation of a typical fundamental diagram. For reasonable fundamental diagrams we refer to [22]. At least we have to assume $u_j^e(x)$ is linear.

We have the following remark on the relation of the linear models given above and known other approaches.

Remark 4.3. We note that for the linear models there is a strong connection to the traffic flow models proposed by Möhring et al.; see, for example, [26, 25, 24]. Especially, the occurence of the so-called transit-times shows the close relation between the models. However, the cost function for the linear problem differs due to the derivation starting with partial differential equations. The starting point of the models introduced in [25, 24] is the transit times τ_e which are assumed to be known functions. They describe the time needed by a flow to pass the arc e. In our formulation the transit times are the functions derived by equation (4.1), i.e.,

$$q \to \tau_j(q)L_j.$$

The case of constant transit times is called “static flow problems” in [25]. In our introduced model this reflects the situation $u_j^e(\rho)$ constant. As pointed out by Möhring et al. this cannot be a realistic assumption. Therefore, they introduced “static traffic flows with congestion” by assuming a dependency of τ_e on q. In our model this approach is reflected by the introduced linear model without dynamics. However, we see by our derivation that congestion in the form of backward-going shock waves is not covered by those models; cf. numerical results below.

Remark 4.4. In the previous sections we derived a hierachy of models. The most accurate and detailed model is based on partial differential equations and is given by (2.16). In a first step we reduce the PDE model to a set of nonlinear equations. We call this model simplified nonlinear model and refer to (3.5). Further simplifications yield linear models with dynamics (4.18). Two different modelings at the junction (denoted by case A and B) are considered. Neglecting the dynamics we obtain the linear model without dynamics which is given by (4.23). Finally, we reduced the model to a min cost flow problem (4.27).

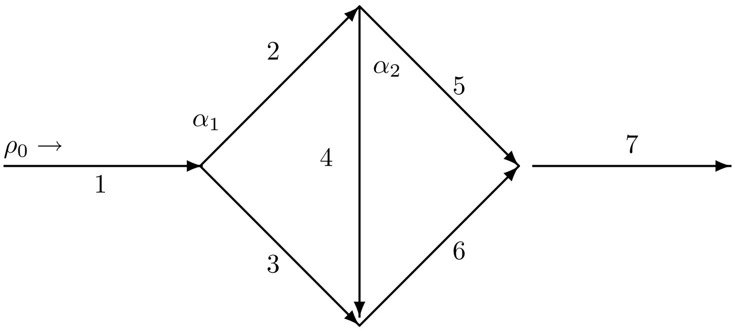


FIG. 5.1. *Example of a network.*

5. Results. We compared the computing times for different models and networks. All results have been obtained on a 1.0 Ghz Pentium III processor machine with 2 GB RAM, 256 KB first-level cache, and Debian Linux v3.0 as operating system.

5.1. Test case for comparing network models. For the purpose of comparing the models we introduce a network with two controls and seven roads, as in Figure 5.1. As an example we use the smooth and concave family of flux functions

$$f_j(\rho) = \rho u_j^e(\rho) = 4\rho(1 - \rho/M_j). \tag{5.1}$$

The function $\tau_j(\rho)$ is then given by (4.1), i.e.,

$$\tau_j(q) = \frac{M_j}{2\left(M_j + \sqrt{M_j^2 - M_j q}\right)}, \quad 0 \leq q \leq M_j. \tag{5.2}$$

If not stated otherwise we assume

$$T = 5 \text{ and } L_j := b_j - a_j = 1 \ \forall j = 1, \dots, 7. \tag{5.3}$$

We define q_0 to be the known inflow given at $x = a_1$.

5.1.1. Comparison of the values of the objective function. We compare the derived models on the sample network. We compute the objective function of the corresponding model for all admissible choices of the control variables α_1 and α_2. In the context of the linear models this implies computing the objective for all choices $q_1, \dots, q_7$ satisfying the constraints. As described in section 4 the fluxes q_j and the controls are related. For example, we obtain for the first roads of our sample network

$$q_1 = q_0, \ q_2 = \alpha_1 q_1, \ q_3 = (1 - \alpha_1) q_1.$$

In all subsequent plots we draw contour lines of the objective function against α_1 and α_2. We choose different maximal fluxes M_j on the roads to obtain different test cases.

Test Case 1. Free flow.

We set $M_j = 1$ for all roads and $q_0 = 96\% M_1$. We compute the objective function (2.15) by a trapezoid rule. The underlying partial differential equations is solved by a first order Godnuov scheme with $N = 100$ discretization points for each road j. The objective function (3.6) is computed by the formulas given in section 3. For the linear models we computed the function (4.25) with $D = 1000$ variables for each arc

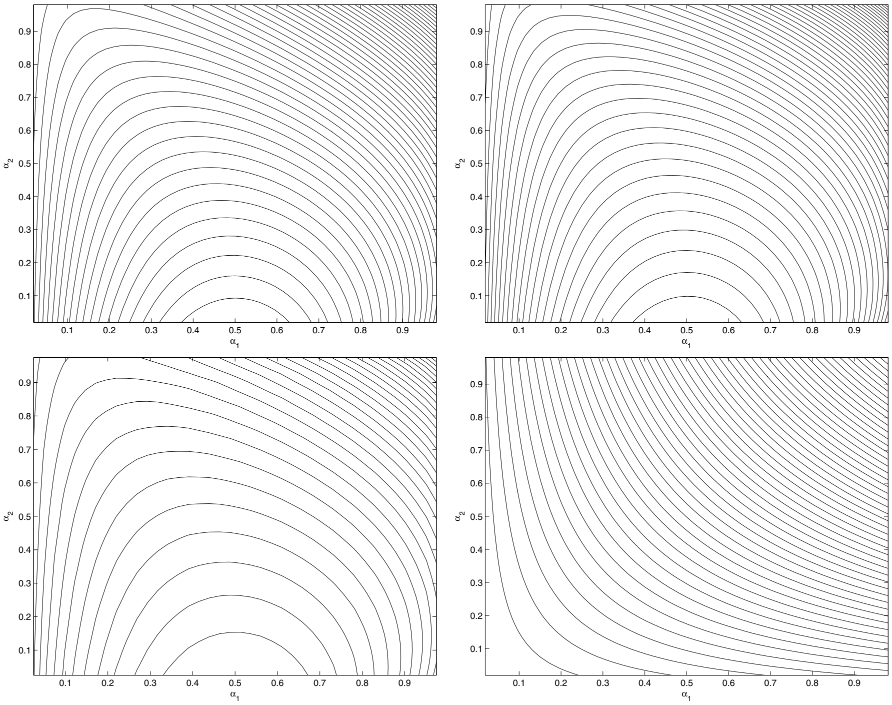


FIG. 5.2. *Test Case* 1: *Contour lines of the functions for partial differential equation* (2.15), *simplified nonlinear* (3.6), *linear without dynamics* (4.25), *and min cost flow model* (4.26) *(top left to bottom right).*

j. Note, that the function $\xi \to G(\xi)$ is at least monotone for the choice (5.1). For comparison we include a plot of the function for the min cost flow problem (4.26), where we set $u_j^e(\rho) = 2$ for this calculation. The results are given in Figure 5.2. The minimizer of all problems is $(\alpha_1, \alpha_2) = (\frac{1}{2}, 0)$. In case of the min cost flow problem we lose the uniqueness of the minimizer. Furthermore, the qualitative behavior differs significantly from the other models.

Test Case 2. Backward-going shock waves.

When deriving the simplified models we neglected backward-going shock waves. This was an essential part of the simplification of the dynamics. We compare the simplified nonlinear model (3.5) with the model based on partial differential equations (2.16) in a case with backward-going shock waves. We set $M_1 = M_2 = M_4 = M_6 = 2$, $M_3 = 1$, $M_5 = 0.5$, and $q_0 = 75\% M_1$. We used the same discretization as previously and compare the contour lines of (2.15) and (3.6) in Figure 5.3. We observe that in the PDE case the domain of admissible controls is larger than in the case of the simplified nonlinear model. This effect is due to backward-going shock waves which occur on some roads in the PDE model. Controls generating these waves are not admissible in the simplified nonlinear model. In our special case the region for the optimal control coincides. We skip results on the linear model since they approximate the algebraic model.

Test Case 3. Neglecting dynamics.

For the linear and simplified nonlinear models we considered models with and without dynamics. In this test case we highlight the influence of the correct modeling

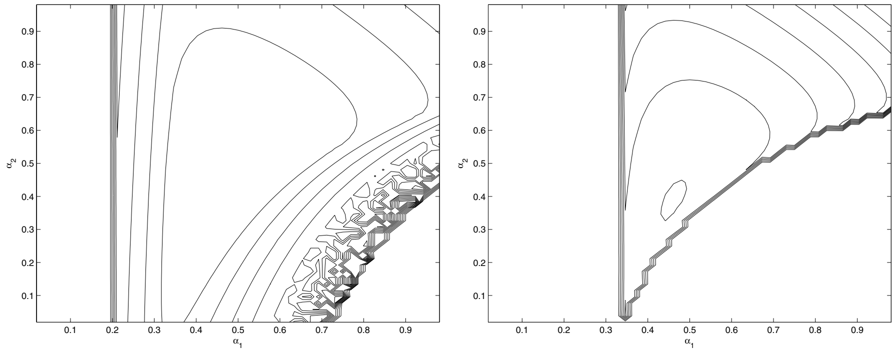


FIG. 5.3. *Test Case* 2: *Contour lines for the functions for PDE and simplified nonlinear model (left to right).*

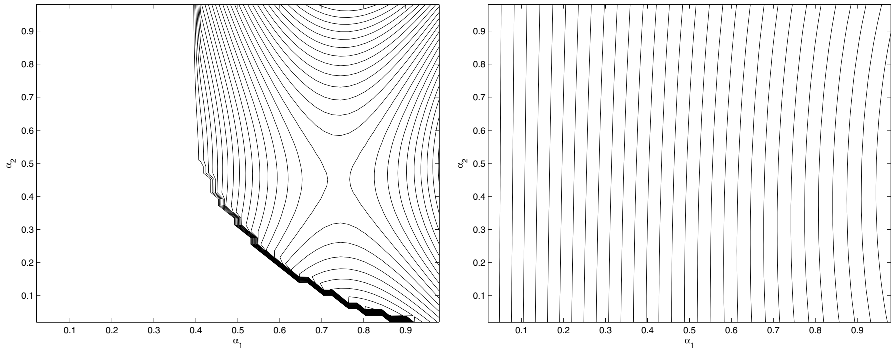


FIG. 5.4. *Test Case* 3: *Contour lines for the functions* (3.6) *for simplified model with (left) and without dynamics.*

of the dynamics. We compute the corresponding functions (4.25) and (4.14) for the following setting: $M_j = 2$, $q_0 = 96\% M_1$ and $L_1 = L_7 = L_5 = 2$, $L_4 = L_6 = 1$, $L_2 = 2.5$, $L_3 = 15$. In the dynamic case we allow only those controls α_1, α_2 where the incoming flux reaches $x = b_7$ with $t_7 \leq T$. The simplified nonlinear model is given in section 3. Besides this model we compute (3.6) for the case

$$t_j = 0 \ \forall j = 1, \ldots, 7,$$

too. This allows us to study the effect of the dynamics. The results are given in Figure 5.4. Note that in the region $\alpha_1 < 30\%$ the routed traffic does not reach the outgoing road. Furthmore, the functions differ significantly in their qualitative behavior.

Test Case 4. Discretization points for linear models.

The linear models depend strongly on the number of discretization points D. We study the qualitative behavior of the contour lines when decreasing D. We consider the linear model without dynamics and its objective function (4.25). We set $M_j = 1$, $q_0 = 0.96\%$. We plot the contour lines for $D = 5$ and $D = 25$ discretization points in Figure 5.5.

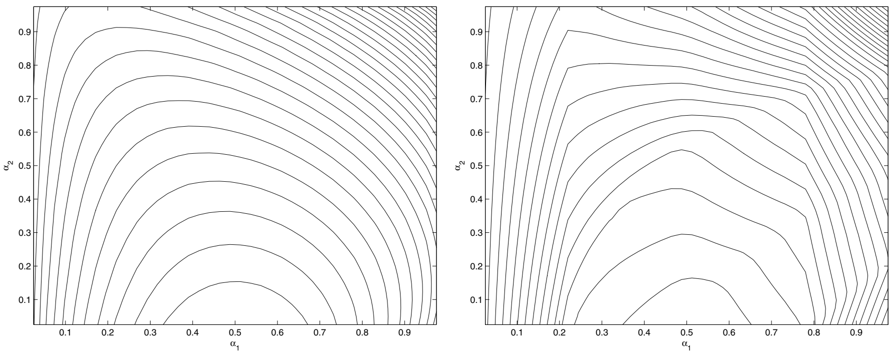


FIG. 5.5. *Test Case* 4: *Contour lines for the functions* (4.25) *for linear model without dynamics and varying $D = 25$ (left), resp., $D = 5$ (right).*

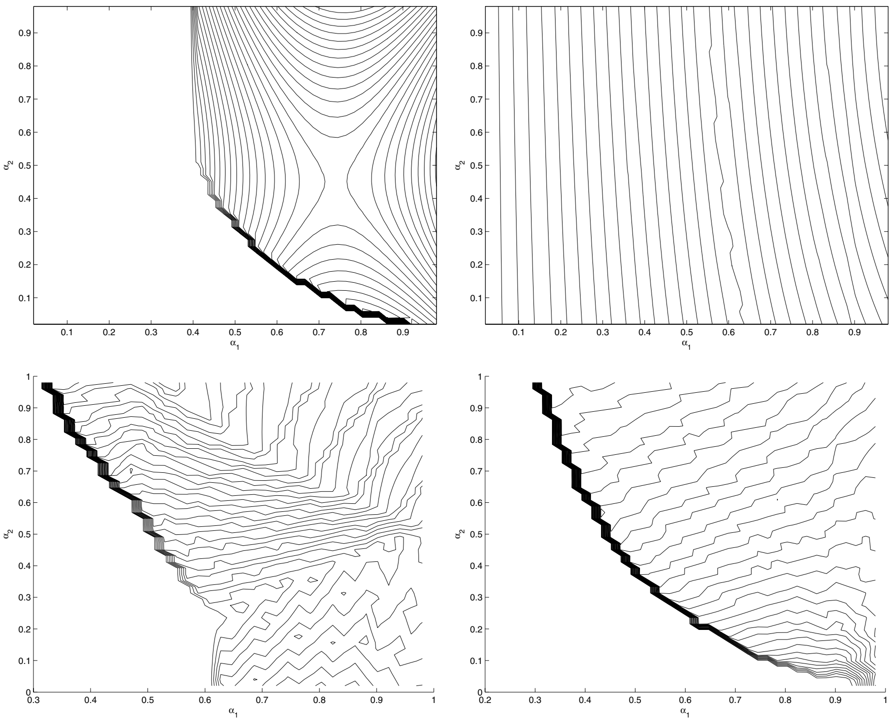


FIG. 5.6. *Test Case* 5: *Contour lines for the functions for simplified nonlinear* (3.6) *and different linear models with dynamics* (4.14). *Simplified nonlinear (upper left), Case A (upper right), Case B with $N_i = N_j = 5$, resp., $N_i = N_j = 25$ (lower row left to right).*

Test Case 5. Linearization of the dynamics.

In this case we consider the influence of the various possible discretizations for the coupling in t_j. We consider the same setting as in Test Case 3. We compare the qualitative behavior of the objective function for the simplified nonlinear model with the linear models with dynamics given in section 4.1. We compare the different discretization, Cases A and B. The results are given in Figure 5.6. We used $D_q = D_t = 100$ variables for the discretization of the flux and the time on each road for

Table 5.1
CPU times for sample network and different models.

Model and Scheme	Parameters	CPU time
Godunov scheme for PDE model	N=100	135.65 s
Godunov scheme for PDE model	N=50	45.17 s
Simplified nonlinear model		0.05 s
Linear Model with dynamics (B)	$D_q = D_t = 100$, $N_i \cdot N_j = 25$	0.02 s
Linear Model without dynamics	$D_q = 100$	0.01 s

any linearized model. We calculate Cases B with $N_i = N_j = 5$ and $N_i = N_j = 25$, respectively, discretization points for each junction of the merging type.

5.1.2. Optimization on the sample network. We consider the optimization problems introduced and compare computing times on the sample network.

As in the previous section we solved the partial differential equations model (2.16) with a Godunov scheme with N discretization points. The objective function is discretized using the trapezoid rule. For standard nonlinear optimization routines we need at least the gradient of the objective function. We compute an approximation by finite differences. Other approaches (using adjoint formulas) are investigated in [13]. In case of the simplified nonlinear model (3.5) the gradient can be calculated analytically.

For all nonlinear optimization problems the L-BFGS-B optimizer [4, 34, 5] is used. This method is a gradient projection method with a limited memory BFGS approximation of the Hessian and is able to consider bound constraints. The default settings are $m = 17$, $factr = 1.d + 5$, $pgtol = 1.d - 8$, and $isbmin = 1$.

The linear model without dynamics (4.23) is a pure linear programming problem. We solved it using ILOG CPLEX 8.1 [20]. As a default strategy, we set the network simplex method to solve the linear programs. For our test cases, this method outperforms other solution techniques, such as primal or dual simplex. In case of the linear model with dynamics (4.18) we have a mixed-integer problem. Among the currently most successful methods for solving these problems are linear programming–based branch-and-bound algorithms, where the underlying linear programming relaxations are possibly strengthend by cutting planes. Fortunately, today's state-of-the-art commercial MIP solvers (such as CPLEX [20]) can handle mixed-integer programs even for our large-size problem instances.

For the setting of Test Case 1 we have the following result on the computational times (CPU times); see Table 5.1. The parameters (D_q, D_t) describe the discretization of the nonlinear function. The parameter $N_i \cdot N_j$ describes the total number of discretization points for the function $a_k(\cdot,\cdot)$ at the merging junctions. Therefore, the only models reasonable to test on large-scale networks are the simplified nonlinear and the linear models.

5.2. Large-scale network optimization. The network considered next is shown in Figure 5.7.

There, every node in the top row is controllable via a separate control α_v. There are only one source and one sink. The prescribed inflow is again $q_0 = 96\% M_1$ and all streets have the same maximal flux, $M_j = 1.0$ Then, the optimal controls are $\alpha_1 = 0.5$ and $\alpha_v = 1.0 \; \forall \, v \neq 1$. The results are given in Table 5.2. The number of discretization points for the flux q per road is denoted by D_q and for the time by D_t. The number of discretization points for each function a_k (cf. (4.6)) in model B is denoted by $N_i N_j$. Note that all nodes in the bottom row are of the merging type.

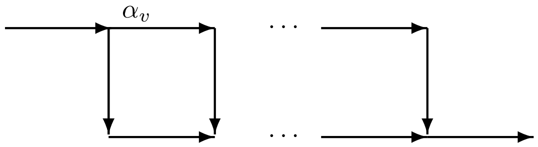


FIG. 5.7. *General layout of a large scale network.*

TABLE 5.2

CPU times for large-scale networks. n.a.-not available: those quantities do not appear in the corresponding models.

Model	# Roads	D_q	D_t	N_iN_j	Gap	CPU time
Simplified nonlinear model	240	n.a.	n.a.	n.a.	n.a.	6 s
Linear with dynamics (B)		10	10	25	1%	11 m
		10	10	25	10%	3.8 m
		10	10	9	0.1%	2.6 m
		10	10	9	10%	57 s
Linear with dynamics (A)		100	10	n.a.	0.1%	33.08 s
		10	10	n.a.	0.1%	4.78 s
Linear without dynamics		100	n.a.	n.a.	0.1%	<0.01 s
Simplified nonlinear model	1,500	n.a.	n.a.	n.a.	n.a.	57 m
Linear with dynamics (B)		10	10	25	10%	4.7 h
		10	10	9	10%	26 m
		5	5	9	10%	5 m
Linear with dynamics (A)		100	10	n.a.	0.1%	180.01 m
		10	10	n.a.	0.1%	13.69 m
Linear without dynamics		1000	n.a.	n.a.	0.1%	24.98 s
		100	n.a.	n.a.	0.1%	12.75 s
		5	n.a.	n.a.	0.1%	1.8 s
Simplified nonlinear model	15,000	n.a.	n.a.	n.a.	n.a.	>4d
Linear with dynamics (B)		5	5	9	10%	6.2 h
Linear without dynamics		100	n.a.	n.a.	n.a.	22.79 m
		10	n.a.	n.a.	n.a.	7.33 m
Linear without dynamics	150,000	10	n.a.	n.a.	n.a.	16.77 h

To improve the performance of CPLEX we increased the optimality gap from 0.001% (default setting) to 10%. We present results for other optimality gaps, too.

6. Summary.

- A hierachy a traffic network models ranging from PDE models to simple combinatorial models of min cost flow type has been developed.
- A variety of different network topologies has been investigated. Combinatorial and continuous optimization approaches using these models have been implemented and compared.
- The investigation shows the advantages and disadvantages of the different models and optimization procedures. In particular, for very large networks discrete optimization procedures are superior in terms of computation time.
- However, the simplified models developed here do not contain more complicated dynamic situations like backward-going shocks, i.e., traffic jams. To include such situations one has to use the original PDE model or to derive more sophisticated models from the PDE network.
- One could combine the models described here in a coupling strategy for very large networks. The main part of the network can be simulated using simple linear models. More complicated dynamic models may be used in regions where the detailed dynamic behavior is important.

Acknowledgment. We thank Rolf Möhring for his valuable comments on time-dependent network flow models.

REFERENCES

[1] A. Aw and M. Rascle, *Resurrection of second order models of traffic flow*, SIAM J. Appl. Math., 60 (2000), pp. 916–938.

[2] C. Bardos, A. Y. LeRoux, and J. C. Nedelec, *First order quasilinear equations with boundary conditions*, Comm. Partial Differential Equations, 4 (1979), pp. 1017–1034.

[3] M. Blinkin, *Problem of optimal control of traffic flow on highways*, Automat. Remote Control, 37 (1976), pp. 662–667.

[4] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, *A limited memory algorithm for bound constrained optimization*, SIAM J. Sci. Comput., 16 (1995), pp. 1190–1208.

[5] R. Byrd, J. Nocedal, and R. Schnabel, *Representations of quasi-Newton matrices and their use in limited memory methods*, Math. Program., 63 (1994), pp. 129–156.

[6] M. Carey and E. Subrahmanian, *An approach for modeling time-varying flows on congested networks*, Transp. Res. B, (2000), pp. 157–183.

[7] G. M. Coclite and B. Piccoli, *Traffic flow on a road network*, SIAM J. Math. Anal., to appear.

[8] R. Colombo, *Hyperbolic phase transitions in traffic flow*, SIAM J. Appl. Math., 63 (2002), pp. 708–721.

[9] C. M. Dafermos, *Polygonal approximations of solutions of the initial value problem for a conservation law*, J. Math. Anal. Appl., 38 (1972), pp. 33–41.

[10] L. R. Ford and D. R. Fulkerson, *Constructing maximal dynamic flows from static flows*, Oper. Res., (1958), pp. 419–433.

[11] J. Greenberg, *Extension and amplification of the Aw-Rascle model*, SIAM J. Appl. Math., 62 (2001), pp. 729–745.

[12] J. Greenberg, A. Klar, and M. Rascle, *Congestion on multilane highways*, SIAM J. Appl. Math., 63 (2003), pp. 818–833.

[13] M. Gugat, M. Herty, A. Klar, and G. Leugering, *Adjoint calculus for traffic flow networks*, JOTA, 126 (2005), pp. 589–616.

[14] M. Günther, A. Klar, T. Materne, and R. Wegener, *Multivalued fundamental diagrams and stop and go waves for continuum traffic flow equations*, SIAM J. Appl. Math., 64 (2003), pp. 468–483.

[15] D. Helbing, *Improved fluid dynamic model for vehicular traffic*, Phys. Rev. E, 51 (1995), p. 3164.

[16] M. Herty and A. Klar, *Modelling, simulation and optimization of traffic flow networks*, SIAM J. Sci. Comput., 25 (2003), pp. 1066–1087.

[17] M. Herty and A. Klar, *Simplified dynamics and optimization of large scale traffic networks*, Math. Models Methods Appl. Sci., 14 (2004), pp. 1–23.

[18] H. Holden and N. H. Risebro, *A mathematical model of traffic flow on a network of unidirectional road*, SIAM J. Math. Anal., 4 (1995), pp. 999–1017.

[19] H. Holden and N. H. Risebro, *Front Tracking for Hyperbolic Conservation Laws*, Springer, New York, Berlin, Heidelberg, 2002.

[20] ILOG CPLEX Division, *Using the CPLEX Callable Library*, available online from http://www.cplex.com.

[21] O. Jahn, R. H. Möhring, and A. S. Schulz, *Optimal routing of traffic flows with length restrictions in networks with congestion*, in Operations Research Proceedings, K. Inderfurth et al., eds., Springer-Verlag, Berlin, 2000, pp. 437–442.

[22] A. Klar, R. D. Kuehne, and R. Wegener, *Mathematical models for vehicular traffic*, Surveys Math. Indust., 6 (1996), p. 215.

[23] A. Klar and R. Wegener, *Kinetic derivation of macroscopic anticipation models for vehicular traffic*, SIAM J. Appl. Math., 60 (2000), pp. 1749–1766.

[24] E. Köhler, K. Langkau, and M. Skutella, *Time-expanded graphs for flow-dependent transit times*, in Algorithms—ESA '02, Lecture Notes in Comput. Sci. 2461, R. Möhring and R. Raman, eds., Springer, Berlin, 2002, pp. 599–611.

[25] E. Köhler, R. H. Möhring, and M. Skutella, *Traffic networks and flows over time*, in DFG Research Center: Mathematics for Key Technologies, J. Kramer, ed., Berliner Mathematische Gesellschaft, 2002, pp. 49–70.

[26] E. Köhler and M. Skutella, *Flows over time with load-dependent transit times*, in Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms, 2002, pp. 174–183.

[27] A. Kotsialos, M. Papageorgiou, M. Mangeas, and H. Haj-Salem, *Coordinated and integrated control of motorway networks via non-linear optimal control*, Trans. Res. C, 10 (2002), pp. 65–84.

[28] R. D. Kühne, *Macroscopic freeway model for dense traffic*, in 9th International Symposium on Transportation and Traffic Theory, N. Vollmuller, ed., VNU Science Press, Utrecht, the Netherlands, 1984, pp. 21–42.

[29] K. Langkau, *Flows over Time with Flow-Dependent Transit Times*, Ph.D. thesis, Fakultät II, Institut für Mathematik, Technische Universität Berlin, 2003.

[30] M. J. Lighthill and J. B. Whitham, *On kinematic waves*, Proc. Roy. Soc. Edinburgh A, 229 (1983), pp. 281–345.

[31] P. Nelson, *A kinetic model of vehicular traffic and its associated bimodal equilibrium solutions*, Transport Theory Statist. Phys., 24 (1995), pp. 383–408.

[32] H. J. Payne, *FREFLO: A macroscopic simulation model of freeway traffic*, Transp. Res. Record, 722 (1979), pp. 68–75.

[33] G. B. Whitham, *Linear and Nonlinear Waves*, Wiley, New York, 1974.

[34] C. Zhu, R. H. Byrd, J. Lu, and J. Nocedal, *L-BFGS-B: Fortran Subroutines for Large Scale Bound Constrained Optimization*, Tech. Report, NAM-11, EECS Department, Northwestern University, Evanston, IL, 1994.

SIAM J. OPTIM.
Vol. 16, No. 4, pp. 1177–1200

A SQP-SEMISMOOTH NEWTON-TYPE ALGORITHM APPLIED TO CONTROL OF THE INSTATIONARY NAVIER–STOKES SYSTEM SUBJECT TO CONTROL CONSTRAINTS[*]

M. HINTERMÜLLER[†] AND M. HINZE[‡]

Abstract. SQP methods for the optimal control of the instationary Navier–Stokes equations with pointwise constraints on the control are considered. Due to the presence of the constraints, the quadratic subproblems (QPs) of SQP require a more sophisticated solver when compared to the unconstrained case. In this paper, a semismooth Newton method is proposed for efficiently solving the QPs. The convergence analysis, which is performed in an appropriate function space setting, relies on the concept of slant differentiability for proving locally superlinear convergence of the QP-solver. For the analysis of the outer SQP-iteration a generalized equations approach is utilized. Sufficient conditions for guaranteeing strong regularity of the generalized equation are established which, in turn, allows one to argue a quadratic rate of convergence of the SQP-method. The paper ends with a report on numerical results supporting the theoretical findings.

Key words. box constraints, generalized equations, Navier–Stokes, optimal control, semismooth Newton, sequential quadratic programming

AMS subject classifications. 90C55, 65K10, 49M37

DOI. 10.1137/030601259

1. Introduction.

In this paper we continue our efforts in devising efficient numerical algorithms for the optimal control of the instationary Navier–Stokes equations. While the contributions in [22, 27] focus on unconstrained problems with respect to the control, the goal of the present work is to extend the framework in [22] to problems involving box constraints on the control variable.

Throughout we focus on the problem

$$
\begin{array}{lll}
 & \text{minimize}\quad J(y,u) & \text{over}\quad (y,u)\in W\times U_{\mathrm{ad}}\\
 & \text{subject to} & \\
(1.1) & \quad \dfrac{\partial y}{\partial t}+(y\cdot\nabla)y-\nu\Delta y+\nabla\varpi=u & \text{in } Q=(0,T)\times\Omega,\\
 & \quad \operatorname{div} y=0 & \text{in } Q,\\
 & \quad y(t,\cdot)=0 & \text{on } \Sigma=(0,T)\times\partial\Omega,\\
 & \quad y(0,\cdot)=y_0 & \text{in } \Omega
\end{array}
$$

with U_{ad} denoting the closed convex subset of the Hilbert space U of controls given by

$$
U_{\mathrm{ad}}=\{v\in U : a\le v\le b \text{ a.e. in } Q\}, \tag{1.2}
$$

[*] Received by the editors October 17, 2003; accepted for publication (in revised form) October 17, 2005; published electronically March 3, 2006.
http://www.siam.org/journals/siopt/16-4/60125.html

[†] Department of Computational and Applied Mathematics, Rice University, Houston, TX, and Department of Mathematics, University of Graz, Heinrichstraße 36, 8010 Graz, Austria (michael.hintermueller@uni-graz.at). This author was supported by Spezialforschungsbereich F013 funded by the Austrian Science Foundation FWF.

[‡] Department of Mathematics, Technical University of Dresden, Dresden, Germany (hinze@math.tu-dresden.de). This author was supported by Sonderforschungsbereich 609 funded by the Deutsche Forschungsgemeinschaft.

where the bounds $a < b$ are sufficiently regular. In fact, for our existence result (Theorem 3.1) we require $a, b \in L^2(Q)$. In section 6, however, for arguing locally superlinear convergence of our QP-solver we require $a, b \in L^q(Q)$ for some $q > 2$. In (1.1), $T > 0$ denotes the finite time horizon and ϖ the pressure. The space W corresponding to the state variable (velocity field) y and the choice of cost functionals J will be specified below. The variable u will be referred to as the control variable.

In the unconstrained case, i.e., $U_{\text{ad}} = U$, the development of numerical techniques for the optimal control of the stationary as well as instationary Navier–Stokes equations has received a considerable amount of attention. Here we refer to the monographs and selected papers [3, 6, 11, 15, 17, 18, 19, 22, 27, 32] covering distributed and boundary control problems. However, the presence of constraints on u requires a different algorithmic approach and usually complicates the numerical treatment significantly. If one wishes to apply a gradient related approach (like, e.g., in [17]), then the fact that $U_{\text{ad}} \subset U$ requires Hilbert space projections and modifications in potential line search techniques for globalization. Recently, SQP techniques have become feasible for solving the unconstrained version of (1.1); see, for instance, [19, 22, 27]. With respect to convergence speed the latter approach is typically superior to gradient methods. With regard to the computational complexity of SQP, the efficient solution of the linear-quadratic subproblems (QPs) is essential. In the presence of control constraints the QPs inherit the constraints from the problem formulation. As a consequence, compared to the unconstrained case more sophisticated QP-solvers have to be applied in order to compute appropriate search directions.

Motivated by the fast local convergence properties of SQP-techniques, we will extend the unconstrained approach of, e.g., [22, 27] to the control constrained case. The numerical QP-framework in this paper is related to the primal-dual active set strategy (pdAS) as introduced in [4] and further analyzed and tested in [5, 20, 21]. In particular, the latter results prove this method to be extremely efficient in the case of control constraints. Recently, in [23] it was shown that the pdAS is a particular instance of a generalized Newton method for a class of optimization problems in function spaces. In the present context, we utilize an inexact version of pdAS which is due to the large size of the problem and in order to save computation time. This is in the spirit of inexact Newton techniques for smooth problems; see, e.g., [10, 16].

As a key result it will turn out that pdAS provides a framework which efficiently deals with the constraints of the type (1.2) and requires only a moderate number of modifications in a SQP-environment for unconstrained problems in order to include the constrained case, too. This is of particular advantage since it allows one to extend one's favorite SQP-solver quite easily. The alterations needed for including constraints essentially comprise the storage of index sets referring to whether u is equal to one of the bounds and changes in the conjugate gradient (CG) method for solving the linear systems arising in pdAS when solving the QP-problems. Moreover, if the initial control u^0 is feasible, i.e., $u^0 \in U_{\text{ad}}$, and exact QP-solutions are computed, then the algorithm produces only feasible iterates with respect to the control variable.

Besides the numerical justification of our approach by a report on an excerpt of extensive numerical tests, an infinite dimensional convergence analysis for the SQP-algorithm is provided. By utilizing the concept of slant differentiability of a not necessarily (Fréchet) differentiable mapping between Banach spaces [7, 23] (see also [37] for a related notion), the locally superlinear convergence of the primal-dual active set algorithm (inner iteration) is established. For the analysis of the SQP-method (outer iteration) a generalized equations approach is utilized. In a different context, the potential of generalized equations for the analysis of algorithms for constrained optimal control problems was exploited previously in, e.g., [1, 12, 34] and the references therein. Under a strong regularity property (see [31]) we prove that the SQP-iteration converges at a locally quadratic rate.

To the best of our knowledge, the only contribution dealing with control constrained optimal control of the instationary Navier–Stokes equations in a generalized Newton framework is given by [36]. In this paper, a Newton algorithm for computing a solution to the first order optimality conditions of the reduced problem

$$\text{minimize} \quad \hat{J}(u) = J(y(u), u) \quad \text{subject to } u \in U_{\mathrm{ad}} \tag{1.3}$$

is considered. Here $y(u)$ denotes the solution of the Navier–Stokes equations for given u. In [36] a locally superlinear rate of convergence of the generalized Newton method is established. Note the difference from our approach. First, we use a SQP-framework which requires a different convergence analysis yielding convergence at a locally quadratic rate. We utilize the pdAS, or equivalently a generalized Newton method, as the QP-solver and prove its locally superlinear convergence. In our tests, typically the pdAS terminates after one or two iterations with a solution fulfilling a stopping rule similar to inexact Newton methods. The stopping tolerance is tuned in such a way that fast progress of the outer iteration is maintained. Further, our generalized Newton method does not require a separate smoothing step for its analysis. Indeed we take advantage of inherent smoothing properties due to the structure of the control problem; see the proof of Proposition 5.4. On the numerical level, we use a different discretization concept which provides discrete, numerically computable controls without explicitly discretizing the controls. The controls are discretized implicitly through the optimality conditions in terms of the adjoint variables. For more details on this technique see [26].

The rest of the paper is organized as follows. In the next section we introduce the precise functional analytic setting of (1.1) and notation. Further we establish some basic results required for the convergence analysis. Section 3 is devoted to the first order optimality system for the underlying control problem. Due to the constraints the first order system involves a so-called complementarity system consisting of primal and dual inequality conditions and a nonlinear equality. Relying on the concept of complementarity functions, we reformulate the latter system as a set of nonsmooth equalities. In section 4 we introduce the reduced first order system and the SQP-algorithm. Due to the equality related to the complementarity function which is nondifferentiable, an active set type QP-solver is the focus of section 5. The section also includes the convergence analysis for the QP-solver. In section 6 local quadratic convergence of the SQP-iteration is established. In section 7 we provide a report on numerical results obtained by our SQP-method.

2. Preliminaries. For the convenience of the reader we collect the analytical preliminaries for a proper formulation of problem (1.1) and to establish convergence results for the algorithms presented in this paper. To define the spaces and operators required for the investigation of (1.1) we introduce the solenoidal spaces

$$H = \overline{\{v \in C_0^\infty(\Omega)^2 \colon \ \mathrm{div}\, v = 0\}}^{|\cdot|_{L^2}}, V = \overline{\{v \in C_0^\infty(\Omega)^2 \colon \ \mathrm{div}\, v = 0\}}^{|\cdot|_{H^1}}$$

with the superscripts denoting closures in the respective norms. We have (see [32])

$$V \hookrightarrow H = H^* \hookrightarrow V^*$$

with $\hookrightarrow$ denoting the continuous injection. The spaces H^* and V^* denote the dual spaces of H and V, respectively. Further, we define the Banach spaces

$$W_q^p = \{v \in L^p(V) \colon v_t \in L^q(V^*)\} \quad \text{and} \quad Z := L^2(V) \times H, \tag{2.1}$$

where W_q^p is endowed with the norm

$$|v|_{W_q^p} = |v|_{L^p(V)} + |v_t|_{L^q(V^*)},$$

abbreviate

$$W := W_2^2, \tag{2.2}$$

and set $\langle \cdot, \cdot \rangle := \langle \cdot, \cdot \rangle_{L^2(V^*), L^2(V)}$. Here $L^2(V)$ is an abbreviation for $L^2(0,T;V)$ and similarly $L^2(V^*) = L^2(0,T;V^*)$. Note that we also have $L^2(V) \hookrightarrow L^2(Q)^2 \hookrightarrow L^2(V^*)$. We further need the following embedding result.

Proposition 2.1. *Let $\epsilon \in (0,1)$. For all $1 \le p < \frac{3}{2} + \frac{1+\epsilon}{1-\epsilon} =: \delta_\epsilon$ there holds*

$$W_{1+\epsilon}^2 \hookrightarrow L^p(Q)^2.$$

Proof. In [36, Lemma A1] the result is proved for $\epsilon = \frac{1}{3}$. It follows from [2, Theorem 1.1] that $W_{1+\epsilon}^2 \hookrightarrow L^q(H)$ for all $q \in [1, \frac{4(1+\epsilon)}{1-\epsilon})$. The remainder of the proof follows the lines of the proof of Lemma A1 in [36]. □

We note that up to a set of measure zero in $(0,T)$, elements $v \in W$ can be identified with elements in $C([0,T];H)$. In our convergence analysis we also need

$$H^{2,1}(Q) = \big\{v \in L^2\big(0,T;H^2(\Omega)\big) : v_t \in L^2(H)\big\}$$

endowed with the norm

$$|v|_{H^{2,1}(Q)} := |v|^2_{L^2(V \cap H^2(\Omega)^2)} + |v_t|^2_{L^2(H)}.$$

In [33] (compare [30]) it is shown that for $\Omega \subset \mathbb{R}^2$

$$H^{2,1}(Q) \hookrightarrow L^\infty\big(0,T;H^1(\Omega)\big) \cap L^q(Q) \quad \text{for} \quad 1 \le q < +\infty. \tag{2.3}$$

In (1.2) U denotes the Hilbert space of controls which is identified with its dual U^*. Throughout we adopt the frequent choice $U = L^2(Q)^2$. Note that $w \in L^2(0,T;H)$ satisfies $w \in L^2(Q)^2$.

Concerning the class of cost functionals $J \colon W \times U \to \mathbb{R}$ considered herein, we invoke the following assumption.

Assumption 2.2.

- $J(y,u) = J_1(y) + J_2(u)$ is bounded from below, weakly lower semi-continuous, and twice Fréchet differentiable with locally Lipschitzian second derivative.
- $J_2(u) = \frac{\alpha}{2}\, |u|_U^2$ which implies that J is radially unbounded in u, i.e., $J(y,u) \to \infty$ as $|u|_U \to \infty$ for every $y \in W$.
- $J_{1_y}(y), J_{1_{yy}}(y)v \in L^{1+\epsilon}(V^*) \cap W^*$ for all $v \in W$ and for some $\epsilon \in (0,1)$.

Our assumptions on J are satisfied for the tracking type functional

$$J(y,u) = \frac{1}{2} \int_Q |y - z|^2 dx\, dt + \frac{\alpha}{2}\, |u|_U^2 \tag{2.4}$$

and functionals involving the vorticity of the fluid like

$$J(y,u) = \frac{1}{2}\int_Q |\nabla_x \times y(t,\cdot)|^2\, dx\, dt + \frac{\alpha}{2}\, |u|_U^2, \tag{2.5}$$

where $\alpha, \gamma > 0$ and $z \in W$ are given. These two functionals are even infinitely Fréchet differentiable on $W \times U$.

Associated with the governing equation in (1.1) we define the nonlinear mapping $e\colon W \times U \to Z^*$ by

$$e(y,u) = (\tfrac{\partial y}{\partial t} + (y\cdot\nabla)y - \nu\Delta y - u, y(0) - y_0),$$

where $y_0 \in H$. Note that we could also include a class of linear operators B acting on the control u on the right-hand side of the equation and in the first component in e. As long as B fulfills certain regularity requirements, like in, e.g., [36], it poses no difficulty. In variational form the constraints in (1.1) can be equivalently expressed as follows: given $u \in U_{\mathrm{ad}}$ find $y \in W$ such that $y(0) = y_0$ in H and

$$\langle y_t, v\rangle + \langle (y\cdot\nabla)y, v\rangle + \nu(\nabla y, \nabla v)_{L^2(L^2)} = \langle u, v\rangle \;\; \forall\, v \in L^2(V). \tag{2.6}$$

Utilizing e, the control problem (1.1) can be rewritten as

$$\min_{(y,u)\in W\times U_{\mathrm{ad}}} J(y,u) \;\text{ subject to }\; e(y,u) = 0 \;\text{ in } Z^*. \tag{2.7}$$

For the analysis of the SQP-method we shall frequently refer to the variational solution of the linearized Navier–Stokes system and the adjoint equations in the solenoidal setting. For this purpose we state the following proposition, which is proved in [22]; compare also [27] for a similar analytic framework. It is also essential for Newton and quasi-Newton methods.

PROPOSITION 2.3. *Let $y \in W$, $v_0 \in H$, and $g \in L^2(V^*)$. Then the system of linearized Navier–Stokes equations*

$$A(y)v = (g, v_0) \text{ in } Z^* \Leftrightarrow \begin{cases} v_t + (v\cdot\nabla)y + (y\cdot\nabla)v - \nu\Delta v = g & \text{in } L^2(V^*), \\ v(0) = v_0 \quad \text{in } H \end{cases} \tag{2.8}$$

admits a unique variational solution $v \in W$. For $y \in W \cap L^\infty(V) \cap L^2(H^2(\Omega)^2)$, $v_0 \in V$, and $g \in L^2(H)$ the unique solution v of (2.8) is an element of $H^{2,1}(Q)$ and satisfies the a priori estimate

$$|v|_{H^{2,1}(Q)} \le C(|y|_{L^\infty(V)}, |y|_{L^2(H^2(\Omega)^2)})\left\{|g|_{L^2(H)} + |v_0|_V\right\}.$$

Concerning the adjoint equation we state the following result.

PROPOSITION 2.4. *Let $y \in W$ and $f \in W^*$. Then the adjoint equation*

$$A(y)^* w = f \quad \text{in } W^*$$

admits a unique variational solution $w = (w^1, w^0) \in Z$. If $f \in L^q(V^*) \cap W^*$ $(1 \le q \le \infty)$, then for every $0 \le \epsilon \le \min\{q-1, \frac{1}{3}\}$ the function w^1 is an element of $W^2_{1+\epsilon}$ and the variational solution of

$$\begin{cases} -w^1_t + (\nabla y)^\top w^1 - (y\cdot\nabla)w^1 - \nu\Delta w^1 = f, \\ w^1(T) = 0, \end{cases} \tag{2.9}$$

and it satisfies $w^1(0) = w^0$. If in addition $y \in L^\infty(V)$, and $f \in L^2(V^*)$, then $w^1 \in W$. For $y \in W \cap L^\infty(V) \cap L^2(H^2(\Omega)^2)$, $v_0 \in V$, and $f \in L^2(H)$ the unique solution w^1 of (2.9) *is an element of $H^{2,1}(Q)$ and satisfies the a priori estimate*

$$|w^1|_{H^{2,1}(Q)} \le C(|y|_{L^\infty(V)}, |y|_{L^2(H^2(\Omega)^2)})\, |f|_{L^2(H)}.$$

Further properties of the linearized and adjoint equations can be found in [27].

For the application of the SQP-method to (1.1) we need second order information of the Lagrangian L, which is defined below in (3.1). The basic ingredients are the derivatives of the operator e which were characterized in [22]; compare also [27]. For the convenience of the reader we state the following proposition.

PROPOSITION 2.5. *The operator $e = (e^1, e^2)\colon W \times U \to Z^*$ is infinitely Fréchet differentiable with Lipschitz continuous first derivative, constant second derivative, and vanishing third and higher derivatives. The actions of the first two derivatives of e^1 are given by*

$$\begin{aligned}\langle e^1_x(x)(w,s),\phi\rangle = \langle w_t,\phi\rangle + \langle (w\cdot\nabla)y,\phi\rangle + \langle (y\cdot\nabla)w,\phi\rangle \\ + \nu(\nabla w,\nabla\phi)_{L^2(L^2)} - \langle s,\phi\rangle_{L^2(L^2)},\end{aligned}$$

where $x = (y,u) \in W \times U, (w,s) \in W \times U$, and $\phi \in L^2(V)$ and

$$\langle e^1_{xx}(x)(w,s)(v,r),\phi\rangle = \langle e^1_{yy}(x)(w,v),\phi\rangle \tag{2.10}$$

$$= \langle (w\cdot\nabla)v,\phi\rangle + \langle (v\cdot\nabla)w,\phi\rangle =: \langle v, M(\phi)w\rangle_{W,W^*}, \tag{2.11}$$

where $(v,r) \in W \times U$ and $M : L^2(V) \to \mathcal{L}(W,W^*)$.

Next we introduce the Lagrange function related to problem (2.7) with $U_{\mathrm{ad}} = U$, i.e., $\hat{L} : W \times U \times Z \to \mathbb{R}$ with

$$\hat{L}(y,u,p) = J(y,u) + \langle p, e(y,u)\rangle_{Z,Z^*}.$$

According to Proposition 2.5 we have

$$\begin{aligned}\langle \hat{L}_{yy}(y,u,p)v,w\rangle &= \langle J_{yy}(y,u)v,w\rangle + \langle e_{yy}(y,u)(v,w),p\rangle \\ &= \langle v,w\rangle + \langle (v\cdot\nabla)w,p\rangle + \langle (w\cdot\nabla)v,p\rangle\end{aligned}$$

with $v, w \in W$.

3. First order optimality and the QP-subproblem. The starting point for devising algorithms to find a local solution of (1.1) are the first order necessary conditions which will be derived in this section.

According to the results in the previous section we can write (1.1) in the compact form

$$\text{(P)}\qquad \begin{array}{ll} \text{minimize} & J(y,u) \quad \text{over } (y,u) \in W \times U \\ \text{subject to} & e(y,u) = 0 \quad \text{in } Z^*, \\ & a \le u \le b \quad \text{a.e. in } Q \end{array}$$

with $a, b \in L^2(Q)^2$.

The existence of a solution of (P) follows from standard arguments. For the sake of completeness we include the short proof.

THEOREM 3.1. *Problem* (P) *admits a (global) solution $(y^*, u^*) \in W \times U_{ad}$.*

Proof. Let $\{u_n\}_{n\in\mathbb{N}} \subset U_{\mathrm{ad}}$ be a minimizing sequence for problem (P). Due to the radial unboundedness of $J_2(u)$ this sequence is bounded and, thus, contains a weakly (in U) convergent subsequence, which we again denote by $\{u_n\}_{n\in\mathbb{N}}$. Since U_{ad} is convex it is weakly closed and the limit u^* of the subsequence is an element of U_{ad}. The a priori estimates stated above now ensure that the unique solutions y_n of $e(y_n, u_n) = 0$ in Z^* form a bounded sequence in W, which, in turn, contains a weakly (in W) convergent subsequence. Let $\tilde{y} \in W$ denote its limit. It follows from the analysis provided by Temam in [32] that $\tilde{y} = y(u^*)$. Since J is weakly lower semicontinuous the pair (y^*, u^*), in fact, is a solution of (P). $\square$

In the unconstrained case, SQP-methods can be derived by applying Newton's method to the first order optimality system; see [27]. Due to the presence of the constraints on u this approach has to be generalized. Associated with (P) we consider the Lagrange functional

$$L(x, p, \lambda) = J(x) + \langle e(x), p\rangle_{Z^*,Z} + (a - u, \lambda_a) + (u - b, \lambda_b), \tag{3.1}$$

where we used $\lambda = (\lambda_a, \lambda_b) \in U \times U$ and $x = (y, u)$. Here and throughout we denote by $(\cdot, \cdot)$ the $L^2(Q)^2$ inner product. Let us next state the first order necessary conditions of (P).

Theorem 3.2. *An optimal solution* $x^* = (y^*, u^*) \in W \times U_{ad}$ *to* (P) *is characterized by the existence of Lagrange multipliers* $p^* \in Z$ *and* $(\lambda_a^*, \lambda_b^*) = \lambda^* \in U^2$ *satisfying*

$$\text{(OS)} \qquad \begin{cases} J_y(x^*) + e_y^*(x)p^* = 0, \\ J_u(x^*) + e_u^*(x^*)p^* - \lambda_a^* + \lambda_b^* = 0, \\ e(x^*) = 0, \\ a - u^* \le 0, \quad \lambda_a^* \ge 0, \quad (a - u^*, \lambda_a^*) = 0, \\ u^* - b \le 0, \quad \lambda_b^* \ge 0, \quad (u^* - b, \lambda_b^*) = 0. \end{cases}$$

The last two equation in (OS) form the so-called complementarity system. Note that in the case where $U_{\mathrm{ad}} = U$ only the first three equations in (OS) with $\lambda^* = 0$ have to be taken into account.

Given a point (x, p, λ) close to a locally optimal solution (x^*, p^*, λ^*) let us now apply a generalized Newton step to the system

$$\begin{cases} J_y(x) + e_y^*(x)p = 0, \\ J_u(x) + e_u^*(x)p - \lambda_a + \lambda_b = 0, \\ e(x) = 0, \\ a - u \le 0, \quad \lambda_a \ge 0, \quad (a - u, \lambda_a) = 0, \\ u - b \le 0, \quad \lambda_b \ge 0, \quad (u - b, \lambda_b) = 0. \end{cases} \tag{3.2}$$

To unburden the notation, subsequently we will neglect the argument (x, p, λ). The generalized Newton step $(\delta_y, \ldots, \delta_{\lambda_b})$ satisfies

$$\begin{pmatrix} L_{yy} & 0 & e_y^* & 0 & 0 \\ 0 & L_{uu} & e_u^* & -\,\mathrm{id} & \mathrm{id} \\ e_y & e_u & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta_y \\ \delta_u \\ \delta_p \\ \delta_{\lambda_a} \\ \delta_{\lambda_b} \end{pmatrix} = - \begin{pmatrix} L_y \\ L_u \\ e \end{pmatrix} \tag{3.3}$$

for the first three equations in (OS). For the complementarity system we define

$$\hat{\lambda}_a = \lambda_a + \delta_{\lambda_a}, \quad \hat{\lambda}_b = \lambda_b + \delta_{\lambda_b}.$$

Thus, we obtain

$$a - u - \delta_u \leq 0, \quad \hat{\lambda}_a \geq 0, \quad (a - u - \delta_u, \hat{\lambda}_a) = 0, \tag{3.4a}$$

$$u + \delta_u - b \leq 0, \quad \hat{\lambda}_b \geq 0, \quad (u + \delta_u - b, \hat{\lambda}_b) = 0. \tag{3.4b}$$

Note that in the last equations in (3.4a), respectively, (3.4b), we keep the quadratic terms $-(\delta_u, \delta_{\lambda_a})$, respectively, $(\delta_u, \delta_{\lambda_b})$. This allows us to establish a link between (3.3)–(3.4) and the constrained minimization problem (QP) introduced below. Observe that the second equation in (3.3) implies

$$L_{uu}\delta_u + e_u^*\delta_p + \hat{\lambda} = -(J_u + e_u^* p), \tag{3.5}$$

where we use

$$\hat{\lambda} = -\hat{\lambda}_a + \hat{\lambda}_b.$$

Further note that the system (3.4) can be rewritten as a nonsmooth equation of the form

$$\max(\hat{\lambda} + \sigma(u + \delta_u - b), 0) + \min(\hat{\lambda} + \sigma(u + \delta_u - a), 0) = \hat{\lambda} \tag{3.6}$$

for arbitrarily fixed real $\sigma > 0$. In fact, it is an easy exercise to show that (3.4) and (3.6) are equivalent.

These considerations finally result in the following system which has to be solved in order to obtain the Newton direction $(\delta_y, \delta_u, \delta_p)$ with associated Lagrange multiplier $\hat{\lambda}$:

$$\begin{cases} L_{yy}\delta_y + e_y^*\delta_p = -L_y, \\ L_{uu}\delta_u + e_u^*\delta_p + \hat{\lambda} = -(J_u + e_u^* p), \\ e_y\delta_y + e_u\delta_u = -e, \\ \Psi(\delta_u, \hat{\lambda}; u) = \hat{\lambda} \end{cases} \tag{3.7}$$

with

$$\Psi(\delta_u, \hat{\lambda}; u) := \max(\hat{\lambda} + \sigma(u + \delta_u - b), 0) + \min(\hat{\lambda} + \sigma(u + \delta_u - a), 0). \tag{3.8}$$

In the case where the operator matrix

$$L_{xx} := \begin{pmatrix} L_{yy} & 0 \\ 0 & L_{uu} \end{pmatrix}$$

is positive semidefinite, i.e., it satisfies $\langle L_{xx}\delta_x, \delta_x\rangle \geq 0$ for all $\delta_x = (\delta_y, \delta_u) \in W \times U$, the system (3.7) represents the first order necessary and sufficient condition of the QP-problem

$$\text{(QP)} \qquad \begin{array}{ll} \text{minimize} & \langle \hat{L}_x, \delta_x\rangle + \frac{1}{2}\langle \hat{L}_{xx}\delta_x, \delta_x\rangle \quad \text{over } \delta_x \in W \times U \\ \text{subject to} & e + e_x\delta_x = 0 \quad \text{in } Z^*, \\ & a - u \leq \delta_u \leq b - u \quad \text{a.e. in } Q. \end{array}$$

Here we used

$$\hat{L}(x, p) = J(x) + \langle e(x), p\rangle_{Z^*,Z},$$

the Lagrangian for the unconstrained version of (P). Note that due to the affine linear nature of the inequality constraints we have

$$\hat{L}_{xx}(x,p) = L_{xx}(x,p,\lambda)$$

and further

$$\hat{L}_y(x,p) = L_y(x,p,\lambda).$$

Hence, compared to the unconstrained control problem as considered, e.g., in [22] the objective functional of the QP-problems remains the same, and only the additional constraints on δ_u must be realized. However, this requires a more sophisticated QP-solver, which is the subject of section 5.

4. Reduced system and the SQP-algorithm. Solving, in every iteration of a numerical algorithm, a time-dependent (sub)problem of the type (3.7) is a formidable task due to the size of the problem. Our aim is now to derive a reduced version of (3.7) which is more tractable numerically. For this purpose observe that the third equation in (3.7) yields

$$\delta_y = -e_y^{-1}(x)(e(x) + e_u(x)\delta_u). \tag{4.1}$$

Utilizing (4.1) in the first equation of (3.7) results in

$$\begin{aligned} \delta_p &= -e_y^{-*}(x)(\hat{L}_y(x,p) + \hat{L}_{yy}(x,p)\delta_y) \\ &= -e_y^{-*}(x)\Big(\hat{L}_y(x,p) - \hat{L}_{yy}(x,p)\big(e_y^{-1}(x)(e(x)+e_u(x)\delta_u)\big)\Big), \end{aligned} \tag{4.2}$$

where e_y^{-*} denotes the adjoint of the inverse e_y^{-1}. If we insert (4.1)–(4.2) in the second equation of (3.7), then we obtain (again after neglecting the argument (x,p,λ))

$$(L_{uu} + e_u^* e_y^{-*}\hat{L}_{yy}e_y^{-1}e_u)\delta_u + \hat{\lambda} = -\hat{L}_u + e_u^* e_y^{-*}(\hat{L}_y - \hat{L}_{yy}e_y^{-1}e). \tag{4.3}$$

Taking into account that

$$-\hat{L}_u + e_u^* e_y^{-*}\hat{L}_y = -J_u + e_u^* e_y^{-*}J_y,$$

then (4.3) simplifies to

$$T^*\hat{L}_{xx}T\delta_u + \hat{\lambda} = -T^*J_x - e_u^* e_y^{-*}\hat{L}_{yy}e_y^{-1}e \tag{4.4}$$

with

$$T = \begin{pmatrix} -e_y^{-1}e_u \\ \mathrm{id} \end{pmatrix} \quad \text{and} \quad \hat{L}_{xx} = \begin{pmatrix} \hat{L}_{yy} & 0 \\ 0 & \hat{L}_{uu} \end{pmatrix} = \begin{pmatrix} L_{yy} & 0 \\ 0 & L_{uu} \end{pmatrix}.$$

Summarizing our computations, it turns out that the solution $(\delta_x, \delta_p, \hat{\lambda})$ of (3.7) can be computed by solving the *reduced system*

$$\begin{cases} T^*\hat{L}_{xx}T\delta_u + \hat{\lambda} = -T^*J_x - e_u^* e_y^{-*}\hat{L}_{yy}e_y^{-1}e, \\ \Psi(\delta_u, \hat{\lambda}; u) = \hat{\lambda} \end{cases} \tag{OR}$$

for $(\delta_u, \hat{\lambda})$ and then performing efficient backward substitution in (4.1) and (4.2).

Let $r = r(x,p)$ denote the negative right-hand side in (4.4), i.e.,

$$r = T^* J_x + e_u^* e_y^{-*} \hat{L}_{yy} e_y^{-1} e.$$

Assuming that

$$H = T^* \hat{L}_{xx} T$$

is positive semidefinite, then the reduced system (OR) represents the first order necessary and sufficient condition for the *reduced QP-problem*

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\langle T^* \hat{L}_{xx} T \delta_u, \delta_u \rangle + \langle r, \delta_u \rangle \\ &\text{subject to} && a - u \le \delta_u \le b - u \quad \text{a.e. in } Q. \end{aligned} \tag{R}$$

The basic SQP-algorithm will be specified next. Subsequently (OR^n) refers to problem (OR) with $(x,p) = (x^n, p^n)$, $n \in \mathbb{N}_0$.

Algorithm 4.1. SQP-framework.

```
1. Choose $x^0 = (y^0, u^0) \in W \times U_{ad}, p^0 \in Z, \lambda^0 \in U^2$, sufficiently close to a local
   solution; set $n = 0$.
2. Do until convergence
   (a) Compute the solution $(\delta_u^n, \hat{\lambda}^n)$ of $(OR^n)$.
   (b) Compute $\delta_y^n, \delta_p^n$ from (4.1)–(4.2) at $(x^n, p^n)$.
   (c) Update $x^n = x^{n-1} + \delta_x^n$ and $p^n = p^{n-1} + \delta_p^n$. Set $n = n + 1$.
```

Note that this SQP-algorithm requires globalization in order to allow an arbitrary initial choice. A globalization with respect to the requirements for unconstrained optimal control of the Navier–Stokes equations can be found in [22]. This strategy may also be applied in the present context. In fact, due to the affine character of the inequality constraints, the Hessian of the constrained and the unconstrained problems coincide. Also the line search technique in [22] remains valid as long as the control iterates u^n remain feasible during the iteration. The key ingredient of this globalization strategy is to check positive definiteness properties of

$$H^n := T^*(x^n) \hat{L}_{xx}(x^n, p^n) T(x^n)$$

for all n. If H^n is not positive definite in the direction δ_u^n, then a positive definite approximation $\tilde{H}^n$ of H^n is computed and the QP-subproblem with H^n replaced by $\tilde{H}^n$ is solved yielding a new δ_u^n. Here, positive definiteness of, e.g., $\tilde{H}^n$ refers to the existence of some $\epsilon > 0$ such that $\langle \tilde{H}^n \delta_u, \delta_u \rangle \ge \epsilon |\delta_u|_U^2$. It can be shown that H^n is positive definite sufficiently close to a local solution which satisfies the strong second order sufficient conditions; see [27]. This implies that we eventually have $\tilde{H}^n = H^n$. Consequently, resorting to local arguments we may assume throughout that H^n is positive definite for all n.

5. An efficient QP-solver.

As noticed earlier, the computation of a solution to (OR^n) requires a more sophisticated solver compared to the unconstrained case. This is due to the nonsmooth equation $\Psi(\delta_u, \hat{\lambda}; u) = \hat{\lambda}$. Here we adopt the primal-dual active set strategy (pdAS) in order to solve (OR^n).

Recall that we use $\hat{\lambda} = \lambda_b - \lambda_a$. The key step of pdAS consists in estimating the a-active, b-active, and inactive sets at the solution $(y^*, u^*, p^*, \hat{\lambda}^*)$ given by

$$\mathcal{A}_*^a := (\mathcal{A}_{*,1}^a, \mathcal{A}_{*,2}^a)^\top, \quad \mathcal{A}_{*,i}^a := \{u_i^* = a_i\}, \quad i = 1, 2, \tag{5.1a}$$

$$\mathcal{A}_*^b := (\mathcal{A}_{*,1}^b, \mathcal{A}_{*,2}^b)^\top, \quad \mathcal{A}_{*,i}^b := \{u_i^* = b_i\}, \quad i = 1, 2, \tag{5.1b}$$

$$\mathcal{I}_* := Q^2 \setminus (\mathcal{A}_*^a \cup \mathcal{A}_*^b) \qquad \text{(componentwise union)}, \tag{5.1c}$$

respectively. Note that from our first order characterization, Theorem 3.2, we have (in the almost everywhere sense)

$$\hat{\lambda}^*_{|\mathcal{A}^a_*} \leq 0, \quad \hat{\lambda}^*_{|\mathcal{A}^b_*} \geq 0, \quad \hat{\lambda}^*_{|\mathcal{I}_*} = 0. \tag{5.2}$$

Now assume that we are given u^n and we want to determine the solution δ^n_u, $\hat{\lambda}^n$ to (OR^n) by the pdAS. In view of (5.1) and (5.2), the aim is to compute $(\delta^n_u, \hat{\lambda}^n)$ such that $u^n + \delta^n_u$ and $\hat{\lambda}^n$ satisfy

$$a \leq u^n + \delta^n_u \leq b \quad \text{and} \quad \hat{\lambda}^n_{|\mathcal{A}^a_n} \leq 0, \quad \hat{\lambda}^n_{|\mathcal{A}^b_n} \geq 0, \quad \hat{\lambda}^n_{|\mathcal{I}_n} = 0$$

simultaneously. Given estimates $\delta_{u,l-1}$ and $\hat{\lambda}_{l-1}$ of δ^n_u and $\hat{\lambda}^n$, we define the following approximations of the active and inactive sets in (5.1) by $(i = 1, 2)$:

$$\mathcal{A}^a_l := (\mathcal{A}^a_{l,1}, \mathcal{A}^a_{l,2})^\top, \quad \mathcal{A}^a_{l,i} := \left\{ \left(\hat{\lambda}_{l-1} + \sigma(u^n + \delta_{u,l-1} - a) \right)_i < 0 \right\}, \tag{5.3a}$$

$$\mathcal{A}^b_l := (\mathcal{A}^b_{l,1}, \mathcal{A}^b_{l,2})^\top, \quad \mathcal{A}^b_{l,i} := \left\{ \left(\hat{\lambda}_{l-1} + \sigma(u^n + \delta_{u,l-1} - b) \right)_i > 0 \right\}, \tag{5.3b}$$

$$\mathcal{I}_l := Q^2 \setminus (\mathcal{A}^a_l \cup \mathcal{A}^b_l) \qquad \text{(componentwise union)}. \tag{5.3c}$$

Above the scalar $\sigma > 0$ is arbitrarily fixed. In section 5.1 we will see that $\sigma = \alpha > 0$ is of particular interest. The choice (5.3) is related to (3.6). In a discussion following Proposition 5.4 a detailed motivation is given. As it will turn out, it is associated to a generalized derivative of Ψ.

Let us next specify the pdAS as utilized in step 2(a) of the SQP-algorithm. For convenience we use $r^n = r(x^n, p^n)$.

Algorithm 5.1. Primal-dual active set strategy.

2(a.0) *Initialize* $\delta_{u,0} = 0$, $\hat{\lambda}_0 = -r^n$; *set* $l = 1$. *Choose a small* $\epsilon > 0$.

2(a.1) *Determine* $\mathcal{A}^a_l$, $\mathcal{A}^b_l$ *and* $\mathcal{I}_l$ *from* (5.3).

2(a.2) *If* $l \geq 2$ *and* $\mathcal{A}^a_l = \mathcal{A}^a_{l-1}$, $\mathcal{A}^b_l = \mathcal{A}^b_{l-1}$, *or*

$$|\Psi(\delta_{u,l-1}, \hat{\lambda}_{l-1}; u^n) - \hat{\lambda}_{l-1}|_{(L^2)^2} \leq \epsilon,$$

then $\delta^n_u = \delta_{u,l-1}$, $\hat{\lambda}^n = \hat{\lambda}_{l-1}$ *and RETURN (to Algorithm* 4.1*); otherwise go to step* 2(a.3)*.*

2(a.3) *Fix*

$$\begin{aligned} \delta_{u,l} &= a - u^n \quad \text{on } \mathcal{A}^a_l, \\ \delta_{u,l} &= b - u^n \quad \text{on } \mathcal{A}^b_l, \\ \hat{\lambda}_l &= 0 \quad \text{on } \mathcal{I}_l \end{aligned}$$

and obtain $\delta_{u,l}|_{\mathcal{I}_l}$, $\hat{\lambda}_l|_{\mathcal{A}^a_l \cup \mathcal{A}^b_l}$ *from solving*

$$H^n \delta_{u,l} + \hat{\lambda}_l = -r^n. \tag{5.4}$$

Put $l := l + 1$ *and go to* 2(a.1)*.*

The reduced problem (5.4), which has to be solved in every iteration of the pdAS, can hardly be solved by means of direct solvers due to the size of the problem. Since H^n is positive definite and symmetric, we apply the CG-method. In the context of pdAS the CG-method has to take care of the particular settings according to step 2(a.3) of pdAS, i.e.,

$$\delta_{u,l|\mathcal{A}_l^a} = (a - u^n)_{|\mathcal{A}_l^a}, \quad \delta_{u,l|\mathcal{A}_l^b} = (b - u^n)_{|\mathcal{A}_l^b}, \quad \text{and} \quad \hat{\lambda}_{l|\mathcal{I}_l} = 0. \tag{5.5}$$

Hence, the CG-method operates essentially only on $\mathcal{I}_l$. Consequently, we consider the following subspace CG-method, where $\mathcal{A}_l = \mathcal{A}_l^a \cup \mathcal{A}_l^b$.

ALGORITHM 5.2. SUBSPACE CG-METHOD.

2(a.3.0) *Initialize*

$$v_{0|\mathcal{I}_l} := 0, \quad v_{0|\mathcal{A}_l^a} = (a - u^n)_{|\mathcal{A}_l^a}, \quad v_{0|\mathcal{A}_l^b} = (b - u^n)_{|\mathcal{A}_l^b},$$
$$d_{0|\mathcal{I}_l} = r^n_{|\mathcal{I}_l} - (H^n v_0)_{|\mathcal{I}_l} =: g_{0|\mathcal{I}_l}, \quad k := 0.$$

2(a.3.1) *Do until convergence*

```
(a) $t_k := \frac{|g_{k|\mathcal{I}_l}|^2_{(L^2)^2}}{(d_k, H^n d_k)}$,
(b) $v_{k+1} = v_k + t_k d_k$,
(c) $g_{k+1|\mathcal{I}_l} = g_{k|\mathcal{I}_l} + t_k (H^n d_k)_{|\mathcal{I}_l}$,
(d) $\beta_k = \frac{|g_{k+1|\mathcal{I}_l}|^2_{(L^2)^2}}{|g_{k|\mathcal{I}_l}|^2_{(L^2)^2}}$,
(e) $d_{k+1|\mathcal{I}_l} = -g_{k+1|\mathcal{I}_l} + \beta_k d_{k|\mathcal{I}_l}$,
(f) $d_{k+1|\mathcal{A}_l} = 0$,
(g) $k = k + 1$.
```

2(a.3.2) *Set* $\delta_{u,l} = v_k$ *and compute* $\hat{\lambda}_l = -r^n - H^n \delta_{u,l}$.

Note that we do not require a partitioning of H^n according to active and inactive sets. Rather we achieve the settings (5.5) by fixing $d_{k+1|\mathcal{A}_l} = 0$ in step 2(a.3.1)(f). Finally, we remark that the stopping tolerance $\epsilon > 0$ of the pdAS and the criterion for terminating the subspace CG-method have to be adjusted appropriately; see our choices in section 7.

5.1. Convergence properties of the primal-dual active set strategy.

Let us now turn toward the convergence analysis of the primal-dual active set strategy. For this purpose we recall the concept of slant differentiability of a function as introduced in [7]. In [23] this concept is utilized for proving locally superlinear convergence of the primal dual active set strategy for a class of constrained optimization problems in function spaces. This convergence result relies on the fact that the primal-dual active set strategy is equivalent to a semismooth Newton method.

Let $F : D \subset X \to Y$ be a mapping from an open subset D of the Banach space X with values in the Banach space Y. The following definition is taken from [7] (see also [23]).

DEFINITION 5.3. *The mapping* $F : D \subset X \to Y$ *is called* slantly differentiable *in* D *if there exists a family of mappings* $G \in \mathcal{L}(X, Y)$ *satisfying*

$$\lim_{h \to 0} \frac{\|F(x+h) - F(x) - G(x+h)h\|_Y}{\|h\|_X} = 0 \quad \text{for } x \in D.$$

The mapping G *is called* slanting function *for* F *in* D.

In [23] (see also [37]) it is observed that max : $L^{q_1}(\Omega) \to L^{q_2}(\Omega)$ is slantly differentiable for $1 \le q_2 < q_1 \le +\infty$. If $q_1 \le q_2$ this property does not hold true. Note that in our problem setting we have $u, \delta_u, \hat{\lambda} \in L^2(Q)^2$. Hence, the slant differentiability concept with respect to δ_u or $\hat{\lambda}$ cannot be applied to $\Psi(\delta_u, \hat{\lambda}; u)$ immediately.

In our numerical tests we use the following modified version of the primal dual active set algorithm which operates on δ_u only. For its derivation consider equation (4.3), which gives a relation between $\hat{\lambda}$ and δ_u, i.e.,

$$\hat{\lambda} = -\hat{L}_u + e_u^* e_y^{-*}\big(\hat{L}_y - \hat{L}_{yy} e_y^{-1}(e + e_u \delta_u)\big) - L_{uu}\delta_u. \tag{5.6}$$

Note that the cost functionals in (2.4) and (2.5) yield

$$J_u(u) = \alpha u \quad \text{and} \quad L_{uu}(x, p, \lambda)\delta_u = \alpha \delta_u.$$

Thus, from (5.6) it follows

$$\hat{\lambda} = -\alpha(u + \delta_u) + S\delta_u + t \tag{5.7}$$

with

$$\begin{aligned} t = t(y, u, p) &= e_u^* e_y^{-*}(J_y - \hat{L}_{yy} e_y^{-1} e), \\ S = S(y, u, p) &= -e_u^* e_y^{-*} \hat{L}_{yy} e_y^{-1} e_u. \end{aligned}$$

Using (5.7) for $\hat{\lambda}$ in $\Psi(\delta_u, \hat{\lambda}; u)$ with the particular choice $\sigma = \alpha$ yields

$$\hat{\Psi}(\delta_u) := \max(S\delta_u - \alpha b + t, 0) + \min(S\delta_u - \alpha a + t, 0). \tag{5.8}$$

Relation (5.8) allows us to replace the componentwise active set estimates in (5.3) by

$$\mathcal{A}_{l,i}^a := \left\{ (S^n \delta_{u,l-1} - \alpha a + t^n)_i < 0 \right\}, \quad i = 1, 2, \tag{5.9a}$$

$$\mathcal{A}_{l,i}^b := \left\{ (S^n \delta_{u,l-1} - \alpha b + t^n)_i > 0 \right\}, \quad i = 1, 2. \tag{5.9b}$$

Here we use $S^n = S(y^n, u^n, p^n)$ and analogously for t^n. We call the resulting algorithm, which iterates on $\delta_{u,l}$ only, the *reduced primal-dual active set strategy* (rpdAS).

As a candidate for a slanting function for $\hat{\Psi}$ we consider

$$G(\delta_u) = g_{\max}(S\delta_u - \alpha b + t)S + g_{\min}(S\delta_u - \alpha a + t)S \tag{5.10}$$

with

$$g_{\max}(w)(x) = \begin{cases} 1 & \text{if } w(x) > 0, \\ 0 & \text{else,} \end{cases}$$

and

$$g_{\min}(w)(x) = \begin{cases} 1 & \text{if } w(x) < 0, \\ 0 & \text{else.} \end{cases}$$

Let us motivate this particular choice of G with respect to the pdAS, respectively, rpdAS. Assume for the moment that G is a slanting function for $\hat{\Psi}$. We apply a generalized version of Newton's method for iteratively solving

$$H^n \delta_u + \hat{\lambda} + r^n = 0, \tag{5.11a}$$

$$\hat{\Psi}(\delta_u) + \alpha(u + \delta_u) - S\delta_u - t = 0. \tag{5.11b}$$

Suppose that $\delta_{u,l-1}, \hat{\lambda}_{l-1}$ are the actual iterates in the Newton process. Then, using $G(\delta_{u,l-1})$, we obtain the following equations for the increments $d_u, d_{\hat{\lambda}}$:

$$H^n d_u + d_{\hat{\lambda}} = -r^n - H^n \delta_{u,l-1} - \hat{\lambda}_{l-1}, \tag{5.12a}$$

$$\chi_{\mathcal{A}_l^b} S d_u + \chi_{\mathcal{A}_l^a} S d_u + \alpha d_u - S d_u = -\hat{\Psi}(\delta_{u,l-1}) - \alpha(u + \delta_u) + S\delta_u + t. \tag{5.12b}$$

By $\chi_{\mathcal{S}}$ we denote the characteristic function of a set $\mathcal{S} \subset Q^2$. On $\mathcal{A}_l^b$ equation (5.12b) yields

$$\delta_{u,l-1} + d_u = b - u.$$

Analogously, on $\mathcal{A}_l^a$ we obtain

$$\delta_{u,l-1} + d_u = a - u.$$

Thus rpdAS is regained. As a consequence, if G in (5.10) is a slanting function for $\hat{\Psi}$, then rpdAS is equivalent to a generalized Newton method for the nondifferentiable system (5.11).

With respect to the desired slant differentiability relation we have the following result.

PROPOSITION 5.4. *Let Assumption* 2.2 *be satisfied with* $\epsilon \in (0, \frac{1}{3}]$ *and let* δ_ϵ *as in Proposition* 2.1. *Further let* $q \in (2, \delta_\epsilon)$ *be arbitrarily fixed, and* $a, b \in L^q(Q)^2$. *Then the mapping* G *defined in* (5.10) *is a slanting function for* $\hat{\Psi} : L^2(Q)^2 \to L^2(Q)^2$ *with* $t \in L^q(Q)^2$, $y \in W$, $p \in Z$, *and* $S : L^2(Q)^2 \to W^2_{1+\epsilon} \hookrightarrow L^q(Q)^2$.

Proof. First observe that $e_u = (-\operatorname{id}, 0)$ which, by Proposition 2.3, yields $e_y^{-1} e_u \delta_u \in W$ for $\delta_u \in L^2(Q)^2$. Next consider $w = \hat{L}_{yy} v$ with $v = e_y^{-1} e_u \delta_u$ in detail. Since $p \in Z$ and $y \in W$, a straightforward estimation gives $\langle e_{yy}(y,u)(\cdot, v), p\rangle_{Z^*,Z} \in L^{\frac{4}{3}}(V^*) \cap W^*$. Since by Assumption 2.2 $J_{1_{yy}}(y)v \in L^{1+\epsilon}(V^*) \cap W^*$ we obtain $\hat{L}_{yy} v \in L^{1+\epsilon}(V^*) \cap W^*$. The regularity results of Proposition 2.4 now yield

$$z = e_y^{-*} w \in W^2_{1+\epsilon},$$

where by Proposition 2.1 the space $W^2_{1+\epsilon}$ continuously embeds into $L^p(Q)^2$ for all $2 < p < \delta_\epsilon$. Since $e_u = (-\operatorname{id}, 0)$, this immediately yields $S\delta_u \in L^q(Q)^2$. A similar argument proves that $e_y^{-*} J_y$ and $e_y^{-*} \hat{L}_{yy} e_y^{-1} e$ are elements of $W^2_{1+\epsilon}$, respectively, and hence $t \in L^q(Q)^2$.

For the remainder of the assertion we restrict ourselves to proving that $\hat{\psi} : L^2(Q)^2 \to L^2(Q)^2$ with

$$\hat{\psi}(\delta_u) = \max(S\delta_u - \alpha b + t, 0)$$

is slantly differentiable with slanting function

$$G_{\max}(\delta_u) = g_{\max}(S\delta_u - \alpha b + t, 0) S.$$

Applying the analogous arguments to the min-term in $\hat{\Psi}$ then proves slant differentiability of $\hat{\Psi}$ with the slanting function G as defined in (5.10).

For $h \in L^2(Q)^2$ and $2 < q < \delta_\epsilon$ consider

$$\begin{aligned} &\lim_{h \to 0} \frac{|\hat{\psi}(\delta_u + h) - \hat{\psi}(\delta_u) - G_{\max}(\delta_u + h)h|_{(L^2)^2}}{|h|_{(L^2)^2}} \\ &= \lim_{h \to 0} \frac{|Sh|_{(L^q)^2}}{|h|_{(L^2)^2}} \, \frac{|\hat{\psi}(\delta_u + h) - \hat{\psi}(\delta_u) - G_{\max}(\delta_u + h)h|_{(L^2)^2}}{|Sh|_{(L^q)^2}}. \end{aligned}$$

Since $S : L^2(Q)^2 \to L^q(Q)^2$ is bounded we may proceed as in the proof of [23, Theorem 4.1] to argue

$$\begin{aligned}&\lim_{h\to 0} \frac{|Sh|_{(L^q)^2}}{|h|_{(L^2)^2}} \frac{|\hat{\psi}(\delta_u + h) - \hat{\psi}(\delta_u) - G_{\max}(\delta_u + h)h|_{(L^2)^2}}{|Sh|_{(L^q)^2}} \\ &\leq C \lim_{h\to 0} \frac{|\hat{\psi}(\delta_u + h) - \hat{\psi}(\delta_u) - G_{\max}(\delta_u + h)h|_{(L^2)^2}}{|Sh|_{(L^q)^2}} = 0\end{aligned}$$

with some constant $C > 0$. This completes the proof. □

For a related result in the context of the reduced problem (1.3) see [36, Theorem 11].

Before we state our convergence result for the rpdAS let us discuss the assumptions in Proposition 5.4. We define $\tilde{Z} = W^2_{1+\epsilon} \times H$, and we suppose that $a, b \in L^q(Q)^2$ with $q > 2$ are given. Assume further that the SQP-method is initialized by

$$(y^0, u^0, p^0, \hat{\lambda}^0) \in W \times L^2(Q)^2 \times \tilde{Z} \times L^2(Q)^2.$$

Note that by the structure of the QP-problems (QP) at $(y,u,p) = (y^{n-1},u^{n-1},p^{n-1})$, $n \geq 1$, the corresponding first order system analogous to (3.7), and Propositions 2.3 and 2.4 we have

$$(\delta^n_y, \delta^n_u, \delta^n_p, \hat{\lambda}^n) \in W \times L^2(Q)^2 \times \tilde{Z} \times L^2(Q)^2.$$

Consequently, the iterates of the SQP-algorithm satisfy

$$(y^n, u^n, p^n, \hat{\lambda}^n) \in W \times L^2(Q)^2 \times \tilde{Z} \times L^2(Q)^2 \quad \forall\, n \geq 0.$$

Applying Proposition 5.4 yields for all $n \in \mathbb{N}_0$

$$t^n = t(y^n, u^n, p^n) \in L^q(Q)^2 \quad \forall\, 2 < q < \delta_\epsilon.$$

Now we can apply the convergence result of [7] (see also [23, 37]) for Newton's method for slantly differentiable mappings.

Theorem 5.5. *Let the assumptions of Proposition* 5.4 *be satisfied. Then the reduced pdAS is equivalent to a generalized Newton method and converges at a locally superlinear rate, i.e.,*

$$|\delta_{u,l+1} - \delta^n_u|_{(L^2)^2} = o(|\delta_{u,l} - \delta^n_u|_{(L^2)^2}) \quad \forall\, l$$

with $\delta_{u,0}$ sufficiently close to δ^n_u, the solution to (OR^n).

The above convergence result is only a local result, i.e., one has to find an initial point close to the solution of the QP-subproblem (OR). In our numerical test runs we have never observed problems with respect to convergence of the rpdAS with initialization $\delta_{u,0} = 0$. In fact, after two or three SQP iterations it turns out that the rpdAS requires only one iteration to terminate successfully; see, e.g., Tables 7.1–7.3. Further, if the SQP-method converges to a solution (stationary point) of the control problem, we have that δ^n_u approaches zero in the course of the SQP-iterations. Thus, it is to be expected that initializing with $\delta_{u,0} = 0$ yields a starting point for rpdAS such that Theorem 5.5 holds true.

6. Convergence analysis of the SQP-iteration. We analyze the SQP-iteration by utilizing generalized equations; see, e.g., [31]. Let $\epsilon \in (0, \frac{1}{3}]$, and let $2 < q < \delta_\epsilon$ with δ_ϵ as is Proposition 2.1. We consider the spaces

$$\begin{aligned} D &= W \times L^q(Q)^2 \times \tilde{Z}, \\ R &= L^{1+\epsilon}(V^*) \cap W^* \times Z^* \times L^q(Q)^2, \end{aligned}$$

and we recall the definitions of Z, W, and $W^2_{1+\epsilon}$ in (2.1)–(2.2). The norms $|\cdot|_D$ and $|\cdot|_R$ are the sums of the component norms, respectively. From now on we assume that $a, b \in L^q(Q)^2$. Note that we use $L^q(Q)^2$ in the definitions of D and R rather than $L^2(Q)^2$. This is feasible because if we initialize the SQP-method by $u^0 \in L^q(Q)^2$ and observe that due to $a, b \in L^q(Q)^2$ we have $\delta_u^n \in L^q(Q)^2$ for all n, then $u^n + \delta_u^n \in L^q(Q)^2$ for all n. Utilizing the tools employed in the proof of Proposition 5.4 for the analysis of r^n and $H^n \delta_u^n$, we further obtain $\hat{\lambda}^n \in L^q(Q)^2$ for all n.

First we convert the system (3.2) into the generalized equation

$$0 \in F(d) + T(u) \tag{6.1}$$

with $d = (y, u, p)$ by defining $F : D \to R$ as

$$\begin{aligned} F_1(d) &:= J_y(y, u) + e_y^*(y, u)p, \\ F_2(d) &:= e(y, u), \\ F_3(d) &:= J_u(y, u) + e_u^*(y, u)p \end{aligned}$$

and the set-valued map $T : D \to 2^R$ as

$$T(u) := (\{0\}, \{0\}, N(u)).$$

By $N(u)$ we denote the cone

$$N(u) = \begin{cases} \{\varphi \in L^q(Q)^2 : (\varphi, v - u) \le 0 \text{ for all } v \in U_{\mathrm{ad}}\} & \text{if } u \in U_{\mathrm{ad}}, \\ \emptyset & \text{else.} \end{cases}$$

Observe that $N(u)$ is the normal cone of $U_{\mathrm{ad}} \subset U$ intersected with $L^q(Q)^2$. Further, T has a closed graph, and F is of class $\mathcal{C}^{1,1}$.

The generalized Newton method for (6.1) is defined as follows. Let d^n denote the actual iterate. Then the next iterate d^{n+1} is defined by

$$\text{find } \delta_d : \quad 0 \in F(d^n) + F'(d^n)\delta_d + T(d^n + \delta_d), \tag{6.2a}$$

$$d^{n+1} = d^n + \delta_d. \tag{6.2b}$$

A straightforward computation verifies that (6.2) is equivalent to (3.7) at $(y, u, p) = (y^n, u^n, p^n)$.

In order to prove quadratic convergence of the process (6.2) we use the concept of strong regularity of (6.1). The notion of strong regularity of a generalized equation was introduced in [31].

Definition 6.1. The inclusion (6.1) is called *strongly regular* at $d^* \in D$ if there exist $r_1, r_2, C_L > 0$ such that for all perturbations $\eta \in B_{r_1}(0_R)$ the linearized equation

$$\eta \in F(d^*) + F'(d^*)(d - d^*) + T(d) \tag{6.3}$$

admits a unique solution $d \in B_{r_2}(d^*)$ with a Lipschitz continuous solution operator $d : B_{r_1}(0_R) \to B_{r_2}(d^*)$, i.e.,

$$|d(\eta_1) - d(\eta_2)|_D \leq C_L |\eta_1 - \eta_2|_R \ \forall \, \eta_1, \eta_2 \in B_{r_1}(0_R).$$

For the following discussion we rely on the assumption which is stated next. Let $x^* = (y^*, u^*)$ denote a solution to problem (1.1) with p^* as the corresponding adjoint state.

Assumption 6.2.

- There exists $c > 0$ such that $(J_{uu}(x^*)v, v)_U \geq c|v|_U^2$ for all $v \in U$.
- $J_{yy}(x^*)$ is positive semidefinite.
- $J_y(x^*) \in L^{1+\epsilon} \cap W^*$ is sufficiently small.

We point out that the smallness assumption on J_y can be relaxed by requiring the positive definiteness of the Hessian of the Lagrange function on the subspace associated with the linearized constraints; see [35] for further details in this respect. We are now prepared to formulate the local convergence theorem for (6.2).

Theorem 6.3. *Let d^* denote a solution of* (6.1) *which satisfies Assumption* 6.2. *Then there exist constants $r > 0$ and $C > 0$ such that for all starting values $d^0 \in B_r(d^*)$ the generalized Newton method* (6.2) *generates a sequence $\{d^n\} \subset B_r(d^*)$ which converges quadratically to d^*, i.e.,*

$$|d^{n+1} - d^*|_D \leq C|d^n - d^*|_D^2 \ \forall \, n \geq 0.$$

Following the concepts in [1, 12, 34], for a proof of Theorem 6.3 we first investigate the strong regularity of (6.1) at $d^* = (y^*, u^*, p^*)$. This is the content of the next lemma.

Lemma 6.4. *Let d^* denote a solution of* (6.1) *which satisfies Assumption* 6.2. *Then the generalized equation* (6.1) *is strongly regular at d^*.*

Proof. We check the conditions of definition 6.1 by utilizing the analysis developed in [27].

First we show that the generalized equation (6.3) admits a unique solution $d = (x, p)^\top$. For this purpose let $\eta = (\eta_1, \eta_2, \eta_3)^\top \in R$, and define

$$\begin{aligned} f(y,u) := \langle J_y, \delta_y \rangle_{W^*,W} + \frac{1}{2}\langle J_{yy}\delta_y, \delta_y \rangle_{W^*,W} + \langle e_{yy}(\delta_y, \delta_y), p^* \rangle_{Z^*,Z} + (J_u, \delta_u)_U \\ + \frac{1}{2}(J_{uu}(\delta_u), \delta_u)_U - \langle \eta_1, \delta_y \rangle_{W^*,W} - (\eta_3, \delta_u)_U, \end{aligned}$$

where we use $\delta_y = y - y^*$, $\delta_u = u - u^*$, $J = J(x^*)$, and $e = e(x^*)$ for the ease of notation. Next we consider the minimization problem

$$\min_{y,u} f(y,u) \text{ subject to } e + e_y\delta_y + e_u\delta_u = \eta_2 \text{ in } Z^* \text{ and } u \in U_{\text{ad}}. \tag{6.4}$$

Note that its necessary optimality condition coincides with (6.3). These conditions also would be sufficient if f were convex. But lack of convexity can only arise through the term

$$\langle e_{yy}(\delta_y, \delta_y), p^* \rangle_{Z^*,Z}$$

since it may be negative. Using a similar technique to the one in the proof of [27, Lemma 5.1], it is not difficult to show that for $J_y(x^*)$ small enough, there holds for some $\kappa > 0$

$$\ell_{xx}(d)(\delta_x, \delta_x) \geq \kappa \left(|\delta_y|_W^2 + |\delta_u|_{L^2(Q)^2}^2 \right) \tag{6.5}$$

for all $\delta_x \in W \times U_{\rm ad}$ with $e_y\delta_y + e_u\delta_u = 0$ in Z^*. Here we have set $\delta_x = (\delta_y, \delta_u)$, and ℓ denotes the Lagrangian associated to the minimization problem (6.4), i.e., $\ell : W \times U_{\rm ad} \times \tilde{Z} \to \mathbb{R}$ with

$$\ell(d) = f(y,u) + \langle e + e_y(y-y^*) + e_u(u-u^*) - \eta_2, p\rangle_{Z^*,Z}.$$

The relation (6.5) implies that (6.4), under Assumption 6.2, admits a unique solution $(y(\eta), u(\eta))$.

Next we argue Lipschitz continuity of the solution w.r.t η. First note that the solution $(y(\eta), u(\eta))$ satisfies the following variational inequality:

$$\ell'(y(\eta), u(\eta), p(\eta))(y - y(\eta), u - u(\eta), p - p(\eta)) \geq 0 \quad \forall\, (y,u,p) \in W \times U_{\rm ad} \times Z. \tag{6.6}$$

Here $p(\eta)$ denotes the adjoint state associated with $(y(\eta), u(\eta))$. We denote by prime the differentiation w.r.t. (y,u,p), and we use $d(\eta) = (y(\eta), u(\eta), p(\eta))$. Now let $\eta^1, \eta^2 \in R$ be given. To simplify the notion we define $\delta_y^\eta = y(\eta^1) - y(\eta^2)$ and analogously for δ_u^η, δ_x^η, and δ_p^η. Below the constant C can take different values on different occasions. A straightforward computation shows that

$$\begin{aligned} 0 &\leq \ell'(d(\eta^2))(\delta_x^\eta, \delta_p^\eta) - \ell'(d(\eta^1))(\delta_x^\eta, \delta_p^\eta) \\ &= \langle \eta_1^1 - \eta_1^2, \delta_y^\eta\rangle_{W^*,W} - \langle \eta_2^1 - \eta_2^2, \delta_p^\eta\rangle_{Z^*,Z} + (\eta_3^1 - \eta_3^2, \delta_u^\eta) \\ &\quad - \ell''(d^*)(\delta_x^\eta, \delta_x^\eta). \end{aligned} \tag{6.7}$$

Proposition 2.3 and 2.4 yield

$$|\delta_y^\eta|_W \leq C(|\delta_u^\eta|_{L^2(Q)^2} + |\eta_2^1 - \eta_2^2|_{Z^*}), \tag{6.8}$$

$$|\delta_p^\eta|_Z \leq C(|\delta_y^\eta|_W + |\eta_1^1 - \eta_1^2|_{W^*}). \tag{6.9}$$

From (6.5) and (6.7)–(6.9) we infer

$$\kappa|\delta_u^\eta|^2_{L^2(Q)^2} \leq C\big(|\eta^1 - \eta^2|_R |\delta_u^\eta|_{L^2(Q)^2} + |\eta^1 - \eta^2|_R^2\big). \tag{6.10}$$

Using Young's inequality we obtain

$$|\eta^1 - \eta^2|_R |\delta_u^\eta|_{L^2(Q)^2} \leq \frac{1}{2\kappa}|\eta^1 - \eta^2|_R^2 + \frac{\kappa}{2}|\delta_u^\eta|^2_{L^2(Q)^2}.$$

Therefore (6.10) implies

$$|\delta_u^\eta|_{L^2(Q)^2} \leq C|\eta^1 - \eta^2|_R. \tag{6.11}$$

Further Proposition 2.4 yields the existence of a constant $C > 0$ such that

$$|\delta_p^\eta|_{\tilde{Z}} \leq C(|\delta_y^\eta|_W + |\eta_1^1 - \eta_1^2|_{L^{1+\epsilon}(V^*)\cap W^*}). \tag{6.12}$$

The variational inequality (6.6) yields

$$\alpha u(\eta) - \eta_3 + e_u^* p(\eta) \in N(u(\eta)).$$

This is equivalent to

$$u(\eta) = P_{U_{\rm ad}}\big(\alpha^{-1}(\eta_3 - e_u^* p(\eta))\big) \in L^q(Q)^2,$$

where $P_{U_{\mathrm{ad}}}$ denotes the (pointwise) projection onto U_{ad}. Hence, we have

$$|u(\eta^1) - u(\eta^2)|_{L^q(Q)^2} \leq C\big(|\eta_3^1 - \eta_3^2|_{L^q(Q)^2} + |\delta_p^\eta|_{\tilde{Z}}\big).$$

Finally, combining the last estimate with (6.11)–(6.12) and (6.8) results in

$$|d(\eta^1) - d(\eta^2)|_D \leq C|\eta^1 - \eta^2|_R.$$

This completes the proof. □

We point out that the approach taken in the proof of Lemma 6.4 is related to the technique utilized in [34, sections 3–4].

Once we have established the strong regularity of (6.1) at d^*, the proof of Theorem 6.3, i.e., the locally quadratic convergence rate for (6.2), follows from [1, 12]; see also [34, Theorem 3.3]. Since the Newton process (6.2) is equivalent to the SQP-iteration in Algorithm 4.1 we readily obtain the same locally quadratic convergence rate for $\{(x^n, p^n)\}$ produced by Algorithm 4.1.

Remark 6.5. The smallness assumption imposed on $J_y(y^*, u^*)$ is commonly used in the literature; see [27] and the references therein. In the case of tracking-type functionals it can be guaranteed in the case of exact (or ϵ) controllability of the desired state; compare [8, 13].

7. Numerical experiments.

The control problem considered here is the tracking of the Stokes flow z in a cavity; see Figure 7.1, left. Its formulation is given by (P) with the cost functional

$$J(y,u) := \frac{1}{2}\int_Q |y - z|^2\, dxdt + \frac{\alpha}{2}\int_Q |u|^2\, dxdt.$$

Here, $Q := (0,T) \times \Omega$ with $\Omega := (0,1)^2$ and $T := 1$. The desired state $z(t,x) = (z_1, z_2)^\top$ is chosen such that $z(t,\cdot) = s(\cdot)$ for every time instance $t \in (0,T)$, where $s = (s_1, s_2)^\top$ denotes the stationary Stokes flow in the domain Ω with inhomogeneous boundary condition $s_1 = 1, s_2 = 0$ on $(0,1) \times \{1\}$ and $s = 0$ on the rest of $\partial\Omega$. The same boundary conditions are prescribed for the flow y on $(0,T) \times \partial\Omega$. The value of the kinematic viscosity is $\nu = \frac{1}{400}$, and the initial velocity y_0 is taken as stationary Navier–Stokes flow corresponding to ν, with the same boundary conditions as s. Unless otherwise specified, for the bounds on the controls we use box constraints (constant in time and space) with the values $a = -0.5$ and $b = 0.5$. We note that the

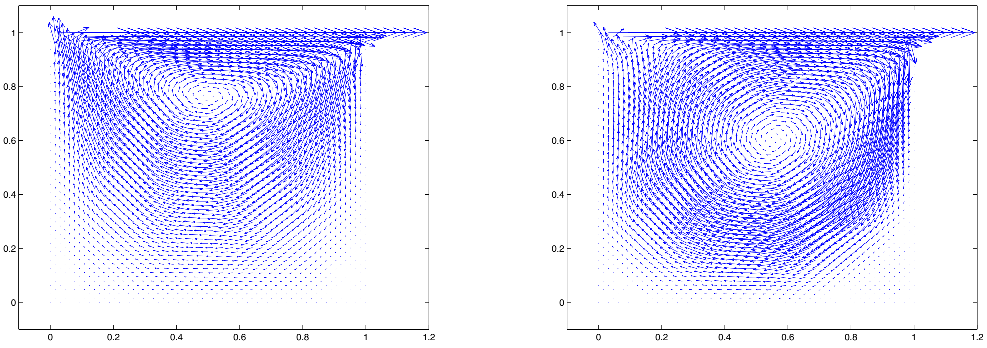


FIG. 7.1. *Target flow s (left) and initial condition (right).*

first term in the cost functional values the control gain when tracking the state z, and the second term measures the control cost, where $\alpha > 0$ denotes a weighting factor.

We mention that sufficiently smooth inhomogeneous boundary conditions may be incorporated into our theoretical framework through a transformation of the form $y \to y - y'$ with y' satisfying the inhomogeneous boundary conditions. This concept even carries over to Dirichlet boundary control; see [14, 28]. However, it was mentioned in [24] and [36], that the boundary conditions taken here are not smooth enough to allow this transformation. In this respect, also the robustness of our numerical approach is tested by the chosen numerical example.

For the results presented an equidistant time discretization is chosen with step length $\delta t = 0.00625$, and for the spatial discretization the Taylor–Hood finite element [29] is used on a grid containing 1024 triangles with 2113 velocity and 545 pressure nodes. The time integration for forward systems like (2.8) is performed semi-implicitly, i.e., implicitly in the diffusive part and explicitly in the convective parts. For adjoint systems like (2.9) we take the transpose of the scheme applied to the corresponding forward system.

Numerical results for this flow configuration in the unconstrained case are presented in [27] (Newton's method), [24] (SQP method), and [25] (comparison of Newton's and the SQP method). The constrained case with the same bounds is considered in [36], where a semismooth Newton method is applied for the numerical solution of the reduced control problem (1.3).

Let us comment on our discrete approach. We discretize each part of the overall solution algorithm separately. For the subproblems, like, e.g., (3.7), we apply a discrete concept which is closely related to [26]. In our approach, the control variables are not discretized explicitly but implicitly through the first order optimality conditions (like, e.g., (3.7)). We resolve the corresponding active sets in (5.9) only on the grid induced by the velocity nodes (vertices of the triangulation together with edge midpoints). For this reason, it suffices to manage control functions in terms of their function values on the same grid, since control functions enter into our algorithmical approach only in terms of arguments of functionals. To evaluate functionals numerically, we utilize appropriate quadrature rules based on function values. For the function values of the control obtained by this procedure we may pick a Rothe function which is piecewise constant (or piecewise linear and continuous) in time, piecewise linear on the velocity grid, and continuous in space. This function is then contained in U_{ad}. See [26] for a detailed discussion of related discretization concepts, including error estimates and numerical examples.

Utilizing this discrete technique, we obtain approximations of the reduced gradient and of "reduced Hessian times increment" operations with respective approximation errors of the order of the discretization error of the Navier–Stokes equations. (See [9] for an analysis of the latter type of error.) As a consequence, it is only meaningful to monitor fast convergence of the SQP-framework of Algorithm 4.1 for stopping tolerances of the order of the discretization error of the state equations.

Finally we note that in the practical implementation the component sets of the active sets are computed via their complements, the componentwise inactive sets, where the inequalities are relaxed by relative errors of the order of the machine precision.

In what follows all iterates represent discrete quantities. Including the primal, adjoint, and control variables, the number of unknowns in the discretized control problem is 2.40288×10^6. The termination criterion for the outer SQP-iteration in Algorithm 4.1 is chosen as $|(\delta x^n, \delta p^n)| \leq \text{tol}_{\text{SQP}} := 5 \times 10^{-3}$. Here $|\cdot|$ is the norm on $W \times U \times Z$. Note that the results in [9] let us expect a discretization error of $\mathcal{O}(\delta t)$ for

Table 7.1
Performance of Algorithm 4.1 *for* $\alpha = 1.\text{e-}1$.

Iteration	pdas-steps	CG-steps	q^n	l^n	c^n	$J(x^n)$
1	4	1:5,2:5,3:6,4:0	3.53	1	4.48372e-2	1.18135e-2
2	2	1:8,2:0	2.06	5.84e-1	1.50152e-2	5.50584e-3
3	2	1:7,2:0	1.12	1.86e-1	5.03267e-3	4.83819e-3
4	2	1:7,2:0	9.33	2.88e-1	7.41022e-4	4.7917e-3
5	2	1:7,2:0	14.66	1.3e-1	9.99308e-5	4.79280e-3

our discretization described above. discretization error is discussed in detail below. In iteration n, the stopping tolerance in Algorithm 5.1 is chosen as $\epsilon = \text{tol}^n$ with

$$\text{tol}^n = 0.1|\Psi(\delta_{u,0}, \hat{\lambda}_0; u^n) - \hat{\lambda}_0|_{(L^2)^2}.$$

Alternatively, we stop Algorithm 5.1 as soon as the active nodes of two successive iterations coincide. The inner CG-loop of Algorithm 5.1, i.e., Algorithm 5.2, is terminated if the iterate v_k satisfies

$$|(H^n v_k - r^n)_{\mathcal{I}_l}|_{(L^2)^2} < 0.01|(H^n v_0 - r^n)_{\mathcal{I}_l}|_{(L^2)^2}.$$

All computations were performed on a Dell laptop computer with 1.7 GHz CPU. In the tables that follow, the column of CG-steps is to be read as follows: $l : b$ indicates that in the lth pdas-iteration (Algorithm 5.1) b cycles of Algorithm 5.2 are performed until its stopping criterion is met.

We present test runs for $\alpha = 0.1$ and $\alpha = 0.01$. In both runs the Lagrange multiplier $\hat{\lambda}$ in the pdas-Algorithm 5.1 is initialized with the right-hand side of system (OR). Subsequently we use

$$q^n = \frac{|(\delta_x^n, \delta_p^n)|}{|(\delta_x^{n-1}, \delta_p^{n-1})|^2} \text{ and } l^n = \frac{|(\delta_x^n, \delta_p^n)|}{|(\delta_x^{n-1}, \delta_p^{n-1})|}$$

to study the convergence speed of the SQP-method and

$$c^n = |\hat{\Psi}(\delta_u^n) + \alpha(u^n + \delta_u^n) - S^n \delta_u^n - t^n|_{(L^2)^2}$$

to measure the residual in the complementarity system at iteration n.

Run 1 (for $\alpha = 0.1$). The state, control, and adjoint variables of the equality constraint are initialized with zero, respectively. In Figure 7.1 the desired flow (left plot) together with the initial flow (right plot) is shown.

In Table 7.1 we summarize our numerical results. The optimal value of the cost functional for the unconstrained problem is $J^* = 4.76846856\text{e-}3$, which is only slightly smaller than the corresponding value in the constrained case. This is because in the numerical solution there are only 8451 active controls. This corresponds to approximately 1.25% of the total number of controls. From Table 7.1 we see that the active set for the QPs is identified after at most four iterations of the primal-dual active set strategy (Algorithm 5.1) for solving the quadratic subproblems. We recall that whenever the number of pdas-iterations is 2, then the active set is detected immediately (within the stopping tolerance), and Algorithm 5.1 stops successfully after the first cycle. This behavior is typical in our test runs also for other choices of the parameters involved in the optimization problem. Moreover, we can study the impact of the discretization error on the convergence speed of the algorithm. Iterations 1–3 indicate

Table 7.2
Performance of Algorithm 4.1 *for* $\alpha = 1$.e-1 *and bounds* $a = -0.05$, $b = 0.05$.

Iteration	pdas-steps	CG-steps	q^n	l^n	c^n	$J(x^n)$
1	2	1:4,2:0	8.89	1	6.27174e-3	1.18135e-2
2	2	1:4,2:0	2.68	3.02e-1	8.43381e-4	8.92351e-3
3	2	1:5,2:0	0.72	2.44e-2	4.99023e-5	8.91613e-3

Table 7.3
Performance of Algorithm 4.1 *for* $\alpha = 1$.e-2.

Iteration	pdas-steps	CG-steps	q^n	l^n	c^n	$J(x^n)$
1	3	1:12,2:16,3:0	16.52	1	1.31661e-3	2.09569e-3
2	2	1:13,2:0	2.60	1.57e-1	1.50761e-4	2.05383e-3
3	2	1:15,2:0	13.51	1.29e-1	1.55306e-5	2.05460e-3

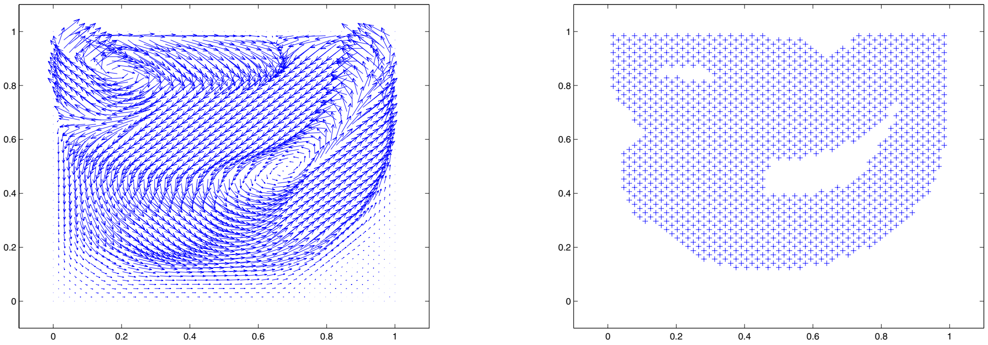


Fig. 7.2. *Optimal control (left) together with active set at* $t = 0.1$, *Run 2.*

the fast local convergence behavior. At iteration 4 the order of the discretization error is reached and subsequently the convergence speed is reduced; compare the last two rows in the q^n and l^n columns, respectively.

In Table 7.2 we study the influence of the box constraints. In this case we pick $a = -0.05$ and $b = 0.05$. As one can see, our algorithm detects the numerical solution after three iterations of Algorithm 4.1. Moreover, for the present choice of parameters, the active sets are detected immediately, and we observe fast local convergence. The numerical solution is active in 447663 controls, which corresponds to 66.21% of the total number of controls.

Run 2 ($\boldsymbol{\alpha = 0.01}$). Initializing state, control, and adjoint of the equality constraint with zero does not yield convergence for the current choice of parameters. This reflects the local character of the method. Instead, we initialize Algorithm 4.1 with the numerical solution $(x^0, p^0) = ((y^*, u^*), p^*)$ obtained for $\alpha = 0.02$. Our numerical findings are summarized in Table 7.3. Obviously, our choice (x^0, p^0) is a good initial guess which delivers fast local convergence after three iterations, with only a moderate number of CG iterations within Algorithm 5.1. In the numerical solution 90204 controls are active. This corresponds to 13.34% of the total number of controls. In the unconstrained case the cost functional takes the optimal value $J^* = 1.47922$e-3.

In Figure 7.2 a spatial snapshot of the control together with the corresponding active set at $t = 0.1$ is presented. Figure 7.3 shows the optimally controlled flow at $t = 0.1, t = 0.5$, and $t = 1$ (from left to right).

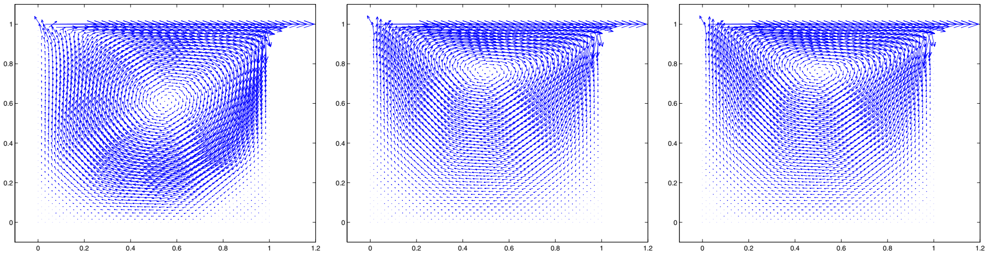


FIG. 7.3. *Optimally controlled flow at* $t = 0.1$ *(left),* $t = 0.5$ *(middle), and* $t = 1$ *(right),* $\alpha = 1.\text{e-}2$*, Run 2.*

With respect to our initial choice (x^0, p^0), we note that in [36] convergence of a semismooth Newton algorithm with zero initial control is reported for a similar parameter setting as in the present run. This suggests that a Newton technique for solving (1.3) exhibits a more robust behavior w.r.t. the initial choice. For the unconstrained case, in [25] a similar observation was made.

REFERENCES

[1] W. Alt, *The Lagrange-Newton method for infinite-dimensional optimization problems*, Numer. Funct. Anal. Optim., 11 (1990), pp. 201–224.

[2] H. Amann, *Compact embeddings of vector-valued Sobolev and Besov spaces*, Glas. Mat. Ser. III, 35 (2000), pp. 161–177.

[3] M. Berggren, *Numerical solution of a flow-control problem: Vorticity reduction by dynamic boundary action*, SIAM J. Sci. Comput., 19 (1998), pp. 829–860.

[4] M. Bergounioux, K. Ito, and K. Kunisch, *Primal-dual strategy for constrained optimal control problems*, SIAM J. Control Optim., 37 (1999), pp. 1176–1194.

[5] M. Bergounioux, M. Haddou, M. Hintermüller, and K. Kunisch, *A comparison of a Moreau-Yosida based active set strategy and interior point methods for constrained optimal control*, SIAM J. Optim., 11 (2000), pp. 495–521.

[6] T.R. Bewley, *Flow control: New challenges for a new rennaissance*, Progr. Aerospace Sci., 37 (2001), pp. 21–58.

[7] X. Chen, Z. Nashed, and L. Qi, *Smoothing methods and semismooth methods for nondifferentiable operator equations*, SIAM J. Numer. Anal., 38 (2000), pp. 1208–1216.

[8] F. Colonius and K. Kunisch, *Output least squares stability for parameter estimation in two point value problems*, J. Reine Angew. Math., 370 (1986), pp. 1–29.

[9] K. Deckelnick and M. Hinze, *Error estimates in space and time for tracking-type control of the instationary Stokes system*, in Control and Estimation of Distributed Parameter Systems, W. Desch, F. Kappel, and K. Kunisch, eds., Internat. Ser. Numer. Math., Birkhäuser, Basel, 143, 2002, pp. 87–103.

[10] R. Dembo, S. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[11] M. Desai and K. Ito, *Optimal controls of Navier–Stokes equations*, SIAM J. Control Optim., 32 (1994), pp. 1428–1446.

[12] A. Dontchev, *Local analysis of a Newton-type method based on partial linearization*, in Proceedings of the AMS-SIAM Summer Seminar in Applied Mathematics on Mathematics and Numerical Analysis: Real Number Algorithms, AMS Lectures in Appl. Math. 32, J. Renegar, M. Shub, and S. Smale, eds., AMS, Providence, RI, 1996, pp. 295–306.

[13] A. L. Dontchev and T. Zolezzi, *Well-posed Optimization Problems*, Lecture Notes in Math., 1543, Springer, Berlin, 1993.

[14] A. V. Fursikov, M. D. Gunzburger, and L. S. Hou, *Boundary value problems and optimal boundary control for the Navier–Stokes systems: The two-dimensional case*, SIAM J. Control Optim., 36 (1998), pp. 852–894.

[15] O. Ghattas and J. Bark, *Optimal control of two- and three-dimensional incompressible Navier–Stokes flow*, J. Comput. Phys., 136 (1997), pp. 231–244.

[16] S. Eisenstat and H. Walker, *Choosing the forcing terms in an inexact Newton method*, SIAM J. Sci. Comput., 17 (1996), pp. 16–32.

[17] R. Glowinski, *Finite element methods for the numerical simulation of incompressible viscous flow. Introduction to the control of the Navier–Stokes equations*, Lecture Notes in Appl. Math. 28, Springer, New York, 1991, pp. 219–301.

[18] M. Gunzburger and S. Manservisi, *Analysis and approximation of the velocity tracking problem for Navier–Stokes flows with distributed control*, SIAM J. Numer. Anal., 37 (2000), pp. 1481–1512.

[19] M. Heinkenschloss, *Formulation and analysis of a sequential quadratic programming method for the optimal Dirichlet boundary control of Navier–Stokes flow*, in Optimal Control: Theory, Algorithms and Applications, Kluwer Academic Publishers, Dordrecht, the Netherlands, 1998, pp. 178–203.

[20] M. Hintermüller, *A primal-dual active set algorithm for bilaterally control constrained optimal control problems*, Quart. Appl. Math., 61 (2003), pp. 131–161.

[21] M. Hintermüller, *On a globalized augmented Lagrangian-SQP algorithm for nonlinear optimal control problems with box constraints*, in Fast Solution Methods for Discretized Optimization Methods, Internat. Ser. Numer. Math. 138, K. H. Hoffmann, R. Hoppe, and V. Schulz, eds., Birkhäuser, Basel, 2001, pp. 139–153.

[22] M. Hintermüller and M. Hinze, *Globalization of SQP-methods in control of the instationary Navier–Stokes equations*, Math. Model. Numer. Anal., 36 (2002), pp. 725–746.

[23] M. Hintermüller, K. Ito, and K. Kunisch, *The primal-dual active set strategy as a semismooth Newton method*, SIAM J. Optim., 13 (2003), pp. 865–888.

[24] M. Hinze, *The SQP-method for tracking-type control of the instationary Navier–Stokes equations*, in Proceedings of the Algoritmy 2000, Slovak University of Technology, A. Handlovicova, K. Mikula, and M. Komornikova, eds., 2000.

[25] M. Hinze, *A remark on second order methods in control of fluid flow*, Z. Angew. Math. Mech., 81 (2001), pp. 791–792.

[26] M. Hinze, *A generalized discretization concept for optimal control problems with control constraints*, Comput. Optim. Appl., 30 (2005), pp. 45–61.

[27] M. Hinze and K. Kunisch, *Second order methods for optimal control of time-dependent fluid flow*, SIAM J. Optim Control, 40 (2001), pp. 925–946.

[28] M. Hinze and K. Kunisch, *Second order methods for boundary control of the instationary Navier–Stokes system*, Z. Angew. Math. Mech., 84 (2004), pp. 171–187.

[29] P. Hood and C. Taylor, *A numerical solution of the Navier–Stokes equations using the finite element technique*, Comput. & Fluids, 1 (1973), pp. 73–100.

[30] J. L. Lions, *Control of Distributed Singular Systems*, Gauthier–Villars, Paris, 1985.

[31] S. M. Robinson, *Strongly regular generalized equations*, Math. Oper. Res., 5 (1980), pp. 43–62.

[32] R. Temam, *Navier-Stokes Equations*, 2nd ed., Stud. Math. Appl. 2, North–Holland, Amsterdam, 1985.

[33] H. Triebel, *Interpolation Theory, Function Spaces, Differential Operators*, 2nd ed., J. A. Barth-Verlag, Heidelberg, Leipzig, 1995.

[34] F. Tröltzsch and S. Volkwein, *The SQP method for control constrained optimal control of the Burgers equation*, ESAIM Control Optim. Calc. Var., 6 (2001), pp. 649–674.

[35] F. Tröltzsch and D. Wachsmuth, *Second-order sufficient optimality conditions for the optimal control of Navier-Stokes equations*, Tech. report, 30-2003, Department of Mathematics, TU Berlin, 2003.

[36] M. Ulbrich, *Constrained optimal control of Navier-Stokes flow by semismooth Newton methods*, Systems Control Lett., 48 (2003), pp. 297–311.

[37] M. Ulbrich, *Semismooth Newton methods in function spaces*, SIAM J. Optim., 13 (2003), pp. 805–842.

SIAM J. OPTIM.
Vol. 16, No. 4, pp. 1201–1210

LOSS OF SUPERLINEAR CONVERGENCE FOR AN SQP-TYPE METHOD WITH CONIC CONSTRAINTS*

MORITZ DIEHL†, FLORIAN JARRE‡, AND CHRISTOPH H. VOGELBUSCH‡

Abstract. In this short note we consider a sequential quadratic programming (SQP)–type method with conic subproblems and compare this method with a standard SQP method in which the conic constraint is linearized at each step. For both approaches we restrict our attention to convex subproblems since these are easy to solve and guarantee a certain global descent property. Using the example of a simple nonlinear program (NLP) and its conic reformulation we show that the SQP method with conic subproblems displays a slower rate of convergence than standard SQP methods. We then explain why an SQP subproblem that is based on a better approximation of the feasible set of the NLP results in a much slower algorithm.

Key words. semidefinite programming, nonlinear programming, sequential quadratic programming, second order cone

AMS subject classifications. 90C30, 65K05

DOI. 10.1137/050625977

1. Introduction.

We consider a nonlinear program (NLP) and its equivalent conic reformulation. We compare the standard sequential quadratic programming (SQP) method for solving the NLP and a sequential linear conic programming (SLCP) method to solve the equivalent conic reformulation. At each iteration of the SLCP method the constraint function F is replaced by its linear (Taylor) approximation and a convex quadratic objective function is used to model the nonlinear objective and the curvature of F. This step yields a convex quadratic program with linear conic constraints to which we refer to as an SLCP subproblem. The standard SQP subproblem is obtained if the cone is linearized as well. Instead, for the SLCP method, the cone is maintained unperturbed and the convex quadratic objective term is then replaced by an equivalent second order cone constraint, so that the SLCP subproblem is in fact a linear conic program explaining the name SLCP method.

Efficient interior point solvers for solving the conic SLCP subproblems are available only when the Hessian of the conic subproblem is positive semidefinite.[1] Throughout this paper we therefore restrict our comparison to SLCP subproblems that use a positive semidefinite approximation of the Hessian of the Lagrangian.

Our comparison is based on problem (2.1) below. This problem satisfies the strong second order sufficient conditions for local optimality. Hence, the standard SQP approach will always converge to the optimum if the initial point is sufficiently close to the optimum and the semidefinite approximations B_k of the Hessians used in the SQP subproblems are bounded. Thus the assumptions of Theorem 3 of [4] are

*Received by the editors March 4, 2005; accepted for publication (in revised form) October 25, 2005; published electronically March 24, 2006.
http://www.siam.org/journals/siopt/16-4/62597.html

†Universität Heidelberg, Interdisciplinary Center for Scientific Computing (IWR), Im Neuenheimer Feld 368, 69120 Heidelberg, Germany (M.Diehl@IWR.Uni-Heidelberg.de).
‡Universität Düsseldorf, Mathematisches Institut, Universität sstr. 1, 40225 Düsseldorf, Germany (jarre@opt.uni-duesseldorf.de, vogelbusch@opt.uni-duesseldorf.de).

[1] Moreover, even if nonconvex quadratic SQP subproblems could be used, there are examples where the full SQP step starting at a feasible iterate returns an infeasible point and a short multiple of the SQP step is an ascent direction for the objective function. In such situations additional techniques like a trust region approach need to be used to ensure global convergence.

satisfied when the matrices B_k are defined by a damped quasi-Newton update and the resulting SQP method is locally R-superlinearly convergent. If the exact Hessian is used, locally quadratic convergence is achieved.

As the standard SQP approach with a suitable positive semidefinite approximation of the Hessian is superlinearly convergent, one might expect that superlinear convergence will hold for the seemingly "better" approximation of the subproblem in the SLCP approach. The surprising result of sections 3 and 5 is that superlinear convergence is lost if, instead of linearizing the constraint of problem (2.1), the equivalent conic constraint is maintained without change in the SLCP subproblem.

While the restriction to a positive semidefinite Hessian does not destroy superlinear convergence for the standard SQP method it may do so in the case of SQP problems with conic constraints. The reason for this is that the Hessian of the Lagrangian at an optimum is not necessarily positive semidefinite along critical directions when the subproblem includes (nonlinear convex) conic constraints [5]. Thus, the combination of linearization and convexification is efficient, while that of partial linearization (maintaining the nonlinear convex cone) and convexification does not lead to an efficient algorithm.

2. A simple example.

We consider the NLP

$$\min\left\{-x_1^2-(x_2-1)^2 \mid \|x\|_2^2\le 1,\quad x\in\mathbb{R}^2\right\}. \tag{2.1}$$

Problem (2.1) satisfies the strong second order sufficiency conditions for optimality at its solution $(0,-1)^{\mathrm{T}}$ with corresponding multiplier $y=2$.

Let $\mathcal{K}$ be the second order cone in three dimensions, i.e.,

$$\mathcal{K} := \left\{(x_0,x_1,x_2)^{\mathrm{T}}\in\mathbb{R}^3 \mid x_0\ge\sqrt{x_1^2+x_2^2}\right\}.$$

We extend the vector

$$x=\begin{pmatrix}x_1\\x_2\end{pmatrix}\in\mathbb{R}^2 \qquad\text{to}\qquad \hat{x}=\begin{pmatrix}x_0\\x_1\\x_2\end{pmatrix}\in\mathbb{R}^3.$$

With the above definition, problem (2.1) allows an equivalent conic formulation

$$\min\{f(\hat{x}) \mid F(\hat{x})=0,\quad \hat{x}\in\mathcal{K}\}, \tag{2.2}$$

where $f:\mathbb{R}^3\to\mathbb{R}$ is defined by

$$\hat{x}\mapsto f(\hat{x})=-x_1^2-(x_2-1)^2$$

and $F:\mathbb{R}^3\to\mathbb{R}$ is defined by

$$\hat{x}\mapsto F(\hat{x})=x_0-1.$$

The Lagrangian L of (2.2) with Lagrangian multiplier $y\in\mathbb{R}$ and the dual variable $s\in\mathbb{R}^3$ are given by

$$L(\hat{x},\hat{y},\hat{s}) := f(\hat{x})-\hat{y}F(\hat{x})-\langle\hat{s},\hat{x}\rangle,$$

where $\hat{s}$ lies in the dual cone $\mathcal{K}^D$,

$$\mathcal{K}^D=\mathcal{K}.$$

The gradient g of L with respect to $\hat{x}$ is given by

$$\begin{aligned} g(\hat{x},\hat{y},\hat{s}) &:= \nabla_{\hat{x}} L(\hat{x},\hat{y},\hat{s}) = \nabla f(\hat{x}) - \begin{pmatrix} \hat{y} \\ 0 \\ 0 \end{pmatrix} - \hat{s} \\ &= \begin{pmatrix} -\hat{y} & -s_0 \\ -2x_1 & -s_1 \\ -2(x_2-1) & -s_2 \end{pmatrix} \end{aligned}$$

and the Hessian with respect to $\hat{x}$ is given by

$$\begin{aligned} \hat{H}(\hat{x},\hat{y}) &:= \nabla^2_{\hat{x}} L(\hat{x},\hat{y},\hat{s}) = D_{\hat{x}}(\nabla f(\hat{x})) \\ &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -2 \end{pmatrix}. \end{aligned}$$

The global minimizer is

$$\hat{x}^* = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix},$$

with multipliers

$$\hat{s}^* = \begin{pmatrix} s_0 \\ s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} 4 \\ 0 \\ 4 \end{pmatrix} \quad \text{and} \quad \hat{y}^* = -4$$

satisfying $g(\hat{x}^*,\hat{y}^*,\hat{s}^*) = 0$. Observe that the Hessian $\hat{H}(\hat{x},\hat{y})$ is negative semidefinite and independent of $\hat{x},\hat{y}$.

3. Linear convergence with conic constraints and projected Hessian.

In the following we analyze the local convergence properties of a basic SLCP algorithm without globalization. To simplify the notation we define

$$\hat{c}_k := \nabla f(\hat{x}_k).$$

The algorithm iterates $\hat{x}_{k+1} = \hat{x}_k + \Delta\hat{x}_k$, where $\Delta\hat{x}_k$ solves the approximation

$$\min\left\{ \hat{c}_k^{\mathrm{T}} \Delta\hat{x} + \frac{1}{2}\Delta\hat{x}^{\mathrm{T}} \hat{B}_k \Delta\hat{x} \;\middle|\; DF(\hat{x}_k)[\Delta\hat{x}] = -F(\hat{x}_k) \quad \hat{x}_k + \Delta\hat{x} \in \mathcal{K} \right\} \tag{3.1}$$

of the conic problem (2.2). Here, $\hat{B}_k$ is an approximation of the Hessian $\hat{H}(\hat{x}_k,\hat{y}_k)$ of L. Because $\hat{H}$ is constant and no other part of the above conic problem depends on $\hat{y}_k$, we need not regard multiplier iterates here.

Note that the linear equality constraint in (3.1) implies $\hat{x}_{k+1,0} = \hat{x}_{k,0} + \Delta\hat{x}_{k,0} = 1$. Thus we can assume that $\hat{x}_{k,0}$ is fixed to 1 for all $k > 0$ and simplify (3.1) by replacing the cone $\mathcal{K}$ with the inequality constraint:

$$\min\left\{ \begin{pmatrix} -2x_1 \\ -2(x_2-1) \end{pmatrix} \Delta x + \frac{1}{2}\Delta x^{\mathrm{T}} B \Delta x \;\middle|\; \|x + \Delta x\|_2 \le 1 \right\}.$$

For simplicity of notation, we have dropped the iteration index k now. We denote the projections on the (x_1,x_2)-space of the exact Hessian $\hat{H}$ and its approximation $\hat{B}$ by H and B.

If the exact Hessian $B = H$ is used we obtain the problem

$$\min\left\{\begin{pmatrix} -2x_1 \\ -2(x_2-1)\end{pmatrix}^{\mathrm{T}}\Delta x + \frac{1}{2}\Delta x^{\mathrm{T}}\begin{pmatrix} -2 & 0 \\ 0 & -2\end{pmatrix}\Delta x \;\middle|\; \|x+\Delta x\|_2 \le 1\right\}, \tag{3.2}$$

which is nonconvex. This subproblem is equivalent to the initial NLP (2.1) and would thus return the solution of the initial problem in one step. This subproblem is hence "suitably defined."

The subproblems in an SQP algorithm should easily be solved. However, nonconvex quadratic conic problems are about as difficult to solve as general nonlinear conic problems. Given efficient software packages like SeDuMi [6] or SDPT3 [7] for solving convex conic programs we search for a suitable positive semidefinite matrix B such that the above subproblem still yields rapid convergence.

The Hessian of the Lagrangian in (3.2) is $H = -2I$, and the orthogonal projection of H onto the cone of positive semidefinite matrices is simply given by $B = 0$.

We first consider the choice $B = 0$, for which the optimal solution is always on the boundary of the cone. We start close to the optimal solution at the point

$$\begin{pmatrix} x_1 \\ x_2\end{pmatrix} = \begin{pmatrix} \sin(\alpha) \\ -\cos(\alpha)\end{pmatrix},$$

where

$$0 < \alpha \ll 1. \tag{3.3}$$

Without loss of generality we will keep this choice of α also in sections 4 and 5. The case $-1 \ll \alpha < 0$ can be treated analogously.

For $B = 0$ and α as in (3.3) the conic SLCP subproblem is equivalent to

$$\min\left\{\begin{pmatrix} -2\sin(\alpha) \\ 2\,(1+\cos(\alpha))\end{pmatrix}^{\mathrm{T}}(x+\Delta x)\;\middle|\; \|x+\Delta x\|_2 \le 1\right\}, \tag{3.4}$$

the solution of which is given by

$$x + \Delta x = \begin{pmatrix} \sin\left(\frac{\alpha}{2}\right) \\ -\cos\left(\frac{\alpha}{2}\right)\end{pmatrix}.$$

Hence, the (local) convergence for $B = 0$ is linear, with a convergence rate of $\frac{1}{2}$. As indicated in the next section this result does not imply linear convergence for all choices of positive semidefinite B.

4. Superlinear convergence for unbounded positive semidefinite B.

It is well known (see, e.g., [3]) that the orthogonal projection of the Hessian as used in section 3 is not affine invariant and other semidefinite approximations of H, for example, the Hessian of the augmented Lagrangian, may be more suitable to obtain rapid local convergence of an SQP-type method.

In fact, as we show in this section, we can present a sequence of positive semidefinite matrices B_k for which the iterates x_k generated by solution of the SLCP subproblem

$$\min\left\{c^{\mathrm{T}}\Delta x + \frac{1}{2}\Delta x^{\mathrm{T}}B\Delta x \;\middle|\; \|x+\Delta x\|_2 \le 1\right\} \tag{4.1}$$

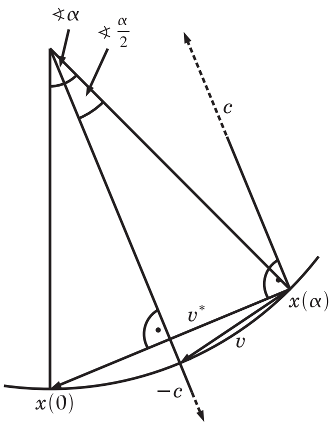


FIG. 4.1. *Visualization of the SLCP subproblem.*

converge quadratically to $(0,-1)^{\mathrm{T}}$. The SLCP subproblems based on the matrices B_k presented here have unique solutions on the boundary of the constraint set of (4.1). Thus we use again

$$x(\alpha) = \begin{pmatrix} x_1(\alpha) \\ x_2(\alpha) \end{pmatrix} = \begin{pmatrix} \sin(\alpha) \\ -\cos(\alpha) \end{pmatrix} \tag{4.2}$$

and prove the quadratic convergence with respect to α as defined in (3.3).

For $B = 0$ the result of the previous section states that the search direction $\Delta x = v$ is approximately equal to

$$v := \begin{pmatrix} \sin\left(\frac{\alpha}{2}\right) \\ -\cos\left(\frac{\alpha}{2}\right) \end{pmatrix} - \begin{pmatrix} \sin(\alpha) \\ -\cos(\alpha) \end{pmatrix} \approx \frac{\alpha}{2} \begin{pmatrix} -1 \\ -\frac{3}{4}\alpha \end{pmatrix}. \tag{4.3}$$

The method is locally superlinearly convergent if and only if this direction is perturbed to approximately

$$v^* := x^* - x(\alpha) = \begin{pmatrix} 0 \\ -1 \end{pmatrix} - \begin{pmatrix} \sin(\alpha) \\ -\cos(\alpha) \end{pmatrix} \approx \alpha \begin{pmatrix} -1 \\ -\frac{1}{2}\alpha \end{pmatrix}. \tag{4.4}$$

In Figure 4.1 the steps v and v^* are visualized.

Note that the direction v^* leading to the optimal solution $x^* = x(0)$ is orthogonal to the vector c. Hence c is the direction we like to penalize, but it is also the gradient of the objective function of (4.1) at $\Delta x = 0$. Also note that as a consequence, the SLCP subproblems (4.1) using

$$B := \frac{1}{\alpha^4} cc^{\mathrm{T}} \tag{4.5}$$

do not necessarily produce superlinearly convergent steps. Let $\mathcal{N}$ be the null space of B. For B as in (4.5) the space of optimal solutions of (4.1) is the intersection of the feasible set and

$$V := \mathcal{N} - \frac{\alpha^4 c}{16} + \mathcal{O}(\alpha^6). \tag{4.6}$$

The affine space $x(\alpha)+V$ includes a point on the boundary of the constraint set of (4.1) of the form $x(\alpha)+(v^*+\mathcal{O}(\alpha^3))$. We call such a point "close" to the optimum v^*.

To obtain a unique optimal solution for each SLCP subproblem we use a rotation with a small angle $\beta>0$ of the vector c to form a matrix B_β. We define

$$\mathrm{rot}_\beta := \begin{pmatrix} \cos(\beta) & -\sin(\beta) \\ \sin(\beta) & \cos(\beta) \end{pmatrix}, \quad c_\beta := \mathrm{rot}_\beta c, \quad \text{and} \quad B_\beta := \frac{1}{\alpha^4} c_\beta c_\beta^{\mathrm{T}}.$$

Note that for $\beta \in (0, \frac{\alpha}{4})$ the objective value of (4.1) can be improved in a direction orthogonal to the penalty direction c_β. This implies that the optimal solution of (4.1) is a unique point on the boundary of the constraint set of (4.1). For $\beta=0$ we have the case of (4.5) while for $\beta=\frac{\alpha}{4}$ we obtain the same SLCP iterates as for $B=0$.

In the following we assume $\beta \in (0, \frac{\alpha}{4})$ and consider the problem

$$\min\left\{ c^{\mathrm{T}}\Delta x + \frac{1}{2}\Delta x^{\mathrm{T}} B_\beta \Delta x \;\middle|\; \|x+\Delta x\|_2 \le 1 \right\}. \tag{4.7}$$

Recall that v^* is in the null space $\mathcal{N}$ of B and $x(\alpha)+v^*$ lies on the boundary of the constraint set of (4.1).

Let $\mathcal{N}_\beta$ be the null space of B_β and let v_β^* be the unique solution of (4.7). Note that the angle between $\mathcal{N}_\beta$ and the objective function c is less than the angle of the null space $\mathcal{N}=\mathcal{N}_{\beta=0}$ of B and c. Therefore, as in (4.6), the vector v_β^* lies on the boundary of the constraint set of (4.7) and is $\mathcal{O}(\alpha^4)$-"close" to points of the null space $\mathcal{N}_\beta$ of B_β.

The null space $\mathcal{N}_\beta$ intersects the boundary of the constraint set of (4.7) twice. Let $\tilde{v}_\beta^*$ be the intersection that is close to v^*. Then $\tilde{v}_\beta^*$ satisfies

$$\tilde{v}_\beta^* = v_\beta^* + \mathcal{O}(\alpha^3)$$

and is given by

$$\tilde{v}_\beta^* = \begin{pmatrix} \sin(2\beta)-\sin(\alpha) \\ \cos(2\beta)+\cos(\alpha) \end{pmatrix} = x(2\beta)-x(\alpha).$$

Thus, for a sequence β_k the points

$$x(\alpha_{k+1}) = x(2\beta_k) + \mathcal{O}(\alpha_k^3)$$

converge superlinearly to $x(0)$ if and only if the angles β_k converge superlinearly to zero, too.

Summarizing, the method is quadratically convergent if we choose $\beta_k=\frac{\alpha_k^2}{2}$ and accordingly c_{β_k} as well as $B_k := \frac{1}{\alpha_k^4} c_{\beta_k} c_{\beta_k}^{\mathrm{T}}$ with $\alpha_k := \arcsin(x_1^k)$.

The main advantage of the augmented Lagrangian over other penalty functions is the fact that under standard assumptions a finite value for the barrier parameter is sufficient to guarantee exactness. In the above analysis, however, the matrices B_k are unbounded! In the next section we show that we cannot obtain superlinear convergence when B_k is bounded.

5. Linear convergence for any choice of bounded positive semidefinite *B*. In this section we prove the following statement.

PROPOSITION 5.1. *Suppose problem* (2.1) *is solved with the SLCP method where the Hessian approximations are given by* any *globally bounded sequence of positive semidefinite matrices. Then it is not possible to have a faster than linear convergence.*

Proof. We assume from now on that the positive semidefinite matrix $\breve{B} = \breve{B}_k$ for the SLCP subproblem (3.1) is bounded independently of k, $\|\breve{B}\|_2 \leq M$, and prove by contradiction that for any such $\breve{B}$ the rate of convergence cannot be better than $\alpha_{k+1} = \frac{\alpha_k}{2} + \mathcal{O}(\alpha_k^2)$.

We denote the solution of the SLCP subproblem (3.1) by $\breve{v}$ and use v^* as defined in (4.4) for the vector to the global optimum of (2.1).

If $x + \breve{v}$ is not on the unit circle, i.e., the boundary of the feasible set, it follows that $\breve{B}\breve{v} = -c$. Since $\breve{v} \to 0$ and $\|c\|_2 \to 4$ this implies $\|\breve{B}\|_2 \to \infty$ in contradiction to our assumption $\|\breve{B}\|_2 \leq M$. Thus we can assume without loss of generality that $x + \breve{v}$ is on the unit circle and use again the notation

$$x(\alpha) = \begin{pmatrix} \sin(\alpha) \\ -\cos(\alpha) \end{pmatrix}$$

with α as in (3.3).

Let $x(\alpha)$ be the kth iterate and let $x(\gamma)$ be the $(k+1)$st iterate; thus we have $\breve{v} = x(\gamma) - x(\alpha)$. We assume for contradiction that

$$0 \leq \gamma \leq \frac{\alpha}{2} - \varepsilon\alpha + \mathcal{O}(\alpha^2)$$

with $0 < \varepsilon \leq \frac{1}{2}$ independent of α. (The SLCP method is quadratically convergent if and only if $\varepsilon = \frac{1}{2}$.) As in (4.3) the optimal solution of the linear SLCP subproblem (3.1) using $B = 0$ is denoted by v.

As $\breve{v}$ is the optimal solution of the SLCP subproblem (3.1) using $B = \breve{B}$, the objective value of v is greater than the objective value of $\breve{v}$:

$$\begin{aligned} c^{\mathrm{T}} v + \frac{1}{2} v^{\mathrm{T}} \breve{B} v \geq c^{\mathrm{T}} \breve{v} + \frac{1}{2} \breve{v}^{\mathrm{T}} \breve{B} \breve{v} \Leftrightarrow \\ c^{\mathrm{T}}(v - \breve{v}) \geq \frac{1}{2}(\breve{v} - v)^{\mathrm{T}} \breve{B}(\breve{v} - v) + v^{\mathrm{T}} \breve{B}(\breve{v} - v). \end{aligned} \tag{5.1}$$

In the following we will evaluate the terms of (5.1) up to $\mathcal{O}(\alpha^4)$ to show that (5.1) cannot be true.

First we analyze the linear term on the left-hand side of (5.1) and obtain from Figure 5.1

$$\begin{aligned} c^{\mathrm{T}}(v - \breve{v}) &= -\|c\|_2 \|v - \breve{v}\|_2 \cos\left(\frac{\pi}{2} - \frac{\alpha}{4} + \frac{\gamma}{2}\right) \\ &= -\|c\|_2 \|v - \breve{v}\|_2 \sin\left(\frac{\alpha}{4} - \frac{\gamma}{2}\right) \\ &= -\|c\|_2 2 \sin\left(\frac{\alpha}{4} - \frac{\gamma}{2}\right) \sin\left(\frac{\alpha}{4} - \frac{\gamma}{2}\right) \\ &= -8 \sin^2\left(\frac{\alpha}{4} - \frac{\gamma}{2}\right) + \mathcal{O}(\alpha^4) \\ &\leq -2\varepsilon^2\alpha^2 + \mathcal{O}(\alpha^4). \end{aligned}$$

For the evaluation of the right-hand side of (5.1) note that

$$\frac{1}{2}(\breve{v} - v)\breve{B}(\breve{v} - v) \geq 0$$

since $\breve{B}$ is positive semidefinite.

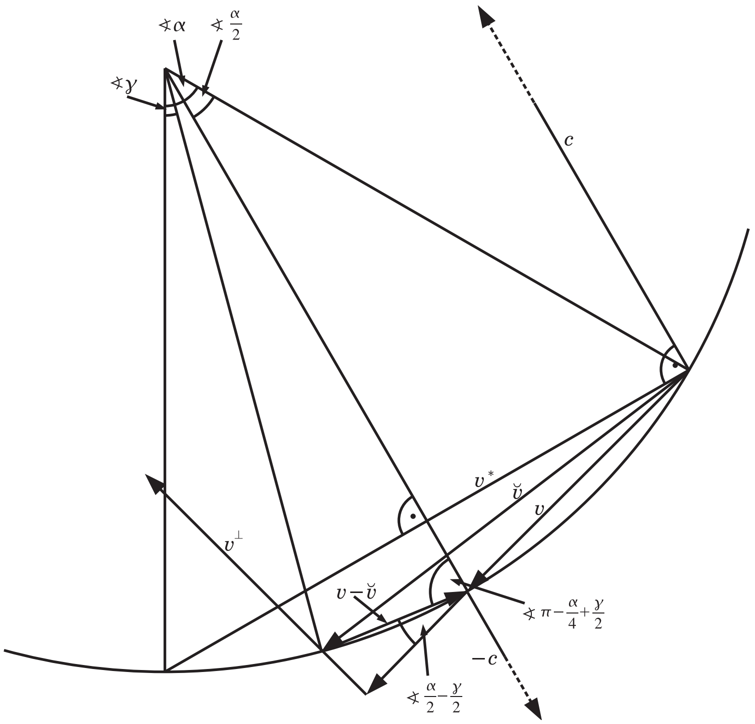

$\measuredangle \alpha$
$\measuredangle \frac{\alpha}{2}$
$\measuredangle \gamma$
c
v^*
$\breve{v}$
v
$v^\perp$
$v-\breve{v}$
$\measuredangle\, \pi-\frac{\alpha}{4}+\frac{\gamma}{2}$
$-c$
$\measuredangle\, \frac{\alpha}{2}-\frac{\gamma}{2}$

FIG. 5.1. *Angles and vectors used in the proof.*

Denote by $v^\perp$ the orthogonal complement of v with norm $\|v\|_2 = \|v^\perp\|_2$ that is obtained by a clockwise rotation of v by 90°. To evaluate the last term of the right-hand side of (5.1) note again by Figure 5.1 that

$$\breve{v} - v = \frac{\|\breve{v}-v\|_2}{\|v\|_2} v \cos\left(\frac{\alpha}{2} - \frac{\gamma}{2}\right) + \frac{\|\breve{v}-v\|_2}{\|v\|_2} v^\perp \sin\left(\frac{\alpha}{2} - \frac{\gamma}{2}\right).$$

Thus we have

$$\begin{aligned} v^{\mathrm{T}} \breve{B}(\breve{v}-v) &= \tfrac{\|\breve{v}-v\|_2}{\|v\|_2} \cos\left(\tfrac{\alpha}{2}-\tfrac{\gamma}{2}\right) v^{\mathrm{T}} \breve{B} v + \tfrac{\|\breve{v}-v\|_2}{\|v\|_2} \sin\left(\tfrac{\alpha}{2}-\tfrac{\gamma}{2}\right) v^{\mathrm{T}} \breve{B} v^\perp \\ &\geq 0 - \sin\left(\tfrac{\alpha}{2}-\tfrac{\gamma}{2}\right) \|\breve{v}-v\|_2 \|\breve{B}\|_2 \|v^\perp\|_2 \\ &\geq -\sin\left(\tfrac{\alpha}{2}\right) 2\sin\left(\tfrac{\alpha}{4}\right) M 2 \sin\left(\tfrac{\alpha}{4}\right) \\ &= -\tfrac{M}{8}\alpha^3 + \mathcal{O}(\alpha^5). \end{aligned}$$

From (5.1) it would therefore follow that

$$-2\varepsilon^2\alpha^2 + \mathcal{O}(\alpha^4) \geq c^{\mathrm{T}}(v-\breve{v}) \geq \tfrac{1}{2}(v-\breve{v})\breve{B}(v-\breve{v}) + v^{\mathrm{T}}\breve{B}\breve{v} \geq -\tfrac{M}{8}\alpha^3 + \mathcal{O}(\alpha^5),$$

i.e., $-2\varepsilon^2\alpha^2 + \mathcal{O}(\alpha^3) \geq 0$, which is not true for α sufficiently small and fixed $\varepsilon > 0$. Therefore, the assumption of faster than linear convergence has led to a contradiction. □

6. Consequences for semidefinite programming. Instead of using the Lorentz cone one can also use the semidefinite cone to formulate problem (2.1). The SLCP subproblems then have the form

$$\min \left\{ \begin{pmatrix} -2x_1 \\ 2-2x_2 \end{pmatrix}^{\mathrm{T}} \Delta x + \frac{1}{2}\Delta x^{\mathrm{T}} B \Delta x \;\middle|\; \begin{pmatrix} 1 & x_1+\Delta x_1 & 0 \\ x_1+\Delta x_1 & 1 & x_2+\Delta x_2 \\ 0 & x_2+\Delta x_2 & 1 \end{pmatrix} \succeq 0 \right\}.$$

Obviously these subproblems produce the same iterates. Thus linear convergence also follows for "sequential semidefinite programming methods" as considered, for example, in [1, 2] with bounded positive semidefinite B.

7. Conclusion. The constraint that a matrix X be positive semidefinite may be formulated as a nonlinear constraint that the determinant of X (and some principal minors) be nonnegative. As the optimal solution of a (nonlinear) program with a semidefiniteness constraint typically results in a solution X for which zero is a multiple eigenvalue, the constraint $\det(X) \geq 0$ is degenerate at the optimal solution. On the other hand, requiring that each of the eigenvalues of X be nonnegative typically yields a nondegenerate but also nonsmooth formulation. One may therefore be tempted to solve a nonlinear semidefinite program by an SQP-type method where the semidefiniteness constraint is maintained in the SLCP subproblem and the Hessian is given, for example, by the Hessian of the augmented Lagrangian or a semidefinite approximation thereof. If the quadratic objective function of such an SLCP subproblem is convex, these subproblems can be reformulated as linear semidefinite programs and can be solved efficiently. The example in this paper, however, shows that such a modified SLCP method cannot be superlinearly convergent in general contrasting the case of standard SQP methods.

The reason for the superiority of a standard SQP method over the presented SLCP method (both with positive definite Hessian approximations) can be seen in the fact that in the standard SQP subproblems all curvature information of the problem is collected in a single entity, namely, the approximation of the Hessian of the Lagrangian, and this matrix is positive definite on the null space of the linearized active constraints at any point satisfying the second order sufficient optimality conditions (and those are the points we are interested in). In the example, the negative curvature of the objective is more than counterweighted by the positive curvature of the constraint, and convex SQP subproblems arise naturally that need no Hessian modification. In the case of a standard SQP method for a general NLP the true Hessian needs only to be modified in the space orthogonal to the null space of the active constraints in order to keep the Hessian approximation positive definite, and the SQP steps are not affected by such Hessian modifications.

In contrast to this, the SLCP method uses curvature information at two places, separate in the objective and in the conic constraints. While this seems to be an advantage at first glance, it makes the Hessian of the Lagrangian independent of the conic constraints and therefore it can no longer profit from their positive curvature. Thus, the Hessian of the Lagrangian does not satisfy a similar positive definiteness condition as in the case of a standard NLP formulation; cf. [5]. If we desire a convex SLCP subproblem, we therefore need to change the Hessian of the Lagrangian considerably. The contribution of this paper was to show that superlinear convergence of the SLCP method cannot generally be enforced by bounded Hessian modifications. While this paper does not make any implications about the global behavior of SLCP methods for the solution of nonconvex problems with conic constraints, the observation made here implies that SLCP methods need to be modified to achieve rapid local convergence.

Acknowledgments. The authors are thankful to both referees for their constructive criticism that helped to improve this note. We also thank Dominikus Noll for pointing out to us the no-gap optimality conditions for SDP.

REFERENCES

[1] R. Correa and C. H. Ramirez, *A global algorithm for nonlinear semidefinite programming*, SIAM J. Optim., 15 (2004), pp. 303–318.

[2] B. Fares, D. Noll, and P. Apkarian, *Robust control via sequential semidefinite programming*, SIAM J. Control Optim., 40 (2002), pp. 1791–1820.

[3] F. Jarre, *On an Approximation of the Hessian of the Lagrangian*, http://www.optimization-online.org/DB_HTML/2003/12/800.html (2003).

[4] M. J. D. Powell, *The convergence of variable metric methods for nonlinearly constrained optimization calculations*, in Nonlinear Programming 3, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, New York, 1978, pp. 27–63.

[5] A. Shapiro, *First and second order analysis of nonlinear semidefinite programs*, Math. Programming Ser. B, 77 (1997), pp. 301–320.

[6] J. F. Sturm, *Using SeDuMi* 1.02*, a MATLAB toolbox for optimization over symmetric cones*, Optim. Methods Softw., 11–12 (1999), pp. 625–653.

[7] R. H. Tutuncu, K. C. Toh, and M. J. Todd, *Solving semidefinite-quadratic-linear programs using SDPT*3, Math. Program. Ser. B, 95 (2003), pp. 189–217.

SIAM J. Optim.
Vol. 16, No. 4, pp. 1211–1229

POLYNOMIAL CONVERGENCE OF INFEASIBLE-INTERIOR-POINT METHODS OVER SYMMETRIC CONES[*]

BHARATH KUMAR RANGARAJAN[†]

Abstract. We establish polynomial-time convergence of infeasible-interior-point methods for conic programs over symmetric cones using a wide neighborhood of the central path. The convergence is shown for a commutative family of search directions used in Schmieta and Alizadeh [*Math. Program.* 96 (2003), pp. 409–438]. Monteiro and Zhang [*Math. Program.*, 81 (1998), pp. 281–299] introduced this family of directions when analyzing semidefinite programs. These conic programs include linear and semidefinite programs. This extends the work of Rangarajan and Todd [*Tech. rep.* 1388, School of OR & IE, Cornell University, Ithaca, NY, 2003], which established convergence of infeasible-interior-point methods for self-scaled conic programs using the NT direction. Our work is built on earlier analyses by Faybusovich [*J. Comput. Appl. Math.*, 86 (1997), pp. 149–175] and Schmieta and Alizadeh [*Math. Program.* 96 (2003), pp. 409–438]. Of independent interest, we provide a constructive proof of Lyapunov lemma in the Jordan algebraic setting.

Key words. conic programming, infeasible-interior-point methods, symmetric cones, Euclidean Jordan algebras, semidefinite programming

AMS subject classifications. 90C25, 90C51, 65Y20, 90C22, 90C05

DOI. 10.1137/040606557

1. Introduction.

There is extensive literature on the analysis of interior-point methods (IPMs) for conic programming. In conic programs, a linear function is minimized over the intersection of an affine space and a closed convex cone. The foundation for solving these problems using IPMs was laid by Nesterov and Nemirovskii [8]. These methods were primarily either primal or dual based. Later, Nesterov and Todd [9] introduced symmetric primal-dual interior-point algorithms on a special class of cones called self-scaled cones, which allowed a symmetric treatment of the primal and the dual. Self-scaled cones are precisely the same as symmetric cones, which have been characterized using Jordan algebras (see Güler [3] and also Faraut and Koranyi [1]). Faybusovich [2] analyzed an interior-point algorithm over the symmetric cones using this characterization of symmetric cones.

Nonnegative orthants, second-order cones, and positive semidefinite cones are important special cases of symmetric cones. Monteiro and Zhang [7] gave a unified analysis of feasible-IPMs for semidefinite programs that used the so-called commutative class of search directions. These search directions include the popular directions such as the NT (Nesterov–Todd), the XS, and the SX directions. As we shall see, symmetric cones, when described using Jordan algebras, bear a striking resemblance to the cone of real symmetric positive semidefinite matrices. This resemblance was exploited by Schmieta and Alizadeh [11], who extended Monteiro and Zhang's analysis to feasible-IPMs over symmetric cones.

[*] Received by the editors April 9, 2004; accepted for publication (in revised form) November 18, 2005; published electronically March 24, 2006. This research was supported in part by NSF through grant DMS-0209457 and ONR through grant N00014-02-1-0057.

http://www.siam.org/journals/siopt/16-4/60655.html

[†] School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY 14853. Current Address: Department of Mechanical Engineering, University of Minnesota, 111 Church Street SE, Minneapolis, MN 55455 (ranga013@umn.edu).

Infeasible-IPMs, unlike feasible-IPMs, do not require that the iterates be feasible to the relevant linear systems but only be in the interior of the cone constraints. As such infeasible points are easy to obtain, infeasible-IPMs are an attractive choice for practical implementations. At the same time, the analysis of infeasible-IPMs presents significant difficulties due to the nonorthogonality of search directions. Zhang [14] analyzed the convergence of an infeasible-interior-point algorithm for semidefinite programming using the XS and SX search directions. Rangarajan and Todd [10] established convergence of an infeasible-IPM for self-scaled cones using the NT direction for a wide neighborhood of the central path. The results in [10] that deal with the nonorthogonality of search directions are adapted from those in Zhang [14].

In this paper, we prove the polynomial convergence of an infeasible-IPM on symmetric cones for the commutative class of search directions, which includes XS, SX, and the NT directions. In the process we give a constructive proof of the Lyapunov lemma in this setting (this is used in proving Lemma 3.5). To our knowledge this is the first time an infeasible-IPM has been analyzed for the NT method using the $\mathcal{N}_{-\infty}$ neighborhood for semidefinite programming (which is a special case of our framework). The complexity result obtained here for symmetric cones using the NT direction compares with the best bound obtained for linear programs (Zhang [13]). Besides the work of Schmieta and Alizadeh, our main tool in this paper is the analysis of an infeasible-IPM for self-scaled conic programming in Rangarajan and Todd [10].

This paper is organized as follows: We start with an introduction to the theory of Jordan algebras. Next we outline the basics of interior-point theory that leads to the algorithm and present its analysis. We present some conclusions in the final section.

2. Euclidean Jordan algebras. Characterization of symmetric cones using Jordan algebras (see Theorem 2.3) forms the fundamental basis for our analysis. This section covers the basic results in Jordan algebras, closely following Schmieta and Alizadeh [11] in presentation. For a comprehensive treatment of Jordan algebras, the reader is referred to Faraut and Koranyi [1]. For the purposes of illustration, we use the space of real symmetric matrices, which yields the cone of positive semidefinite matrices. In this case, the analysis in section 3 specializes to the case of semidefinite programming.

DEFINITION 2.1. *Let $\mathcal{J}$ be an n-dimensional vector space over the field of real numbers along with the bilinear map $\bullet : (x, y) \mapsto x \bullet y \in \mathcal{J}$. Then $(\mathcal{J}, \bullet)$ is a Euclidean Jordan algebra with identity if for all $x, y \in \mathcal{J}$*

1. $x \bullet y = y \bullet x$ (commutativity)*;*
2. $x \bullet (y \bullet x^2) = (x \bullet y) \bullet x^2$*, where* $x^2 = x \bullet x$ (Jordan identity)*;*
3. *there exists a symmetric positive definite quadratic form $\mathcal{Q}$ on $\mathcal{J}$ such that $\mathcal{Q}(x \bullet y, z) = \mathcal{Q}(x, y \bullet z)$;*
4. *there exists an identity element $e \in \mathcal{J}$, i.e., e such that $e \bullet x = x \bullet e$ for all $x \in \mathcal{J}$.*

DEFINITION 2.2. *If $\mathcal{J}$ is a Euclidean Jordan algebra, then its cone of squares is the set*

$$\mathcal{K}(\mathcal{J}) := \{x^2 : x \in \mathcal{J}\}.$$

Let $G(\mathcal{K})$ denote the group of automorphisms of a cone $\mathcal{K}$. $\mathcal{K}$ is a homogeneous cone if $G(\mathcal{K})$ acts on it transitively. That is, if $x, y \in \operatorname{int} \mathcal{K}$, then there exists $g \in G(\mathcal{K})$ such that $g(x) = y$. *Symmetric cones* are cones that are self-dual and homogeneous. Symmetric cones are also precisely the class of self-scaled cones introduced by Nesterov and Todd in [9] (see also Faybusovich [2] and Güler [3]). The following theorem relates symmetric cones and Euclidean Jordan algebras (for a proof, see Theorems III.2.1 and III.3.1 in Faraut and Koranyi [1]).

THEOREM 2.3 (Jordan algebraic characterization of symmetric cones). *A cone is symmetric iff it is the cone of squares of some Euclidean Jordan algebra.*

Example. Let $\mathcal{J} = \mathcal{S}^n$, the space of real symmetric matrices with the operation $X \bullet Y := \frac{XY+YX}{2}$ for $X, Y \in \mathcal{S}^n$. We can choose $\mathcal{Q}(X,Y) := \text{Trace}\,(XY)$ and e to be the identity matrix. Then $(\mathcal{J}, \bullet\,)$ is a Euclidean Jordan algebra with identity. We obtain the cone of symmetric positive semidefinite matrices as the squares of real symmetric matrices.

Since $\bullet$ is a bilinear map, for every $x \in \mathcal{J}$ a linear operator $L(x)$ can be defined such that $L(x)y = x \bullet y$ for all $y \in \mathcal{J}$. For $x, y \in \mathcal{J}$, let

$$Q_{x,y} := L(x)L(y) + L(y)L(x) - L(x \bullet y) \quad \text{and} \quad Q_x := Q_{x,x} = 2L^2(x) - L(x^2),$$

where Q_x is called the quadratic representation of x. Clearly $Q_{x,y}z$ and $Q_x z$ are in $\mathcal{J}$ for all $x, y, z \in \mathcal{J}$. We will use $\mathcal{K}$ to denote $\mathcal{K}(\mathcal{J})$ henceforth, when no confusion can arise.

Example. For $X \in \mathcal{S}^n$, $L(X)$ is the operator from $\mathcal{S}^n$ to itself such that $L(X)[Y] = \frac{XY+YX}{2}$. A further computation shows that $Q_{X,Y}[Z] = \frac{XZY+YZX}{2}$ and $Q_X[Z] = XZX$. The operator Q_X plays an important role in the analysis of IPMs for semidefinite programming. The operator Q_x in Jordan algebras plays a similar role in our analysis.

An element $x \in \mathcal{J}$ is called *invertible* if there exists a $y = \sum_{i=0}^{k} \gamma_i x^i$ for some finite $k < \infty$ and real numbers γ_i such that $y \bullet x = e$, and is written x^{-1}. The following are some of the basic properties of Q_x (see Propositions II.3.1 and II.3.3 in [1]).

LEMMA 2.4. *Let $x, y \in \mathcal{J}$. Then*

1. $Q_x x^{-1} = x$ *(or equivalently* $Q_x L(x^{-1}) = L(x)$*),* $Q_x^{-1} = Q_{x^{-1}}$*, and* $Q_x e = x^2$*;*
2. $Q_{Q_y x} = Q_y Q_x Q_y$.

With each Jordan algebra is associated a characteristic called the rank. We define rank of a Jordan algebra next.

DEFINITION 2.5.

a. *For $x \in \mathcal{J}$, let r be the smallest integer such that the set $\{e, x, x^2, \ldots, x^r\}$ is linearly dependent. Then r is called the degree of x and is denoted by* $\deg(x)$.
b. *The rank of $\mathcal{J}$, denoted by* $\text{rank}\,(\mathcal{J})$*, is defined as the maximum of* $\deg(x)$ *over all $x \in \mathcal{J}$. An element is called regular if its degree equals the rank of the Jordan algebra.*

One of the most important tools that help us in working with Jordan algebras is the spectral decomposition theorem. Towards this end, we discuss the concept of Jordan frames and introduce the result on spectral decomposition.

An *idempotent* c is a nonzero element of $\mathcal{J}$ such that $c^2 = c$. A complete system of orthogonal idempotents is a set $\{c_1, \ldots, c_k\}$ of idempotents, where $c_i \bullet c_j = 0$ for all $i \neq j$, and $c_1 + \cdots + c_k = e$. An idempotent is *primitive* if it is not the sum of two other idempotents. A complete system of orthogonal primitive idempotents is called a *Jordan frame*. Note that in Jordan frames $k = r$; that is, Jordan frames always contain r primitive idempotents.

THEOREM 2.6 (spectral decomposition, Theorem III.1.2 in [1]). *Let $\mathcal{J}$ be a Euclidean Jordan algebra. For $x \in \mathcal{J}$ there exist a Jordan frame $c_1, \ldots, c_r$ and real numbers $\lambda_1, \ldots, \lambda_r$ such that $x = \lambda_1 c_1 + \cdots + \lambda_r c_r$, where the λ_i's are called the eigenvalues of x.*

Using this we can define the following:

1. The square root: $x^{1/2} := \lambda_1^{1/2} c_1 + \cdots + \lambda_r^{1/2} c_r$ whenever all $\lambda_i \geq 0$, and undefined otherwise.
2. The inverse: $x^{-1} := \lambda_1^{-1} c_1 + \cdots + \lambda_r^{-1} c_r$ whenever all $\lambda_i \neq 0$, and undefined otherwise. (This is consistent with our earlier definition by Proposition II.2.4 in [1].)
3. The square: $x^2 := \lambda_1^2 c_1 + \cdots + \lambda_r^2 c_r$. This is consistent with the definition of x^2 as $x \bullet x$.

Note that $(x^{1/2})^2 = x$. It can be shown that an element is in (the interior of) the cone of squares iff all its eigenvalues are nonnegative (positive). The eigenvalues of e are all equal to one.

Definition 2.7. *Let $x \in \mathcal{J}$ and $\lambda_1, \ldots, \lambda_r$ be its eigenvalues. Then*

1. $\text{Trace}\,(x) := \lambda_1 + \cdots + \lambda_r$ *is called the trace of x;*
2. $\text{Det}\,(x) := \lambda_1, \ldots, \lambda_r$ *is called the determinant of x.*

Trace can be shown to be a linear function of x. For the identity element, $\text{Trace}\,(e) = r$ and $\text{Det}\,(e) = 1$ as all its eigenvalues are equal to one.

Next, norms and inner products are defined on $\mathcal{J}$. Since $\text{Trace}\,(x \bullet y)$ is a bilinear function, the inner product can be defined as $\langle x, y \rangle := \text{Trace}\,(x \bullet y)$. For $x \in \mathcal{J}$, with eigenvalues λ_i, $1 \leq i \leq r$, we can define the Frobenius norm (see Proposition III.1.5 in [1]) satisfying the Cauchy–Schwarz inequality:

$$\|x\|_F := \sqrt{\sum_{i=1}^{r} \lambda_i^2} = \sqrt{\text{Trace}\,(x^2)} \quad \text{and} \quad |\langle x, y \rangle| \leq \|x\|_F \|y\|_F.$$

As all the eigenvalues of e are equal to one, $\|e\|_F = \sqrt{r}$.

Example. For a matrix $X \in \mathcal{S}^n$, we have the spectral decomposition that there exists a set of orthonormal vectors $\{q_i, 1 \leq i \leq n\} \subset \Re^n$ and real numbers $\lambda_1, \ldots, \lambda_n$ such that $X = \sum_i \lambda_i q_i q_i^T$. It can be checked that the matrices $q_i q_i^T$ form a primitive system of orthogonal idempotents. The inner product is the usual trace inner product of matrices and the Frobenius norm has its usual definition.

Since $\text{Trace}\,(\cdot, \cdot)$ is associative (see Proposition II.4.3 in [1]), i.e., $\text{Trace}\,(x \bullet (y \bullet z)) = \text{Trace}\,((x \bullet y) \bullet z)$,

$$\langle L(x)p, q \rangle = \text{Trace}\,((x \bullet p) \bullet q) = \text{Trace}\,((p \bullet x) \bullet q) = \text{Trace}\,(p \bullet (x \bullet q)) = \langle p, L(x)q \rangle$$

shows that $L(x)$ is a self-adjoint operator. As the definition of Q_x depends only on $L(x)$ and $L(x^2)$, both of which are self-adjoint, Q_x is also self-adjoint.

We recall parts of Lemmas 12, 13, and 14 in [11] in the next two lemmas.

Lemma 2.8. *Let $x = \lambda_1 c_1 + \cdots + \lambda_r c_r$, using the spectral decomposition. Then the following statements hold.*

1. *The matrices $L(x)$ and Q_x commute and thus share a common system of eigenvectors.*
2. *Every eigenvalue of $L(x)$ can be written as $\frac{\lambda_i + \lambda_j}{2}$ for some $i,\ j \leq r$. In particular, $x \in \mathcal{K}$ (int $\mathcal{K}$) iff $L(x)$ is positive semidefinite (definite). The eigenvalues of x (λ_i) are amongst the eigenvalues of $L(x)$.*
3. *Every eigenvalue of Q_x can be written as $\lambda_i \lambda_j$, for some $i,\ j \leq r$. The square of the eigenvalues of x (λ_i^2) are amongst the eigenvalues of Q_x.*

Henceforth the minimum (maximum) eigenvalue of x will be denoted by $\lambda_{\min}(x)$ ($\lambda_{\max}(x)$).

Lemma 2.9. *Let $x \in \mathcal{J}$; then we have*

$$\lambda_{\min}(x) = \min_{u \neq 0} \frac{\langle u, u \bullet x \rangle}{\langle u, u \rangle}.$$

For $x, y \in \mathcal{J}$, we have

$$\lambda_{\min}(x + y) \geq \lambda_{\min}(x) - \|y\|_F,$$
$$\|x \bullet y\|_F \leq \|x\|_F \|y\|_F.$$

Proof. For proofs of all but the last part, see Lemmas 13 and 14 in [11]. The last part follows from

$$\|x \bullet y\|_F = \|L(x)y\|_F \leq \|L(x)\| \|y\|_F \leq \|x\|_F \|y\|_F.$$

The first equality follows from the definition of $L(x)$, and $\|L(x)\|$ refers to the operator norm induced by $\|\cdot\|_F$. Since the $\|\cdot\|_F$ can be seen as the norm induced by the inner product $\langle \cdot, \cdot \rangle$, the operator norm coincides with the spectral norm. For the second inequality note that the spectral norm of a self-adjoint linear operator is $\|L(x)\| = \max_i |\lambda_i(L(x))|$. Using Lemma 2.8 we can see that $\max_i |\lambda_i(L(x))| = \max_i |\lambda_i(x)| \leq \|x\|_F$. □

We state two useful propositions about the operator Q_x.

Proposition 2.10 (Faraut and Koranyi [1, Proposition III.2.2]). *If $x, y \in$ int $\mathcal{K}$, then $Q_x y \in$ int $\mathcal{K}$. Furthermore, if $x \in$ int $\mathcal{K}$, then Q_x is an automorphism of the cone $\mathcal{K}$.*

Proposition 2.11. *Let $x, s \in$ int $\mathcal{K}$; then*

1. *$Q_{x^{1/2}} s$ and $Q_{s^{1/2}} x$ have the same spectrum;*
2. *if $p \in$ int $\mathcal{K}$ define $\tilde{x} := Q_p x$ and $\tilde{s} := Q_{p^{-1}} s$, then $Q_{x^{1/2}} s$ and $Q_{\tilde{x}^{1/2}} \tilde{s}$ have the same spectrum.*

Furthermore, Trace $(Q_{x^{1/2}} s) = \langle s, x \rangle$.

Proof. See Proposition 21 in [11] for proofs of 1 and 2. Using the self-adjointness of $Q_{x^{1/2}}$ we have

$$\operatorname{Trace}(Q_{x^{1/2}} s) = \operatorname{Trace}((Q_{x^{1/2}} s) \bullet e) = \langle Q_{x^{1/2}} s, e \rangle = \langle s, Q_{x^{1/2}} e \rangle = \langle s, x \rangle,$$

which completes the proof of the proposition. □

Now we are ready to state and prove the Lyapunov lemma for Euclidean Jordan algebras.

Lemma 2.12 (Lyapunov lemma for Euclidean Jordan algebras). *Suppose that $\mathcal{J}$ is a Euclidean Jordan algebra. If $x \in$ int $\mathcal{K}$, $w \in \mathcal{K}$, then there exists $s \in \mathcal{K}$ such that $x \bullet s = w$.*

Proof. Let us set $s = 2 \int_0^\infty Q_{v(t)} w \, dt$, where $x = \sum_{i=1}^r \lambda_i c_i$ is the spectral decomposition of x and $v(t) = \sum_{i=1}^r e^{-\lambda_i t} c_i$. Clearly $v(t) \in \mathcal{J}$ as $c_i \in \mathcal{J}$. By expanding using the spectral decomposition and integrating we obtain $s = 2 \sum_{i,j} \frac{1}{\lambda_i + \lambda_j} Q_{c_i, c_j} w$ and hence s is well defined and $s \in \mathcal{J}$. Observe that $v(t) \in$ int $\mathcal{K}$ as $e^{-\lambda_i t} > 0$ for all t and hence $Q_{v(t)}$ is an automorphism of $\mathcal{K}$. It follows that $Q_{v(t)} w \in \mathcal{K}$. For $u \in \mathcal{K}$, we have

$$\langle s, u \rangle = 2 \left\langle \int_0^\infty Q_{v(t)} w \, dt, u \right\rangle = 2 \int_0^\infty \left\langle Q_{v(t)} w, u \right\rangle \, dt \geq 0.$$

Consequently $s \in \mathcal{K}$. By Proposition II.3.4 in [1] $Q_{v(t)} = e^{-2tL(x)}$. Therefore,

$$\frac{d}{dt} Q_{v(t)} w = \frac{d}{dt} e^{-2tL(x)} w = -2L(x) e^{-2tL(x)} w = -2L(x) Q_{v(t)} w = -2x \bullet Q_{v(t)} w.$$

We can substitute for s in the desired equation and use the above identity to get

$$x \bullet s = 2 \int_0^\infty x \bullet Q_{v(t)} w \; dt = \int_0^\infty -\frac{d}{dt} \left(Q_{v(t)} w \right) \; dt = w,$$
$$\text{as } v(0) = e, \text{ and } v(\infty) = 0. \qquad \square$$

Sturm [12], along with extensions to many properties of symmetric matrices, provides an alternate proof of the Lyapunov lemma, though the proof is not constructive. Another tool that is very useful in the analysis of algorithms is the notion of operator commutativity for the elements of a Jordan algebra. The notion of operator commutativity is not to be confused with the commutativity of elements of the Jordan algebra.

Definition 2.13. *We say two elements x, y of a Jordan algebra $\mathcal{J}$ operator commute if $L(x)L(y) = L(y)L(x)$. In other words, x and y operator commute if for all z, $x \bullet (y \bullet z) = y \bullet (x \bullet z)$.*

Theorem 2.14 (Theorem 27 in [11]). *Let x and y be two elements of Euclidean Jordan algebra $\mathcal{J}$. Then x and y operator commute iff there is a Jordan frame $c_1, \dots, c_r$ such that $x = \sum_{i=1}^r \lambda_i c_i$ and $s = \sum_{i=1}^r \mu_i c_i$ for some λ_i, μ_i.*

A Jordan algebra is called *simple* if it cannot be represented as a direct sum of two Jordan algebras. Simple Jordan algebras have been classified into the following five cases and consequently we have a classification for symmetric cones (see Chapter V in [1]). This classification is due to Jordan, von Neumann, and Wigner [4].

Theorem 2.15 (Faraut and Koranyi [1, Chapter V]). *Let $\mathcal{J}$ be a simple Euclidean Jordan algebra. Then $\mathcal{J}$ is isomorphic to one of the following algebras, where for the matrix algebras, the operation is defined by $X \bullet Y = \frac{1}{2}(XY + YX)$:*

1. *the algebra $\mathcal{E}_{n+1}$, the algebra of quadratic forms in $\Re^{n+1}$ under the operation $x \bullet y = (x^T y; x_0 \bar{y} + y_0 \bar{x})$, where $x = (x_0; \bar{x}), y = (y_0; \bar{y}) \in \Re \times \Re^n$;*
2. *the algebra $(\mathcal{S}^n, \bullet)$ of $n \times n$ symmetric matrices;*
3. *the algebra $(\mathcal{H}_n, \bullet)$ of $n \times n$ complex Hermitian matrices;*
4. *the algebra $(\mathcal{Q}_n, \bullet)$ of $n \times n$ quaternion Hermitian matrices;*
5. *the exceptional Albert algebra, i.e., the algebra $(\mathcal{O}_3, \bullet)$ of 3×3 octonian Hermitian matrices.*

3. Algorithm and analysis.

3.1. Problem background. We begin with the problem statement and discuss some of the theory relevant to developing interior-point algorithms: the perturbed optimality conditions, central path, and the Newton systems that give rise to the commutative class of search directions. In the following subsection, we present the algorithm and analyze its convergence.

Let $\mathcal{J}$ be a Euclidean Jordan algebra of dimension n and rank r, and $\mathcal{K}$ be its cone of squares. Consider the following primal and associated dual problem.

Primal and Dual.

$$(P) \qquad \min\{\langle c, x\rangle : Ax = b, \; x \in \mathcal{K}\} \tag{3.1}$$

and

$$(3.2) \qquad (D) \qquad \max\{\langle b, y\rangle_Y : \ A^*y + s = c, \ s \in \mathcal{K}, \ y \in Y\},$$

where $c \in \mathcal{J}$ and $b \in Y$, a finite dimensional real vector space with an inner product $\langle \cdot, \cdot \rangle_Y$. Here A is a linear operator that maps $\mathcal{J}$ into Y. A^* is defined to be the linear operator that maps Y to $\mathcal{J}$ such that $\langle A^*y, x\rangle = \langle Ax, y\rangle_Y$ for all $x \in \mathcal{J}, y \in Y$.

We call $x \in \mathcal{K}$ primal feasible if $Ax = b$. Similarly $(s, y) \in \mathcal{K} \times Y$ is called dual feasible if $A^*y + s = c$. Let

$$\mathcal{F}^0(P) := \{x \in \mathcal{J} : \ Ax = b, \ x \in \text{int}\,\mathcal{K}\} \text{ and}$$
$$\mathcal{F}^0(D) := \{(s, y) \in \mathcal{J} \times Y : \ A^*y + s = c, \ s \in \text{int}\,\mathcal{K}\}$$

represent the interior feasible solutions of the primal and the dual. For the rest of the paper, we will assume that A is surjective, $\mathcal{F}^0(P) \neq \emptyset$, and $\mathcal{F}^0(D) \neq \emptyset$. For a given primal feasible x and dual feasible (s, y), $\langle s, x\rangle$ is called the duality gap due to the well-known relation

$$\langle b, y\rangle_Y - \langle c, x\rangle = \langle Ax, y\rangle_Y - \langle A^*y + s, x\rangle = \langle s, x\rangle \geq 0.$$

Since the iterates in our algorithm may not satisfy the linear constraints, $\langle s, x\rangle$ will be referred to as the *complementarity.* Let us note that $\langle s, x\rangle = 0$ for feasible (x, s, y) is sufficient for optimality. By Lemma 2.2 in [2], for $x, s \in \mathcal{K}$ $\langle s, x\rangle = 0$ is equivalent to $s \bullet x = 0$. Using our assumptions above, the optimality conditions for the primal and dual problems can be written as

$$(3.3) \qquad \begin{aligned} Ax &= b, \\ A^*y + s &= c, \\ s \bullet x &= 0, \\ x, s &\in \mathcal{K}, y \in Y, \end{aligned}$$

where $s \bullet x = 0$ is usually referred to as the complementary slackness condition.

The perturbed optimality conditions (PC_μ) are obtained by replacing $s \bullet x = 0$ in (3.3) with the "perturbed" complementary slackness condition, $s \bullet x = \mu e$ for $\mu > 0$. Interior-point algorithms follow the solutions to (PC_μ) as μ goes to zero. The perturbed optimality conditions have unique solutions for all positive μ, and these solutions form the so-called central trajectory (see [2]). Note that the duality gap of the solutions is proportional to μ, i.e., $\langle s, x\rangle = \text{Trace}\,(s \bullet x) = \mu\text{Trace}\,(e) = \mu r$. IPMs employ Newton's method to target the solution of $(PC_{\sigma\mu})$, where $\sigma \in (0, 1)$, (x, s, y) is the current iterate, and $\mu = \frac{\langle s,x\rangle}{r}$. Such algorithms are called primal-dual path-following algorithms; primal-dual, because the primal and the dual are treated symmetrically in the optimality conditions.

The following lemma motivates different, but equivalent, ways of forming the perturbed optimality conditions, thus leading to different Newton systems.

Lemma 3.1 (Lemma 28 in [11]). *Let x, s and p be in some Euclidean Jordan algebra $\mathcal{J}$, $x, s \in$ int $\mathcal{K}$ and p invertible. Then $s \bullet x = \mu\, e$ iff $Q_{p^{-1}}(s) \bullet Q_p(x) = \mu\, e$.*

Therefore for a scaling $p \in \text{int}\,\mathcal{K}$, (PC_μ) can be equivalently written as

$$\begin{aligned} \tilde{A}\tilde{x} &= b, \\ \tilde{A}^*y + \tilde{s} &= \tilde{c}, \\ \tilde{s} \bullet \tilde{x} &= \mu e, \\ \tilde{x}, \tilde{s} &\in \mathcal{K}, y \in Y, \end{aligned}$$

where $\tilde{x} = Q_p x$, $\tilde{s} = Q_{p^{-1}} s$, $\tilde{A} = AQ_{p^{-1}}$, and $\tilde{c} = Q_{p^{-1}} c$. We restrict our attention to the following set of scalings:

$$\mathcal{C}(x,s) := \{p : \ p \in \text{int}\, \mathcal{K} \text{ such that } Q_p(x) \text{ and } Q_{p^{-1}}(s) \text{ operator commute}\}.$$

Note that $p = e$ need not be in $\mathcal{C}(x,s)$. For $p = x^{-1/2}$ we get the xs-method, for $p = s^{1/2}$ we get the sx-method, and for the choice of $p = \left[Q_{x^{1/2}}(Q_{x^{1/2}}s)^{-1/2}\right]^{-1/2} = \left[Q_{s^{-1/2}}(Q_{s^{1/2}}x)^{1/2}\right]^{-1/2}$, we get the Nesterov–Todd (NT) method. The Newton equations corresponding to a scaling in $\mathcal{C}(x,s)$ are stated below.

Scaled Newton Equations.

$$\begin{aligned} \tilde{A}^*\triangle y + \triangle\tilde{s} &= \tilde{c} - \tilde{A}^* y - \tilde{s}, \\ \tilde{A}\triangle\tilde{x} &= b - \tilde{A}\tilde{x}, \\ \tilde{s} \bullet \triangle\tilde{x} + \triangle\tilde{s} \bullet \tilde{x} &= \sigma\mu e - \tilde{s} \bullet \tilde{x}. \end{aligned} \tag{3.4}$$

Though $\mathcal{C}(x,s)$ seems to be a restrictive class, it does include some of the most interesting choices of scalings.

Our algorithm will restrict the iterates to the following neighborhood, called the minus-infinity neighborhood, of the central path. For a given constant $\gamma \in [0,1]$

$$\mathcal{N}_{-\infty}(\gamma) := \{(x,s,y) \in \mathcal{K} \times \mathcal{K} \times Y : \ d_{-\infty}(x,s) \le \gamma\mu\}, \tag{3.5}$$

where

$$d_{-\infty}(x,s) := \mu - \lambda_{\min}(z), \ \mu = \frac{\langle s,x\rangle}{r} \quad \text{and} \quad z = Q_{x^{1/2}}s.$$

A few observations about z are in order. As $x^{1/2} \in \mathcal{K}$ and $Q_{x^{1/2}}$ is an automorphism of $\mathcal{K}$, $z \in \mathcal{K}$ and hence $\lambda_i(z)$ are nonnegative. By Proposition 2.11 $\langle s,x\rangle = \text{Trace}\,(z) = \sum_i \lambda_i(z)$. The neighborhood contains the central path and γ represents the size of the neighborhood as it can be shown that the set $\mathcal{N}_{-\infty}(0) \cap \left[\mathcal{F}^0(P) \times \mathcal{F}^0(D)\right]$ is exactly the central path and $\mathcal{N}_{-\infty}(1) \cap \left[\mathcal{F}^0(P) \times \mathcal{F}^0(D)\right] = \mathcal{F}^0(P) \times \mathcal{F}^0(D)$.

Now we discuss the symmetry and scale-invariance of the neighborhoods. By the first statement of Proposition 2.11, $Q_{x^{1/2}}s$ and $Q_{s^{1/2}}x$ have the same spectrum. Hence the centrality measure $d_{-\infty}(x,s)$ and the neighborhood $\mathcal{N}_{-\infty}$ are symmetric with respect to x and s, i.e., $d_{-\infty}(x,s) = d_{-\infty}(s,x)$.

Proposition 3.2. *The neighborhood is scaling invariant; that is, (x,s) is in the neighborhood iff $(\tilde{x},\tilde{s})$ is.*

Proof. Let $\tilde{z} := Q_{\tilde{x}^{1/2}}\tilde{s}$. By the second statement of Proposition 2.11 $\lambda(\tilde{z})$ is the same as $\lambda(z)$. Since $\langle \tilde{s},\tilde{x}\rangle = \left\langle Q_{p^{-1}}s, Q_p x\right\rangle = \langle s,x\rangle$, the result follows by substituting the expressions in the definition of $\mathcal{N}_{-\infty}(\gamma)$. □

Hence the scaling transformations are not just automorphisms of the cone but they also map the neighborhood to itself. As the definition of $\mathcal{N}_{-\infty}$ is independent of y, sometimes y in (x,s,y) is suppressed for convenience and we write (x,s) instead, but y should be clear from the context.

3.2. Algorithm and analysis of convergence. Having discussed the key elements needed for the algorithm, we describe the infeasible-IPM in detail. To keep the analysis self-contained we have mentioned the relevant results from Schmieta and Alizadeh [11] and when dealing with infeasibility of iterates, we have proved the extension of results from Zhang [14] and Rangarajan and Todd [10].

Algorithm IIPM.

1. Let $1 > \beta > \sigma > 0$, $\epsilon^* > 0$, $\gamma \in (0,1)$, $x_0 \in \operatorname{int} \mathcal{K}$, $y_0 \in Y$, and $s_0 \in \operatorname{int} \mathcal{K}$ be given such that $(x_0, s_0, y_0) \in \mathcal{N}_{-\infty}(\gamma)$. Set $k = 0$, $\phi_p^0 = 1$, and $\phi_d^0 = 1$.
2. Choose a $p \in \mathcal{C}(x_k, s_k)$ and form the corresponding scaled iterate. Solve for $(\triangle\tilde{x}_k, \triangle\tilde{s}_k, \triangle y_k)$ from the scaled Newton equations in (3.4) at $(\tilde{x}_k, \tilde{s}_k, y_k)$. Let $(\triangle x_k, \triangle s_k, \triangle y_k) = (Q_{p^{-1}}\triangle\tilde{x}_k, Q_p\triangle\tilde{s}_k, \triangle y_k)$.
3. Let $(x(\alpha), s(\alpha), y(\alpha)) := (x_k, s_k, y_k) + \alpha(\triangle x_k, \triangle s_k, \triangle y_k)$. Compute the largest step length
$$\begin{aligned}&\bar{\alpha}_k \in (0,1] \text{ such that for all } \alpha \in [0, \bar{\alpha}_k], (x(\alpha), s(\alpha), y(\alpha)) \in \mathcal{N}_{-\infty}(\gamma),\\ &\langle s(\alpha), x(\alpha)\rangle \geq \max(\phi_p^k, \phi_d^k)(1-\alpha)\langle s_0, x_0\rangle, \quad \text{and} \quad \langle s(\alpha), x(\alpha)\rangle\\ &\qquad \leq (1-(1-\beta)\alpha)\langle s_k, x_k\rangle.\end{aligned}$$
4. Choose a primal step length $\alpha_p^k > 0$ and a dual step length $\alpha_d^k > 0$ such that
$$\begin{aligned}(x_{k+1}, s_{k+1}, y_{k+1}) &:= (x_k + \alpha_p^k\triangle x_k, s_k + \alpha_d^k\triangle s_k, y_k + \alpha_d^k\triangle y_k) \in \mathcal{N}_{-\infty}(\gamma),\\ \langle s_{k+1}, x_{k+1}\rangle &\geq \max(\phi_p^k(1-\alpha_p^k), \phi_d^k(1-\alpha_d^k))\langle s_0, x_0\rangle, \text{ and}\\ \langle s_{k+1}, x_{k+1}\rangle &\leq (1-(1-\beta)\bar{\alpha}_k)\langle s_k, x_k\rangle.\end{aligned}$$
Set $\phi_p^{k+1} = \phi_p^k(1-\alpha_p^k)$ and $\phi_d^{k+1} = \phi_d^k(1-\alpha_d^k)$.
5. Increase k by 1. If $\langle s_k, x_k\rangle < \epsilon^*\langle s_0, x_0\rangle$, then STOP. Otherwise, repeat step 2.

On the choice of step lengths: if we choose $\alpha_p^k = \alpha_d^k = \bar{\alpha}_k$, all the conditions in Step 4 are satisfied. However, we are free to choose different step lengths as long we can make a comparable progress in the feasibility and complementarity while remaining inside the neighborhood.

Using the Newton equations we can show that ϕ_p^k and ϕ_d^k satisfy the relations

$$Ax_k - b = \phi_p^k(Ax_0 - b) \quad \text{and} \quad A^*y_k + s_k - c = \phi_d^k(A^*y_0 + s_0 - c), \tag{3.6}$$

and hence they represent the relative infeasibilities at (x_k, s_k, y_k). At every iterate we maintain the condition,

$$\langle s_k, x_k\rangle \geq \max(\phi_p^k, \phi_d^k)\langle s_0, x_0\rangle, \tag{3.7}$$

which ensures that the infeasibilities approach zero as the complementarity, $\langle s, x\rangle$, approaches zero. The following theorem forms the skeleton of the convergence argument and sets the agenda for the rest of the paper.

Theorem 3.3. *If $\bar{\alpha}_k \geq \alpha^*$ for all k for some $\alpha^* > 0$, then the infeasible-interior-point method* (IIPM) *will terminate in (x_k, s_k, y_k) such that $\|Ax_k - b\| \leq \epsilon^*\|Ax_0 - b\|$, $\|A^*y_k + s_k - c\| \leq \epsilon^*\|A^*y_0 + s_0 - c\|$, and $\langle s_k, x_k\rangle \leq \epsilon^*\langle s_0, x_0\rangle$ in $\mathcal{O}(\frac{1}{\alpha^*}\ln(\frac{1}{\epsilon^*}))$ iterations.*

Proof. All the conditions in step 3 of IIPM are satisfied for α^*. Since for each k, $\bar{\alpha}_k \geq \alpha^*$, if we choose $k = \lceil\frac{1}{(1-\beta)\alpha^*}\rceil\ln\left(\frac{1}{\epsilon^*}\right)$, then we have

$$\begin{aligned}\ln(\langle s_k, x_k\rangle) &\leq \ln\left(\langle s_{k-1}, x_{k-1}\rangle(1-\alpha^*(1-\beta))\right)\\ &\leq \ln\left(\langle s_0, x_0\rangle(1-\alpha^*(1-\beta))^k\right)\\ &\leq \ln(\langle s_0, x_0\rangle) - k\alpha^*(1-\beta)\\ &\leq \ln(\langle s_0, x_0\rangle) + \ln(\epsilon^*) = \ln(\epsilon^*\langle s_0, x_0\rangle).\end{aligned}$$

The first inequality follows from the decrease in complementarity condition, the second from the same applied inductively, and the third inequality from the identity $1+\xi \leq e^{\xi}$ for all $\xi > -1$. The fourth inequality follows from our assumption on k.

From condition (3.7), it follows that $\max(\phi_p^k, \phi_d^k) \leq \frac{\langle s_k, x_k \rangle}{\langle s_0, x_0 \rangle} \leq \epsilon^*$. Then (3.6) implies that

$$\|Ax_k - b\| \leq \epsilon^* \ \|Ax_0 - b\|, \quad \text{and} \quad \|A^* y_k + s_k - c\| \leq \epsilon^* \ \|A^* y_0 + s_0 - c\|. \qquad \square$$

In the rest of the paper, we prove that such a lower bound on α^* exists and establishes an estimate of the lower bound that leads to the polynomial convergence result for the IIPM. We split this task into three pieces. First, we show that the step size at any iterate can be bounded from below using quantities that depend on the size of the directions and the complementarity ((3.8), (3.9), and (3.10)). Next, we improve this bound to a constant which is independent of the iteration number (Proposition 3.7 and (3.11)). Using a (rather restrictive) assumption on the size of the optimal solutions, we make the final transition to the required polynomial convergence with the help of Theorem 3.3.

Before we proceed to bounding the step lengths, we establish a useful inequality on the minimum eigenvalues (Lemma 3.5). For simplicity, we will often write x, y, s, and $\bar{\phi}$ for x_k, y_k, s_k, and $\max(\phi_p^k, \phi_d^k)$, respectively. The indices should be clear from the context.

Let $(x, s, y) \in \mathcal{N}_{-\infty}(\gamma)$ and satisfy the feasibility condition (3.7). For a fixed $p \in \mathcal{C}(x, s)$, let $(\triangle\tilde{x}, \triangle\tilde{s}, \triangle y)$ be the direction computed in step 2 of the algorithm. We will use the following notation:

$$\begin{aligned}
&\tilde{x}(\alpha) = \tilde{x} + \alpha \triangle \tilde{x}, \quad \tilde{s}(\alpha) = \tilde{s} + \alpha \triangle \tilde{s}, \\
&x(\alpha) = x + \alpha \triangle x, \quad s(\alpha) = s + \alpha \triangle s, \\
&\tilde{\mu}(\alpha) = \mu(\tilde{x}(\alpha), \tilde{s}(\alpha)) = \frac{\langle \tilde{s}(\alpha), \tilde{x}(\alpha) \rangle}{r}, \quad \text{and} \quad \tilde{z}(\alpha) = Q_{\tilde{x}(\alpha)^{1/2}} \tilde{s}(\alpha).
\end{aligned}$$

As a word of caution, since p need not lie in $\mathcal{C}(x(\alpha), s(\alpha))$, $\tilde{x}(\alpha)$ and $\tilde{s}(\alpha)$ do not necessarily operator commute. We collect some basic properties of the scaled directions and the Newton system.

Lemma 3.4. *Given the Newton equations, the following identities hold:*

$$\begin{aligned}
\tilde{s}(\alpha) \bullet \tilde{x}(\alpha) &= (1-\alpha)\, \tilde{s} \bullet \tilde{x} + \alpha\, \sigma \mu e + \alpha^2 \, \triangle \tilde{s} \bullet \triangle \tilde{x}, \\
\langle \tilde{s}, \tilde{x} \rangle &= \langle s, x \rangle, \ \textit{and} \\
\tilde{\mu}(\alpha) &= \mu(1 - \alpha + \sigma\alpha) + \alpha^2 \frac{\langle \triangle s, \triangle x \rangle}{r}.
\end{aligned}$$

Proof. The first equality follows by direct expanding the third equation of the scaled Newton system. The second statement was proven in Proposition 3.2. The last equation follows straightforwardly from the first. $\square$

The following result is absolutely essential in obtaining the bounds on the step lengths. The proof of this result uses the Lyapunov lemma (Lemma 2.12) established earlier.

Lemma 3.5. *Let $(x, s) \in$ int $\mathcal{K} \times$ int $\mathcal{K}$. Then $\lambda_{\min}(s \bullet x) \leq \lambda_{\min}(z)$ and equality holds if x and s operator commute.*

Proof. The proof outline follows Lemma 30 in [11]. First, observe that $Q_{x^{1/2},x^{-1/2}} Q_{x^{1/2}} = L(x)$, because

$$\begin{aligned} Q_{x^{1/2},x^{-1/2}}Q_{x^{1/2}} &= Q_{x^{1/2}}(2L(x^{-1/2})L(x^{1/2}) - I) \\ &= 2(Q_{x^{1/2}}L(x^{-1/2}))L(x^{1/2}) - Q_{x^{1/2}} \\ &= 2L^2(x^{1/2}) - Q_{x^{1/2}} = L(x). \end{aligned}$$

Here, we have used the first statement of Lemma 2.4. As a result we have $Q_{x^{1/2},x^{-1/2}}z = Q_{x^{1/2},x^{-1/2}}Q_{x^{1/2}}s = x \bullet s$.

In Lemma 30 in [11], it is shown that Trace $(Q_{x^{1/2},x^{-1/2}}u) =$ Trace (u). Note that by Lemma 2.12 we know that $\mathcal{K} \subset L(x)(\mathcal{K}) = Q_{x^{1/2},x^{-1/2}}Q_{x^{1/2}}(\mathcal{K}) = Q_{x^{1/2},x^{-1/2}}(\mathcal{K})$, as $Q_{x^{1/2}}$ is an automorphism of $\mathcal{K}$. The result follows from the following two chains of relations:

$$\begin{aligned} \lambda_{\min}(s \bullet x) &= \min_{u \neq 0} \frac{\langle u, (s \bullet x) \bullet u\rangle}{\langle u, u\rangle} = \min_{\text{Trace}\,(u^2)=1} \left\langle u^2, s \bullet x\right\rangle \\ &= \min_{\text{Trace}\,(u^2)=1} \left\langle u^2, Q_{x^{1/2},x^{-1/2}}z\right\rangle \end{aligned}$$

$$\begin{aligned} \min_{\text{Trace}\,(u^2)=1} \left\langle u^2, Q_{x^{1/2},x^{-1/2}}z\right\rangle &= \min_{\text{Trace}\,(u^2)=1} \left\langle z, Q_{x^{1/2},x^{-1/2}}u^2\right\rangle \\ &\leq \min_{\text{Trace}\,(Q_{x^{1/2},x^{-1/2}}u^2)=1} \left\{\left\langle z, Q_{x^{1/2},x^{-1/2}}u^2\right\rangle : Q_{x^{1/2},x^{-1/2}}u^2 \in \mathcal{K}\right\} \\ &= \min\left\{\langle z, t\rangle : \text{Trace}\,(t) = 1, t \in Q_{x^{1/2},x^{-1/2}}(\mathcal{K})\right\} \\ &\leq \min\left\{\langle z, t\rangle : \text{Trace}\,(t) = 1, t \in \mathcal{K}\right\} \\ &= \min_{\text{Trace}\,(v^2)=1} \left\langle z, v^2\right\rangle = \lambda_{\min}(z). \end{aligned}$$

The equality when $\tilde{x}$ and $\tilde{s}$ operator commute is established in Lemma 30 in [11]. Hence the proof of the lemma is complete. □

As a consequence, using Proposition 2.11 and the definition of $\mathcal{N}_{-\infty}(\gamma)$, let us note that

$$\lambda_{\min}(\tilde{s} \bullet \tilde{x}) = \lambda_{\min}(\tilde{z}) = \lambda_{\min}(z) \geq (1-\gamma)\mu.$$

We find an interval for which $(x(\alpha), s(\alpha))$ lies in the neighborhood.

LEMMA 3.6. *Let* $\delta_x = \|\triangle\tilde{x}\|_F$ *and* $\delta_s = \|\triangle\tilde{s}\|_F$. *If* $(x, s) \in \mathcal{N}_{-\infty}(\gamma)$, *then* $(x(\alpha), s(\alpha)) \in \mathcal{N}_{-\infty}(\gamma)$ *for all* $0 \leq \alpha \leq \hat{\alpha}_1$, *where*

$$\hat{\alpha}_1 := \min\left\{1, \frac{\gamma\sigma\langle s, x\rangle}{2(r+1-\gamma)\delta_x\delta_s}\right\}. \tag{3.8}$$

Proof. We first bound the left- and right-hand side of the inequality defining the neighborhood $\mathcal{N}_{-\infty}(\gamma)$. To begin with a bound on the eigenvalue of $z(\alpha)$, we have

$$\begin{aligned} \lambda_{\min}(z(\alpha)) = \lambda_{\min}(\tilde{z}(\alpha)) &\geq \lambda_{\min}(\tilde{s}(\alpha) \bullet \tilde{x}(\alpha)) \\ &= \lambda_{\min}((1-\alpha)\tilde{s} \bullet \tilde{x} + \alpha\sigma\mu e + \alpha^2 \triangle\tilde{s} \bullet \triangle\tilde{x}) \\ &\geq (1-\alpha)\lambda_{\min}(\tilde{s} \bullet \tilde{x}) + \alpha\sigma\mu - \alpha^2\delta_x\delta_s \\ &\geq (1-\alpha)(1-\gamma)\mu + \alpha\sigma\mu - \alpha^2\delta_x\delta_s. \end{aligned}$$

The first equality follows from the second statement of Proposition 2.11, the first inequality follows from Lemma 3.5, the second inequality follows from Lemma 2.9, and the last inequality follows because $(\tilde{x}, \tilde{s}) \in \mathcal{N}_{-\infty}(\gamma)$. Using Lemma 3.4 and Cauchy–Schwarz we can see that

$$\begin{aligned}(1-\gamma)\mu(\alpha) &= (1-\gamma)\left(\mu(1-\alpha+\sigma\alpha)+\alpha^2\frac{\langle \triangle s, \triangle x\rangle}{r}\right)\\ &\leq (1-\gamma)\left[\mu(1-\alpha+\sigma\alpha)+\alpha^2\frac{\delta_x\delta_s}{r}\right].\end{aligned}$$

Using $\langle s, x\rangle = \mu r$, we can see that

$$(1-\alpha)(1-\gamma)\mu+\alpha\sigma\mu-\alpha^2\delta_x\delta_s \geq (1-\gamma)\left[\mu(1-\alpha+\sigma\alpha)+\alpha^2\frac{\delta_x\delta_s}{r}\right]$$

holds for all $\alpha \in [0, 2\hat{\alpha}_1]$. Since the right-hand side of the inequality is positive for all $\alpha \in [0,1]$, $\lambda_{\min}(z(\alpha)) > 0$ for all $\alpha \in [0, \hat{\alpha}_1]$. Let α_0 be the least $\alpha \leq \hat{\alpha}_1$ such that $x(\alpha), s(\alpha) \in \mathcal{K}$ for all $\alpha \leq \alpha_0$ and $x(\alpha_0) \in \partial\mathcal{K}$ (or $s(\alpha_0) \in \partial\mathcal{K}$). Then $\lambda_{\min}(z(\alpha_0)) = 0$, which is a contradiction. Hence $x(\alpha), s(\alpha) \in \operatorname{int} \mathcal{K}$. Hence $(x(\alpha), s(\alpha), y(\alpha)) \in \mathcal{N}_{-\infty}(\gamma)$ for all $\alpha \in [0, \hat{\alpha}_1]$. □

Note that the length of the interval obtained depends on the size of the scaled Newton directions.

For the feasibility condition in step 3 of IIPM we want an $\hat{\alpha}_2$ such that (3.7) holds for all $(x(\alpha), s(\alpha))$, $\alpha \in [0, \hat{\alpha}_2]$. Using Lemma 3.4, the feasibility condition on (x, s) and Cauchy–Schwarz, we get

$$\begin{aligned}\frac{\langle s(\alpha), x(\alpha)\rangle}{\langle s_0, x_0\rangle} - \bar{\phi}(1-\alpha) &= \frac{\langle s, x\rangle}{\langle s_0, x_0\rangle}(1+\alpha(\sigma-1)) + \alpha^2\frac{\langle \triangle s, \triangle x\rangle}{\langle s_0, x_0\rangle} - \bar{\phi}(1-\alpha)\\ &= \left(\frac{\langle s, x\rangle}{\langle s_0, x_0\rangle} - \bar{\phi}\right)(1-\alpha) + \alpha\sigma\frac{\langle s, x\rangle}{\langle s_0, x_0\rangle} + \alpha^2\frac{\langle \triangle s, \triangle x\rangle}{\langle s_0, x_0\rangle}\\ &\geq \frac{\alpha}{\langle s_0, x_0\rangle}(\sigma\langle s, x\rangle - \alpha\delta_x\delta_s).\end{aligned}$$

Therefore the condition $\langle s(\alpha), x(\alpha)\rangle - \bar{\phi}(1-\alpha)\langle s_0, x_0\rangle \geq 0$ holds for all $\alpha \in [0, \hat{\alpha}_2]$, where

$$\hat{\alpha}_2 := \frac{\sigma\langle s, x\rangle}{\delta_x\delta_s}. \tag{3.9}$$

For the last condition in Step 3, Cauchy–Schwarz yields

$$\begin{aligned}\langle s(\alpha), x(\alpha)\rangle &= \langle s, x\rangle(1-\alpha(1-\sigma)) + \alpha^2\langle \triangle s, \triangle x\rangle\\ &\leq \langle s, x\rangle\left(1-\alpha(1-\sigma)+\alpha^2\frac{\delta_x\delta_s}{\langle s, x\rangle}\right).\end{aligned}$$

It suffices to have

$$\left[1-\alpha(1-\sigma)+\alpha^2\frac{\delta_x\delta_s}{\langle s, x\rangle}\right] - (1-\alpha(1-\beta)) = \alpha\left(\alpha\frac{\delta_x\delta_s}{\langle s, x\rangle} - (\beta-\sigma)\right) \leq 0.$$

Solving for α from the above inequality, we can see that the last condition holds for all $\alpha \in [0, \hat{\alpha}_3]$, where

$$\hat{\alpha}_3 := \frac{(\beta-\sigma)\langle s, x\rangle}{\delta_x\delta_s}. \tag{3.10}$$

So far, we have obtained a lower bound on the step sizes in terms of δ_x, δ_s, and $\langle s, x\rangle$. Next we prove Proposition 3.7, where we obtain a constant upper bound (independent of the iteration number) on $\frac{\delta_x \delta_s}{\langle s,x\rangle}$, which appears in (3.8), (3.9), and (3.10). By Theorem 3.3 this constant upper bound leads to a global convergence result. For this we introduce the operator, $G := L(\tilde{s})^{-1}L(\tilde{x})$, which plays a useful role in bounding $\delta_x\delta_s$. Recall the third scaled Newton equation

$$L(\tilde{s})\triangle\tilde{x} + L(\tilde{x})\triangle\tilde{s} = \sigma\mu e - L(\tilde{s})L(\tilde{x})e.$$

Since $\tilde{x}$ and $\tilde{s}$ operator commute, and G is a symmetric matrix, by multiplying this equation by $(L(\tilde{x})L(\tilde{s}))^{-1/2}$, we get

$$G^{-1/2}\triangle\tilde{x} + G^{1/2}\triangle\tilde{s} = \sigma\mu(L(\tilde{x})L(\tilde{s}))^{-1/2}e - G^{1/2}\tilde{s} =: h.$$

The analysis of IIPM is intricate because $\left\langle G^{1/2}\triangle\tilde{s}, G^{-1/2}\triangle\tilde{x}\right\rangle = \langle\triangle s, \triangle x\rangle \neq 0$. Now let us define

$$t^2 := \|G^{1/2}\triangle\tilde{s}\|_F^2 + \|G^{-1/2}\triangle\tilde{x}\|_F^2.$$

The following proposition will lead to a bound on the size of $\frac{\delta_x\delta_s}{\langle s,x\rangle}$.

Proposition 3.7. *$t_k^2 \leq \omega\,\langle s_k, x_k\rangle$, where ω is a constant independent of k.*

We will drop the subscript k on t when we do not need to stress on the iteration number. Before we prove the proposition, let us pause here to see its relevance in bounding $\delta_x\delta_s$. We state the following technical but useful result (Lemma 33 in [11]).

Lemma 3.8. *Let $u, v \in \mathcal{J}$ and G be a positive definite self-adjoint operator. Then*

$$\|u\|_F\|v\|_F \leq \frac{1}{2}\sqrt{\kappa_G}\left(\|G^{1/2}u\|_F^2 + \|G^{-1/2}v\|_F^2\right),$$

where $\kappa_G = \frac{\lambda_{\max}(G)}{\lambda_{\min}(G)}$ is the condition number of G.

Note that in our application, κ_G may depend on the iteration number k, but the following lemma provides a bound on the condition number of G for the methods we are interested in (see Lemma 36 in [11]).

Lemma 3.9. *For the NT method $\kappa_G = 1 =: \kappa$. For the xs and the sx methods,*

$$\textit{if } (x,s) \in \mathcal{N}_{-\infty}(\gamma), \quad \textit{then} \quad \kappa_G \leq \frac{r}{1-\gamma} =: \kappa.$$

Using the above lemmas, we have the following bound on $\delta_x\delta_s$:

$$\delta_x\delta_s \leq \frac{t^2}{2}\sqrt{\kappa} \leq \frac{\omega}{2}\sqrt{\kappa}\,\langle s, x\rangle\,. \tag{3.11}$$

Now we prove the proposition.

Proof (Proposition 3.7). We first note the following identity:

$$\begin{aligned}\|G^{1/2}\triangle\tilde{s} + G^{-1/2}\triangle\tilde{x}\|_F^2 &= \|G^{1/2}\triangle\tilde{s}\|_F^2 + \|G^{-1/2}\triangle\tilde{x}\|_F^2 + 2\,\langle G^{1/2}\triangle\tilde{s}, G^{-1/2}\triangle\tilde{x}\rangle \\ &= \|G^{1/2}\triangle\tilde{s}\|_F^2 + \|G^{-1/2}\triangle\tilde{x}\|_F^2 + 2\,\langle\triangle\tilde{s}, \triangle\tilde{x}\rangle.\end{aligned}$$

Using what we just derived we can show that

$$t^2 + 2\,\langle\triangle\tilde{s}, \triangle\tilde{x}\rangle = \|h\|_F^2 = \sum_i^r \frac{(\sigma\mu - \lambda_i(\tilde{z}))^2}{\lambda_i(\tilde{z})} \leq \left(1 - 2\sigma + \frac{\sigma^2}{1-\gamma}\right)\langle s, x\rangle, \tag{3.12}$$

where the second equality follows from the proof of Lemma 34 in [11] and inequality from Lemma 35 in [11].

We take a small detour to introduce some convenient notation which helps us in stating a key claim in the proof of this proposition, and is also used in the arguments for polynomiality of convergence. Let us assume a reference point (u_0, v_0, r_0) feasible to the equality constraints (and not necessarily in the cone) such that $x_0 - u_0, s_0 - v_0 \in \operatorname{int} \mathcal{K}$, where (x_0, s_0, y_0) is the initial iterate in IIPM. This condition is easily satisfied by scaling the initial point for any given (u_0, v_0, r_0). For a given sequence of iterates $\{(x_k, s_k, y_k)\}$ we define

$$\begin{aligned} u_{k+1} &= (1-\alpha_p^k)(u_k - x_k) + x_{k+1}; \\ r_{k+1} &= (1-\alpha_d^k)(r_k - y_k) + y_{k+1}; \\ v_{k+1} &= (1-\alpha_d^k)(v_k - s_k) + s_{k+1}. \end{aligned}$$

From the above definitions, we can observe the following properties:

$$\begin{aligned} x_{k+1} - u_{k+1} &= \phi_p^{k+1}(x_0 - u_0) \in \operatorname{int} \mathcal{K}; \\ s_{k+1} - v_{k+1} &= \phi_d^{k+1}(s_0 - v_0) \in \operatorname{int} \mathcal{K}; \\ Au_k &= b \text{ and } A^* r_k + v_k = c \text{ for all } k; \\ A(x_k + \triangle x_k - u_k) &= A(x + \triangle x_k) - Au_k = b - b = 0; \\ A^*(y_k + \triangle y_k - r_k) + s_k + \triangle s_k - v_k &= 0. \end{aligned} \tag{3.13}$$

(The third line holds for $k = 0$ by assumption, and then holds for all k by induction using the last two lines.) The following result is the key to proving the proposition.

CLAIM 3.1.

$$\langle s, x\rangle \frac{\langle s_0 - v_0, x_0 - u_0\rangle}{\langle s_0, x_0\rangle} + \langle \triangle s, \triangle x\rangle + \xi t\sqrt{\langle s, x\rangle} \geq 0,$$

where

$$\xi := \sqrt{\frac{r}{1-\gamma}} \left[\frac{\langle s, x-u\rangle + \langle s-v, x\rangle}{\langle s, x\rangle}\right]. \tag{3.14}$$

Due to the technical nature of the claim, the proved is placed in the appendix. For now, we substitute $\langle \triangle s, \triangle x\rangle$ from the inequality in (3.12), to get

$$t^2 \leq \langle s, x\rangle \, \bar{\chi} + 2\sqrt{\langle s, x\rangle} \, \xi t,$$

where

$$\bar{\chi} := 1 - 2\sigma + \frac{\sigma^2}{1-\gamma} + 2\left\{\frac{\langle s_0 - v_0, x_0 - u_0\rangle}{\langle s_0, x_0\rangle}\right\} \text{ is independent of } k. \tag{3.15}$$

Therefore,

$$t_k^2 \leq \langle s_k, x_k\rangle \left(\xi_k + \sqrt{\xi_k^2 + \bar{\chi}}\right)^2.$$

From Lemma 4.1 in [10], we have the following useful bound: Let (x, s, y) be any iterate generated by IIPM and (x^*, s^*, y^*) be an optimal solution to (P) and (D). Then

$$\frac{\langle s, x-u\rangle + \langle s-v, x\rangle}{\langle s, x\rangle} \leq 1 + \frac{\langle s^*, x_0 - u_0\rangle + \langle s_0 - v_0, x^*\rangle + \langle s_0 - v_0, x_0 - u_0\rangle}{\langle s_0, x_0\rangle}.$$

Therefore ξ_k is uniformly bounded by $\bar{\xi}$, where

$$\bar{\xi} = \sqrt{\frac{r}{1-\gamma}}\left\{1 + \frac{\langle s^*, x_0 - u_0\rangle + \langle s_0 - v_0, x^*\rangle + \langle s_0 - v_0, x_0 - u_0\rangle}{\langle s_0, x_0\rangle}\right\}. \tag{3.16}$$

Hence we can choose ω to be

$$\omega = \left(\bar{\xi} + \sqrt{\bar{\xi}^2 + \bar{\chi}}\right)^2. \qquad \square \tag{3.17}$$

Recall that the conclusion of Proposition 3.7 led to a bound on $\delta_x\delta_s$ in (3.11). Hence we can bound from below the $\hat{\alpha}$'s in (3.8), (3.9), and (3.10) in the following way:

$$\hat{\alpha}_1 = \frac{\gamma\sigma\,\langle s, x\rangle}{2\,(r+1-\gamma)\,\delta_x\delta_s} \geq \frac{\gamma\sigma}{(r+1-\gamma)\,\omega\sqrt{\kappa}} =: \bar{\alpha}_1, \tag{3.18}$$

$$\hat{\alpha}_2 = \frac{\sigma\,\langle s, x\rangle}{\delta_x\delta_s} \geq \frac{2\sigma}{\omega\sqrt{\kappa}} =: \bar{\alpha}_2, \quad \text{and} \tag{3.19}$$

$$\hat{\alpha}_3 = \frac{(\beta-\sigma)\,\langle s, x\rangle}{\delta_x\delta_s} \geq \frac{2(\beta-\sigma)}{\omega\sqrt{\kappa}} =: \bar{\alpha}_3. \tag{3.20}$$

Taking into account the above bounds, we define

$$\alpha^* := \min\left(1, \frac{\gamma\sigma}{(r+1-\gamma)\,\omega\sqrt{\kappa}}, \frac{2\sigma}{\omega\sqrt{\kappa}}, \frac{2(\beta-\sigma)}{\omega\sqrt{\kappa}}\right) = \Omega\left(\frac{1}{r\omega\sqrt{\kappa}}\right). \tag{3.21}$$

For this choice of α^*, for $\alpha \in [0, \alpha^*]$ all the conditions in step 3 (and hence step 4 by the remarks following the algorithm) of IIPM are satisfied. This bound implies the global convergence of IIPM by Theorem 3.3. Also, note that since $\langle \triangle\tilde{s}, \triangle\tilde{x}\rangle = 0$ for feasible-IPMs, (3.12) implies that

$$t^2 \leq \left(1 - 2\sigma + \frac{\sigma^2}{1-\gamma}\right)\langle s, x\rangle .$$

Hence ω in the case of feasible-IPMs is replaced by a constant independent of the data and we obtain $\mathcal{O}(r\sqrt{\kappa}\ln(1/\epsilon))$ iteration complexity for feasible-IPMs by Theorem 3.3. This is the bound obtained by Schmieta and Alizadeh in [11].

With some restrictions on the size of initial points, we can show that ω is polynomially bounded and consequently obtains the polynomial convergence of IIPM. Let (u_0, r_0, v_0) be the solution to

$$\begin{gathered} \min\{\|u\|_F \,:\, Au = b\} \quad \text{and} \quad \min\{\|v\|_F \,:\, A^* r + v = c\}, \text{ and} \\ x_0 = s_0 = \rho_0 e \in \operatorname{int}\mathcal{K}, \end{gathered} \tag{3.22}$$

where e is the identity element of the Euclidean Jordan algebra and $\rho_0 > \max(\|u_0\|_2,\ \|v_0\|_2)$. This implies that $x_0 - u_0 \in \operatorname{int}\mathcal{K}$ and $s_0 - v_0 \in \operatorname{int}\mathcal{K}$. Let us assume that for some constant $\Psi > 0$,

$$\rho_0 \geq \frac{1}{\Psi}\rho^* := \frac{1}{\Psi}\min\{\max(\|x^*\|_2, \|s^*\|_2) : (x^*, s^*) \text{ solves } (P) \text{ and } (D)\}. \tag{3.23}$$

(Note that we can always increase ρ_0.) Now we can obtain a bound for ω. First, let us note two useful facts: $\|\cdot\|_F \le \sqrt{r}\|\cdot\|_2$ and $\langle s_0, x_0\rangle = \rho_0^2 r$. Therefore, using Cauchy–Schwarz, we can see that $\langle p, q\rangle \le \|p\|_F\|q\|_F \le r\|p\|_2\|q\|_2$. Now we can bound $\bar{\xi}$ in (3.16) as follows:

$$\begin{aligned}\bar{\xi} &= \sqrt{\frac{r}{1-\gamma}}\left\{1+\frac{\langle s^*, x_0-u_0\rangle + \langle s_0-v_0, x^*\rangle + \langle s_0-v_0, x_0-u_0\rangle}{\langle s_0, x_0\rangle}\right\}\\ &\le \sqrt{\frac{r}{1-\gamma}}\left\{1+\frac{2\rho^*\rho_0 r + 2\rho^*\rho_0 r + 4\rho_0^2 r}{\rho_0^2 r}\right\}\\ &= \sqrt{\frac{r}{1-\gamma}}\left\{5+4\frac{\rho^*}{\rho_0}\right\} \le \sqrt{\frac{r}{1-\gamma}}(5+4\Psi) \text{ (using (3.23))}.\end{aligned}$$

For a bound on $\bar{\chi}$ in (3.15), we have

$$\bar{\chi} = 1-2\sigma+\frac{\sigma^2}{1-\gamma}+2\left\{\frac{\langle s_0-v_0, x_0-u_0\rangle}{\langle s_0, x_0\rangle}\right\} \le 1+\frac{1}{1-\gamma}+2\cdot\frac{4\rho_0^2 r}{\rho_0^2 r} = 9+\frac{1}{1-\gamma}.$$

Therefore,

$$\omega = \left(\bar{\xi}+\sqrt{\bar{\xi}^2+\bar{\chi}}\right)^2 = O(r). \tag{3.24}$$

Having obtained bounds on the key quantities defining α^* in (3.21), we state our main theorem.

THEOREM 3.10. *Suppose that $\kappa_G \le \kappa < \infty$ for all iterations of* IIPM. *Then* IIPM *will terminate in $\mathcal{O}(\sqrt{\kappa}r^2\ln(1/\epsilon^*))$ iterations. Hence the* NT *method takes $\mathcal{O}(r^2\ln(1/\epsilon^*))$ iterations, and the xs and sx methods take $\mathcal{O}(r^{2.5}\ln(1/\epsilon^*))$ iterations.*

Proof. For any $\alpha \in [0, \alpha^*]$, α^* as defined in (3.21), all the conditions in step 3 of IIPM are satisfied. Thus by Theorem 3.3, IIPM will terminate in $k = \lceil\frac{1}{\alpha^*}\rceil\ln\left(\frac{1}{\epsilon^*}\right) = O\left(\sqrt{\kappa}r^2\ln(1/\epsilon^*)\right)$ iterations.

The second part of the theorem follows from the bound on κ in Lemma 3.9 for the xs, the sx, and the NT method. □

4. Appendix.

The following technical result is useful in proving Claim 3.1.

LEMMA 4.1. *If $G = L(\tilde{s})^{-1}L(\tilde{x})$, then $\lambda_{\max}(Q_{\tilde{x}}^{-1}G) = \frac{1}{\lambda_{\min}(\tilde{z})}$. If $q \in \mathcal{K}$ and $\tilde{q} = Q_{p^{-1}}q$, then*

$$\|Q_{\tilde{x}^{1/2}}\tilde{q}\|_F \le \langle\tilde{q}, \tilde{x}\rangle = \langle q, x\rangle.$$

Proof. Suppose $\{\lambda_i : 1 \le i \le r\}$ are the eigenvalues of $\tilde{x}$ with $\lambda_1 \ge \cdots \ge \lambda_i \ge \cdots \ge \lambda_r$ paired with the eigenvectors $\{c_i : 1 \le i \le r\}$ derived from the spectral decomposition. Let the corresponding eigenvalues of $\tilde{s}$ be $\{\mu_i : 1 \le i \le r\}$ with $\mu_1 \ge \cdots \ge \mu_i \ge \cdots \ge \mu_r$. $L(\tilde{x}^{-1})$ and $L(\tilde{s})^{-1}$ commute as operators as $\tilde{x}$ and $\tilde{s}$ operator commute. From section 4, Chapter V of [5] we can see that their eigenvalues come from a common index pair set $I \subset \{(i,j) : 1 \le i, j \le r\}$ (with $(i,i) \in I$ for $1 \le i \le r$) and they share the same eigenspace corresponding to the eigenvalue derived from a pair $(i,j) \in I$. Using Lemmas 2.4 and 2.8, and Theorem 2.14, we then have the following two results:

$$\begin{aligned}\lambda_{\max}\left(Q_{\tilde{x}}^{-1}L(\tilde{s})^{-1}L(\tilde{x})\right) &= \lambda_{\max}\left(Q_{\tilde{x}^{-1}}L(\tilde{x})L(\tilde{s})^{-1}\right)\\ &= \lambda_{\max}\left(L(\tilde{x}^{-1})L(\tilde{s})^{-1}\right)\\ &= \max_{(i,j)\in I}\left[\left(\frac{1}{\lambda_i}+\frac{1}{\lambda_j}\right)\frac{1}{\mu_i+\mu_j}\right],\end{aligned}$$

$$\lambda_{\min}(\tilde{z})^2 = \lambda_{\min}(Q_{\tilde{x}^{1/2}}\tilde{s})^2 = \lambda_{\min}(Q_{Q_{\tilde{x}^{1/2}}\tilde{s}}) = \lambda_{\min}\left(Q_{\tilde{x}^{1/2}}Q_{\tilde{s}}Q_{\tilde{x}^{1/2}}\right) = \lambda_{\min}\left(Q_{\tilde{s}}Q_{\tilde{x}}\right),$$

and $\lambda_{\min}\left(Q_{\tilde{s}}Q_{\tilde{x}}\right) = \min_{(i,j)\in I}\lambda_i\lambda_j\mu_i\mu_j$. For any $(i,j)\in I$, it is straightforward to verify that

$$\min\left(\frac{1}{\lambda_i\mu_i},\frac{1}{\lambda_j\mu_j}\right) \le \left[\left(\frac{1}{\lambda_i}+\frac{1}{\lambda_j}\right)\frac{1}{\mu_i+\mu_j}\right] \le \max\left(\frac{1}{\lambda_i\mu_i},\frac{1}{\lambda_j\mu_j}\right), \quad \text{and}$$

$$\max\left((\lambda_i\mu_i)^2,(\lambda_j\mu_j)^2\right) \ge \lambda_i\lambda_j\mu_i\mu_j \ge \min\left((\lambda_i\mu_i)^2,(\lambda_j\mu_j)^2\right).$$

Hence, the max and min are achieved over the indices $(i,i)\in I$. This proves the first part of the lemma.

For the second part, the equality is easy to see. To show the inequality, note that

$$\lambda_{\max}(Q_{\tilde{x}^{1/2}}\tilde{q}) \le \|Q_{\tilde{x}^{1/2}}\tilde{q}\|_F.$$

For $p := \frac{Q_{\tilde{x}^{1/2}}\tilde{q}}{\|Q_{\tilde{x}^{1/2}}\tilde{q}\|_F}$, $\lambda_{\max}(p)\le 1$, and hence $e-p\in\mathcal{K}$. Since

$$\langle\tilde{q},\tilde{x}\rangle = \langle\tilde{q},Q_{\tilde{x}^{1/2}}e\rangle = \langle Q_{\tilde{x}^{1/2}}\tilde{q},e\rangle = \langle Q_{\tilde{x}^{1/2}}\tilde{q},e-p\rangle + \langle Q_{\tilde{x}^{1/2}}\tilde{q},p\rangle,$$

we have

$$\langle\tilde{q},\tilde{x}\rangle = \langle Q_{\tilde{x}^{1/2}}\tilde{q},e-p\rangle + \langle Q_{\tilde{x}^{1/2}}\tilde{q},p\rangle \ge \langle Q_{\tilde{x}^{1/2}}\tilde{q},p\rangle = \|Q_{\tilde{x}^{1/2}}\tilde{q}\|_F. \qquad \square$$

Proof (Claim 3.1). By expanding $\langle\triangle s+s-v,\triangle x+x-u\rangle$ and using (3.13), we find that

$$\langle\triangle s,\triangle x\rangle + \langle s-v,x-u\rangle + \langle\triangle s,x-u\rangle + \langle s-v,\triangle x\rangle = 0. \tag{4.1}$$

We will now bound the last three terms in the expansion. First, note that $\langle s-v,\triangle x\rangle = \langle\tilde{s}-\tilde{v},\triangle\tilde{x}\rangle = \left\langle G^{1/2}(\tilde{s}-\tilde{v}),G^{-1/2}\triangle\tilde{x}\right\rangle$ and using Cauchy–Schwarz, we see that

$$\langle G^{1/2}(\tilde{s}-\tilde{v}),G^{-1/2}\triangle\tilde{x}\rangle \le \|G^{1/2}(\tilde{s}-\tilde{v})\|_F\|G^{-1/2}\triangle\tilde{x}\|_F \le \|G^{1/2}(\tilde{s}-\tilde{v})\|_F t. \tag{4.2}$$

Next, note that

$$\|G^{1/2}(\tilde{s}-\tilde{v})\|_F^2 = \langle G^{1/2}(\tilde{s}-\tilde{v}),G^{1/2}(\tilde{s}-\tilde{v})\rangle = \langle\tilde{s}-\tilde{v},G(\tilde{s}-\tilde{v})\rangle. \tag{4.3}$$

Since $\tilde{x}$ and $\tilde{s}$ operator commute, operators G and $Q_{\tilde{x}}$ commute. Hence we have

$$\langle\tilde{s}-\tilde{v},G(\tilde{s}-\tilde{v})\rangle = \langle Q_{\tilde{x}}^{1/2}(\tilde{s}-\tilde{v}),Q_{\tilde{x}}^{-1}GQ_{\tilde{x}}^{1/2}(\tilde{s}-\tilde{v})\rangle \le \lambda_{\max}(Q_{\tilde{x}}^{-1}G)\|Q_{\tilde{x}}^{1/2}(\tilde{s}-\tilde{v})\|_F^2. \tag{4.4}$$

By substituting $q=s-v$ in the second part of Lemma 4.1, we get $\|Q_{\tilde{x}}^{1/2}(\tilde{s}-\tilde{v})\|_F \le \langle s-v,x\rangle$. Using (4.3) and (4.4), we see that

$$\|G^{1/2}(\tilde{s}-\tilde{v})\|_F^2 \le \lambda_{\max}(Q_{\tilde{x}}^{-1}G)\|Q_{\tilde{x}}^{1/2}(\tilde{s}-\tilde{v})\|_F^2 \le \frac{1}{\lambda_{\min}(z)}\langle s-v,x\rangle^2.$$

As $(x,s)\in\mathcal{N}_{-\infty}(\gamma)$, $\lambda_{\min}(z)\ge(1-\gamma)\mu$ and from (4.2) we have

$$\langle s-v,\triangle x\rangle \le \|G^{1/2}(\tilde{s}-\tilde{v})\|_F\|G^{-1/2}\triangle\tilde{x}\|_F \le \sqrt{\frac{1}{(1-\gamma)\mu}}\,\langle s-v,x\rangle\, t.$$

Similarly it can be shown that

$$\langle \triangle s, x-u\rangle \leq \sqrt{\frac{1}{(1-\gamma)\mu}}\,\langle s, x-u\rangle\, t.$$

Also using the feasibility condition (3.7), (3.13), and $\bar{\phi} \leq 1$, we get

$$\langle s-v, x-u\rangle \leq \bar{\phi}^2 \langle s_0-v_0, x_0-u_0\rangle \leq \frac{\langle s,x\rangle}{\langle s_0,x_0\rangle}\langle s_0-v_0, x_0-u_0\rangle.$$

Substituting the above bounds into (4.1) and using (3.14), we get

$$\begin{aligned}0 &\leq \langle \triangle s, \triangle x\rangle + \frac{\langle s,x\rangle}{\langle s_0,x_0\rangle}\langle s_0-v_0, x_0-u_0\rangle + \sqrt{\frac{1}{(1-\gamma)\mu}}\,\langle s, x-u\rangle\, t \\ &\quad + \sqrt{\frac{1}{(1-\gamma)\mu}}\,\langle s-v, x\rangle\, t \\ &= \langle \triangle s, \triangle x\rangle + \langle s,x\rangle\frac{\langle s_0-v_0, x_0-u_0\rangle}{\langle s_0,x_0\rangle} + \xi t\sqrt{\langle s,x\rangle}. \qquad \square\end{aligned}$$

5. Conclusion. We have established polynomial convergence of infeasible-interior-point methods for three important methods: the xs, sx, and the NT method. To our knowledge this is the first time an infeasible-interior-point method has been analyzed for the NT method using the $\mathcal{N}_{-\infty}$ neighborhood for both semidefinite programming and conic programs over symmetric cones. We have, in the process, provided a constructive proof of the Lyapunov lemma in the Jordan algebraic setting. The algorithm presented here is closely related to the algorithms used in practice to solve large-scale linear programs [6]. The complexity obtained for the NT method (in this general setting) coincides with the bound obtained for linear programming by Zhang [13]. The work by Rangarajan and Todd [10] shows convergence of the NT method using another neighborhood defined globally over the cone.

Acknowledgments. The author would like to thank Mike Todd for carefully reading through this manuscript and providing valuable feedback, and for insightful discussions. This work would not have been possible without his encouragement and support. The author is grateful to the two anonymous referees for their constructive suggestions and criticisms that led to an improved presentation.

REFERENCES

[1] J. Faraut and A. Koranyi, *Analysis on Symmetric Cones*, Oxford University Press, New York, 1994.

[2] L. Faybusovich, *Linear systems in Jordan algebras and primal-dual interior-point algorithms*, J. Comput. Appl. Math., 86 (1997), pp. 149–175.

[3] O. Güler, *Barrier functions in interior point methods*, Math. Oper. Res., 21 (1996), pp. 860–885.

[4] P. Jordan, J. von Neumann, and E. Wigner, *On an algebraic generalization of the quantum mechanical formalism*, Ann. of Math. (2), 35 (1934), pp. 29–64.

[5] M. Koecher, *The Minnesota Notes on Jordan Algebras and Their Applications*, Lecture Notes in Math. 1710, A. Krieg and S. Walcher eds., Springer-Verlag, Berlin, 1999.

[6] I. J. Lustig, R. Marsten, and D. F. Shanno, *Computational experience with a primal-dual interior point method for linear programming*, Linear Algebra Appl., 152 (1991), pp. 191–222.

[7] R. D. C. Monteiro and Y. Zhang, *A unified analysis for a class of path-following primal-dual path-following interior-point algorithms for semidefinite programming*, Math. Program., 81 (1998), pp. 281–299.

[8] Yu. E. Nesterov and A. S. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM, Philadelphia, 1994.

[9] Yu. E. Nesterov and M. J. Todd, *Self-scaled barriers and interior point methods for convex programming*, Math. Oper. Res., 22 (1997), pp. 1–42.

[10] B. Rangarajan and M. J. Todd, *Convergence of Infeasible-interior-point Methods for Self-scaled Conic Programming*, Tech. report 1388, School of OR & IE, Cornell University, Ithaca, NY, 2003.

[11] S. H. Schmieta and F. Alizadeh, *Extension of primal-dual interior point algorithm to symmetric cones*, Math. Program., 96 (2003), pp. 409–438.

[12] J. F. Sturm, *Similarity and other spectral relations for symmetric cones*, Linear Algebra Appl., 312 (2000), pp. 135–154.

[13] Y. Zhang, *On the convergence of a class of infeasible-interior-point methods for the horizontal linear complementarity problem*, SIAM J. Optim., 4 (1994), pp. 208–227.

[14] Y. Zhang, *On extending some primal-dual interior-point algorithms from linear programming to semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 365–386.

SIAM J. Optim.
Vol. 16, No. 4, pp. 1230–1241

STRONG CONVERGENCE THEOREM BY A HYBRID METHOD FOR NONEXPANSIVE MAPPINGS AND LIPSCHITZ-CONTINUOUS MONOTONE MAPPINGS[*]

NATALIA NADEZHKINA[†] AND WATARU TAKAHASHI[†]

Abstract. In this paper we introduce an iterative process for finding a common element of the set of fixed points of a nonexpansive mapping and the set of solutions of the variational inequality problem for a monotone, Lipschitz-continuous mapping. The iterative process is based on two well-known methods: hybrid and extragradient. We obtain a strong convergence theorem for three sequences generated by this process. Based on this result, we also construct an iterative process for finding a common fixed point of two mappings, such that one of these mappings is nonexpansive and the other is taken from the more general class of Lipschitz pseudocontractive mappings.

Key words. extragradient method, fixed point, hybrid method, monotone mapping, nonexpansive mapping, strong convergence, variational inequality

AMS subject classifications. 47H09, 47J20

DOI. 10.1137/050624315

1. Introduction.

Let C be a closed, convex, and nonempty subset of a real Hilbert space H and let P_C be the metric projection from H onto C. A mapping A of C into H is called *monotone* if

$$\langle Au - Av, u - v\rangle \geq 0$$

for all $u, v \in C$. A mapping A of C into H is called *k-Lipschitz-continuous* if there exists a positive real number k such that

$$\|Au - Av\| \leq k\,\|u - v\|$$

for all $u, v \in C$. Let the mapping A from C to H be monotone and Lipschitz-continuous. The *variational inequality problem* is to find a $u \in C$ such that

$$\langle Au, v - u\rangle \geq 0$$

for all $v \in C$. The set of solutions of the variational inequality problem is denoted by $VI\,(C, A)$. The variational inequality problem was first discussed by Lions [16] and now is well known; there are a lot of different approaches towards solving this problem in finite-dimensional and infinite-dimensional spaces, and the research is intensively continued. This problem has many applications in computational mathematics, mathematical physics, operation research, mathematical economics, optimization theory, and other fields; see, e.g., [10], [20], [31]. At the same time, to construct a mathematical model which is as close as possible to a real complex problem, we often have to use more than one constraint. Solving such problems, we have to obtain some solution which is simultaneously the solution of two or more subproblems or the solution of

[*] Received by the editors February 14, 2005; accepted for publication (in revised form) January 7, 2006; published electronically April 7, 2006.

http://www.siam.org/journals/siopt/16-4/62431.html

[†] Department of Mathematical and Computing Sciences, Graduate School of Information Science and Engineering, Tokyo Institute of Technology, 2-12-1, Oh-Okayama, Meguro, Tokyo 152-8552, Japan (nadezhkina.natalia@is.titech.ac.jp, wataru@is.titech.ac.jp).

one subproblem on the solution set of another subproblem. These subproblems can be given, for example, by two or more different variational inequalities (see, e.g., the lexicographic variational inequality problem [22]) or two or more different fixed point problems (see, e.g., a problem of obtaining a common fixed point of two or more mappings [27], [28]). These subproblems can also be given by problems of different types. Recently Antipin considered a finite-dimensional variant of the variational inequality problem, where the solution should satisfy some related constraint in inequality form [1] or some system of constraints in inequality and equality form [2]. Yamada [30] considered an infinite-dimensional variant of the solution of the variational inequality problem on the set of fixed points of some mapping. Takahashi and Toyoda [29] also formulated an infinite-dimensional variant of the problem of finding a common point of the set of the variational inequality solutions and the set of fixed points of some mapping. The problem of Takahashi and Toyoda can be formulated in the following way. A mapping A of C into H is called *α-inverse-strongly-monotone* if there exists a positive real number α such that

$$\langle Au - Av, u - v\rangle \geq \alpha \left\| Au - Av \right\|^2$$

for all $u, v \in C$; see [6], [17]. It is obvious that an α-inverse-strongly-monotone mapping A is monotone and Lipschitz-continuous. A mapping S of C into itself is called *nonexpansive* if

$$\| Su - Sv \| \leq \| u - v \|$$

for all $u, v \in C$; see [28]. We denote by $F(S)$ the set of fixed points of S. The problem of Takahashi and Toyoda entails finding an element of $F(S) \cap VI(C, A)$ under the assumption that a set $C \subset H$ is closed and convex, a mapping S of C into itself is nonexpansive, and a mapping A of C into H is α-inverse-strongly-monotone. To solving this problem Iiduka and Takahashi [12] introduced the following iterative scheme by a hybrid method:

$$\begin{cases} x_0 = x \in C, \\ y_n = \alpha_n x_n + (1 - \alpha_n) SP_C (x_n - \lambda_n A x_n), \\ C_n = \{ z \in C : \| y_n - z \| \leq \| x_n - z \| \}, \\ Q_n = \{ z \in C : \langle x_n - z, x - x_n \rangle \geq 0 \}, \\ x_{n+1} = P_{C_n \cap Q_n} x \end{cases}$$

for every $n = 0, 1, 2, \ldots,$ where $0 \leq \alpha_n \leq c < 1$ and $0 < a \leq \lambda_n \leq b < 2\alpha$. They showed that if $F(S) \cap VI(C, A)$ is nonempty, then the sequence $\{x_n\}$, generated by this iterative process, converges strongly to $P_{F(S) \cap VI(C,A)} x$. Generally speaking, the algorithm suggested by Iiduka and Takahashi is based on two well-known types of methods, namely, on the projection-type methods for solving variational inequality problems and so-called hybrid or outer-approximation methods for solving fixed point problem. The idea of “hybrid” or “outer-approximation” types of methods was originally introduced by Haugazeau in 1968 and was successfully generalized and extended in recent papers of Bauschke and Combettes [3], [4], Burachik, Lopes, and Svaiter [5], Combettes [8], Nakajo and Takahashi [18], and Solodov and Svaiter [25].

It is easy to see that the class of α-inverse-strongly-monotone mappings in the above-mentioned problem of Takahashi and Toyoda does not contain some important classes of mappings even in a finite-dimensional case. For example, if the matrix in the corresponding linear complementarity problem is positively semidefinite, but not positively definite, then the mapping A will be monotone and Lipschitz-continuous, but not α-inverse-strongly-monotone. It is also easy to see that while α-inverse-strongly-monotone mappings are tightly connected with the important class of nonexpansive mappings, monotone mappings are tightly connected with a more general and also quite important class of Lipschitz pseudocontractive mappings. (A mapping T from C to C is called *pseudocontractive* if $\|Tx - Ty\|^2 \leq \|x - y\|^2 + \|(I - T)\,x - (I - T)\,y\|^2$ for all $x, y \in C$.) Namely, if a mapping T from C to C is nonexpansive, then the mapping $A = I - T$ is 1/2-inverse-strongly-monotone; moreover, $F\,(T) = VI\,(C, A)$ (see, e.g., [29]). At the same time, if a mapping T from C to C is pseudocontractive and k-Lipschitz-continuous, then the mapping $A = I - T$ is monotone and $(k + 1)$-Lipschitz-continuous; moreover, $F\,(T) = VI\,(C, A)$ (see, e.g., proof of Theorem 4.5). So, it seems to be quite natural to try to get some result similar to the result of Iiduka and Takahashi for a more general class of monotone and Lipschitz-continuous mappings. But in this case we cannot apply the previous idea of combining projection-type and hybrid-type methods, because the mapping $SP_C\,(x_n - \lambda_n Ax_n)$ in this case is not nonexpansive and the usual schemes of proof are not applicable. In this paper the main idea is to investigate iterative schemes based on combination of hybrid-type methods and so-called extragradient-type methods. In 1976, for finding a solution of the nonconstrained variational inequality problem in the finite-dimensional Euclidean space $\mathbb{R}^n$ under the assumption that a set $C \subset \mathbb{R}^n$ is closed and convex and a mapping A of C into $\mathbb{R}^n$ is monotone and k-Lipschitz-continuous, Korpelevich [15] introduced the following so-called extragradient method:

$$(1.1) \qquad \begin{cases} x_0 = x \in C, \\ \overline{x}_n = P_C\,(x_n - \lambda Ax_n)\,, \\ x_{n+1} = P_C\,(x_n - \lambda A\overline{x}_n) \end{cases}$$

for every $n = 0, 1, 2, \ldots$, where $\lambda \in (0, 1/k)$. He showed that if $VI\,(C, A)$ is nonempty, then the sequences $\{x_n\}$ and $\{\overline{x}_n\}$, generated by (1.1), converge to the same point $z \in VI\,(C, A)$. The idea of the extragradient iterative process introduced by Korpelevich was successfully generalized and extended not only in Euclidean but also in Hilbert and Banach spaces; see, e.g., the recent papers of He, Yang, and Yuan [11], Gárciga Otero and Iuzem [9], Noor [19], Solodov and Svaiter [26], and Solodov [24].

In the present paper, by combining hybrid and extragradient methods, we introduce an iterative process for finding a common element of the set of fixed points of a nonexpansive mapping and the set of solutions of the variational inequality problem for a monotone, Lipschitz-continuous mapping in a real Hilbert space. Then, we obtain a strong convergence theorem for three sequences generated by this process. Some well-known strong convergence theorems in a Hilbert space follow from this result. Based on our main result, we construct an iterative process for finding a common fixed point of two mappings, one of which is nonexpansive and the other taken from the more general class of Lipschitz pseudocontractive mappings.

2. Preliminaries. Let H be a real Hilbert space with inner product $\langle\cdot,\cdot\rangle$ and norm $\|\cdot\|$ and let C be a closed, convex, and nonempty subset of H. We write $x_n \rightharpoonup x$ to indicate that the sequence $\{x_n\}$ converges weakly to x and $x_n \to x$ to indicate that $\{x_n\}$ converges strongly to x. For every point $x \in H$ there exists a unique nearest point in C, denoted by $P_C x$, such that $\|x - P_C x\| \leq \|x - y\|$ for all $y \in C$. P_C is called the *metric projection* of H onto C. We know that P_C is a nonexpansive mapping from H onto C. It is also known that $P_C x \in C$ and

$$\langle x - P_C x, P_C x - y \rangle \geq 0 \tag{2.1}$$

for all $x \in H$, $y \in C$; see [28] for more details. It is easy to see that (2.1) is equivalent to

$$\|x - y\|^2 \geq \|x - P_C x\|^2 + \|y - P_C x\|^2 \tag{2.2}$$

for all $x \in H$, $y \in C$.

Let A be a monotone mapping of C into H. In the context of the variational inequality problem the characterization of projection (2.1) implies

$$u \in VI(C, A) \Leftrightarrow u = P_C(u - \lambda A u) \quad \forall \lambda > 0.$$

It is also known that H satisfies Opial's condition [21], i.e., for any sequence $\{x_n\}$ with $x_n \rightharpoonup x$ the inequality

$$\liminf_{n \to \infty} \|x_n - x\| < \liminf_{n \to \infty} \|x_n - y\|$$

holds for every $y \in H$ with $y \neq x$.

A set-valued mapping $T : H \to 2^H$ is called *monotone* if for all $x, y \in H$, $f \in Tx$ and $g \in Ty$ imply $\langle x - y, f - g \rangle \geq 0$. A monotone mapping $T : H \to 2^H$ is *maximal* if its graph $G(T)$ is not properly contained in the graph of any other monotone mapping. It is known that a monotone mapping T is maximal if and only if for $(x, f) \in H \times H$, $\langle x - y, f - g \rangle \geq 0$ for every $(y, g) \in G(T)$ implies $f \in Tx$. Let A be a monotone, k-Lipschitz-continuous mapping of C into H and let $N_C v$ be the normal cone to C at $v \in C$, i.e., $N_C v = \{w \in H : \langle v - u, w \rangle \geq 0 \text{ for all } u \in C\}$. Define

$$Tv = \begin{cases} Av + N_C v & \text{if} \quad v \in C, \\ \emptyset & \text{if} \quad v \notin C. \end{cases}$$

It is known that in this case T is maximal monotone, and $0 \in Tv$ if and only if $v \in VI(C, A)$; see [23].

3. Strong convergence theorem. In this section we prove a strong convergence theorem by a combined hybrid-extragradient method for nonexpansive mappings and monotone, k-Lipschitz-continuous mappings.

THEOREM 3.1. *Let C be a closed convex subset of a real Hilbert space H. Let A be a monotone and k-Lipschitz-continuous mapping of C into H and let S be a nonexpansive mapping of C into itself such that $F(S) \cap VI(C, A) \neq \emptyset$. Let $\{x_n\}$, $\{y_n\}$, and $\{z_n\}$ be sequences generated by*

$$\begin{cases} x_0 = x \in C, \\ y_n = P_C(x_n - \lambda_n A x_n), \\ z_n = P_C(x_n - \lambda_n A y_n), \\ C_n = \{z \in C : \|z_n - z\| \leq \|x_n - z\|\}, \\ Q_n = \{z \in C : \langle x_n - z, x - x_n \rangle \geq 0\}, \\ x_{n+1} = P_{C_n \cap Q_n} x \end{cases}$$

for every $n = 0, 1, 2, \ldots$, where $\{\lambda_n\} \subset [a, b]$ for some $a, b \in (0, 1/k)$ and $\alpha_n \subset [0, c]$ for some $c \in [0, 1)$. Then the sequences $\{x_n\}$, $\{y_n\}$, and $\{z_n\}$ converge strongly to $P_{F(S) \cap VI(C,A)} x$.

Proof. It is obvious that C_n is closed and Q_n is closed and convex for every $n = 0, 1, 2, \ldots$. As $C_n = \{z \in C : \|z_n - x_n\|^2 + 2\langle z_n - x_n, x_n - z\rangle \leq 0\}$, we also have that C_n is convex for every $n = 0, 1, 2, \ldots$. As $Q_n = \{z \in C : \langle x_n - z, x - x_n\rangle \geq 0\}$, we have $\langle x_n - z, x - x_n\rangle \geq 0$ for all $z \in Q_n$ and, by (2.1), $x_n = P_{Q_n}x$. Put $t_n = P_C(x_n - \lambda_n Ay_n)$ for every $n = 0, 1, 2, \ldots$. Let $u \in F(S) \cap VI(C, A)$. From (2.2), monotonicity of A, and $u \in VI(C, A)$, we have

$$\begin{aligned}
\|t_n - u\|^2 &\leq \|x_n - \lambda_n Ay_n - u\|^2 - \|x_n - \lambda_n Ay_n - t_n\|^2 \\
&= \|x_n - u\|^2 - \|x_n - t_n\|^2 + 2\lambda_n \langle Ay_n, u - t_n\rangle \\
&= \|x_n - u\|^2 - \|x_n - t_n\|^2 \\
&\qquad +2\lambda_n (\langle Ay_n - Au, u - y_n\rangle + \langle Au, u - y_n\rangle + \langle Ay_n, y_n - t_n\rangle) \\
&\leq \|x_n - u\|^2 - \|x_n - t_n\|^2 + 2\lambda_n \langle Ay_n, y_n - t_n\rangle \\
&= \|x_n - u\|^2 - \|x_n - y_n\|^2 - 2\langle x_n - y_n, y_n - t_n\rangle - \|y_n - t_n\|^2 \\
&\qquad\qquad +2\lambda_n \langle Ay_n, y_n - t_n\rangle \\
&= \|x_n - u\|^2 - \|x_n - y_n\|^2 - \|y_n - t_n\|^2 + 2\langle x_n - \lambda_n Ay_n - y_n, t_n - y_n\rangle .
\end{aligned}$$

Further, since $y_n = P_C(x_n - \lambda_n Ax_n)$ and A is k-Lipschitz-continuous, we have

$$\begin{aligned}
&\langle x_n - \lambda_n Ay_n - y_n, t_n - y_n\rangle \\
&\qquad = \langle x_n - \lambda_n Ax_n - y_n, t_n - y_n\rangle + \langle \lambda_n Ax_n - \lambda_n Ay_n, t_n - y_n\rangle \\
&\qquad \leq \langle \lambda_n Ax_n - \lambda_n Ay_n, t_n - y_n\rangle \leq \lambda_n k \|x_n - y_n\| \|t_n - y_n\| .
\end{aligned}$$

So, we have

$$\begin{aligned}
\|t_n - u\|^2 &\leq \|x_n - u\|^2 - \|x_n - y_n\|^2 - \|y_n - t_n\|^2 + 2\lambda_n k \|x_n - y_n\| \|t_n - y_n\| \\
&\leq \|x_n - u\|^2 - \|x_n - y_n\|^2 - \|y_n - t_n\|^2 + \lambda_n^2 k^2 \|x_n - y_n\|^2 + \|y_n - t_n\|^2 \\
&\leq \|x_n - u\|^2 + \left(\lambda_n^2 k^2 - 1\right) \|x_n - y_n\|^2 \qquad (3.1) \\
&\leq \|x_n - u\|^2 .
\end{aligned}$$

Therefore, from (3.1), $z_n = \alpha_n x_n + (1 - \alpha_n) St_n$, and $u = Su$, we have

$$\begin{aligned}
\|z_n - u\|^2 &= \|\alpha_n x_n + (1 - \alpha_n) St_n - u\|^2 \\
&= \|\alpha_n (x_n - u) + (1 - \alpha_n)(St_n - u)\|^2 \\
&\leq \alpha_n \|x_n - u\|^2 + (1 - \alpha_n) \|St_n - u\|^2 \\
&\leq \alpha_n \|x_n - u\|^2 + (1 - \alpha_n) \|t_n - u\|^2 \qquad (3.2) \\
&\leq \|x_n - u\|^2 + (1 - \alpha_n) \left(\lambda_n^2 k^2 - 1\right) \|x_n - y_n\|^2 \\
&\leq \|x_n - u\|^2
\end{aligned}$$

for every $n = 0, 1, 2, \ldots$ and hence $u \in C_n$. So, $F(S) \cap VI(C, A) \subset C_n$ for every $n = 0, 1, 2, \ldots$. Next, let us show by mathematical induction that $\{x_n\}$ is well-defined and $F(S) \cap VI(C, A) \subset C_n \cap Q_n$ for every $n = 0, 1, 2, \ldots$. For $n = 0$ we have $Q_0 = C$. Hence we obtain $F(S) \cap VI(C, A) \subset C_0 \cap Q_0$. Suppose that x_k is given and $F(S) \cap VI(C, A) \subset C_k \cap Q_k$ for some $k \in N$. Since $F(S) \cap VI(C, A)$ is nonempty, $C_k \cap Q_k$ is a nonempty closed convex subset of C. So, there exists a unique element $x_{k+1} \in C_k \cap Q_k$ such that $x_{k+1} = P_{C_k \cap Q_k} x$. It is also obvious that there holds $\langle x_{k+1} - z, x - x_{k+1}\rangle \geq 0$ for every $z \in C_k \cap Q_k$. Since $F(S) \cap VI(C, A) \subset C_k \cap Q_k$, we have $\langle x_{k+1} - z, x - x_{k+1} \rangle \geq 0$ for $z \in F(S) \cap VI(C, A)$ and hence $F(S) \cap VI(C, A) \subset Q_{k+1}$. Therefore, we obtain $F(S) \cap VI(C, A) \subset C_{k+1} \cap Q_{k+1}$.

Let $l_0 = P_{F(S) \cap VI(C,A)} x$. From $x_{n+1} = P_{C_n \cap Q_n} x$ and $l_0 \in F(S) \cap VI(C, A) \subset C_n \cap Q_n$, we have

$$\|x_{n+1} - x\| \leq \|l_0 - x\| \tag{3.3}$$

for every $n = 0, 1, 2, \ldots$. Therefore, $\{x_n\}$ is bounded. From (3.1) and (3.2) we also obtain that $\{z_n\}$ and $\{t_n\}$ are bounded. Since $x_{n+1} \in C_n \cap Q_n \subset Q_n$ and $x_n = P_{Q_n} x$, we have

$$\|x_n - x\| \leq \|x_{n+1} - x\|$$

for every $n = 0, 1, 2, \ldots$. Therefore, there exists $\lim_{n\to\infty} \|x_n - x\|$. Since $x_n = P_{Q_n} x$ and $x_{n+1} \in Q_n$, using (2.2), we have

$$\|x_{n+1} - x_n\|^2 \leq \|x_{n+1} - x\|^2 - \|x_n - x\|^2$$

for every $n = 0, 1, 2, \ldots$. This implies that

$$\lim_{n\to\infty} \|x_{n+1} - x_n\| = 0.$$

Since $x_{n+1} \in C_n$, we have $\|z_n - x_{n+1}\| \leq \|x_n - x_{n+1}\|$ and hence

$$\|x_n - z_n\| \leq \|x_n - x_{n+1}\| + \|x_{n+1} - z_n\| \leq 2\|x_{n+1} - x_n\|$$

for every $n = 0, 1, 2, \ldots$. From $\|x_{n+1} - x_n\| \to 0$, we have $\|x_n - z_n\| \to 0$.

For $u \in F(S) \cap VI(C, A)$, from (3.2) we obtain

$$\|z_n - u\|^2 \leq \|x_n - u\|^2 + (1 - \alpha_n)\left(\lambda_n^2 k^2 - 1\right)\|x_n - y_n\|^2 .$$

Therefore, we have

$$\begin{aligned} \|x_n - y_n\|^2 &\leq \frac{1}{(1 - \alpha_n)(1 - \lambda_n^2 k^2)}\left(\|x_n - u\|^2 - \|z_n - u\|^2\right) \\ &= \frac{1}{(1 - \alpha_n)(1 - \lambda_n^2 k^2)}(\|x_n - u\| - \|z_n - u\|)(\|x_n - u\| + \|z_n - u\|) \\ &\leq \frac{1}{(1 - \alpha_n)(1 - \lambda_n^2 k^2)}(\|x_n - u\| + \|z_n - u\|)\|x_n - z_n\| . \end{aligned} \tag{3.4}$$

Since $\|x_n - z_n\| \to 0$ and the sequences $\{x_n\}$ and $\{z_n\}$ are bounded, we obtain $\|x_n - y_n\| \to 0$. By the same process as in (3.1), we also have

$$\begin{aligned} \|t_n - u\|^2 &\leq \|x_n - u\|^2 - \|x_n - y_n\|^2 - \|y_n - t_n\|^2 + 2\lambda_n k\|x_n - y_n\|\|t_n - y_n\| \\ &\leq \|x_n - u\|^2 - \|x_n - y_n\|^2 - \|y_n - t_n\|^2 + \|x_n - y_n\|^2 + \lambda_n^2 k^2\|y_n - t_n\|^2 \\ &\leq \|x_n - u\|^2 + \left(\lambda_n^2 k^2 - 1\right)\|y_n - t_n\|^2 . \end{aligned}$$

Then, in contrast with (3.2),

$$\begin{aligned}\|z_n-u\|^2 &= \|\alpha_n x_n+(1-\alpha_n)\,St_n-u\|^2\\ &= \|\alpha_n(x_n-u)+(1-\alpha_n)(St_n-u)\|^2\\ &\le \alpha_n\|x_n-u\|^2+(1-\alpha_n)\|St_n-u\|^2\\ &\le \alpha_n\|x_n-u\|^2+(1-\alpha_n)\|t_n-u\|^2\\ &\le \alpha_n\|x_n-u\|^2+(1-\alpha_n)\left(\|x_n-u\|^2+\left(\lambda_n^2k^2-1\right)\|y_n-t_n\|^2\right)\\ &\le \|x_n-u\|^2+(1-\alpha_n)\left(\lambda_n^2k^2-1\right)\|y_n-t_n\|^2\\ &\le \|x_n-u\|^2\end{aligned}$$

and, rearranging as in (3.4),

$$\begin{aligned}\|t_n-y_n\|^2 &\le \frac{1}{(1-\alpha_n)(1-\lambda_n^2k^2)}\left(\|x_n-u\|^2-\|z_n-u\|^2\right)\\ &= \frac{1}{(1-\alpha_n)(1-\lambda_n^2k^2)}(\|x_n-u\|-\|z_n-u\|)(\|x_n-u\|+\|z_n-u\|)\\ &\le \frac{1}{(1-\alpha_n)(1-\lambda_n^2k^2)}(\|x_n-u\|+\|z_n-u\|)\|x_n-z_n\|.\end{aligned}$$

Since $\|x_n-z_n\|\to 0$ and the sequences $\{x_n\}$ and $\{z_n\}$ are bounded, we obtain $\|t_n-y_n\|\to 0$. As A is k-Lipschitz-continuous, we have $\|Ay_n-At_n\|\to 0$. From $\|x_n-t_n\|\le\|x_n-y_n\|+\|y_n-t_n\|$ we also have $\|x_n-t_n\|\to 0$. Since $z_n=\alpha_n x_n+(1-\alpha_n)\,St_n$, we have $(1-\alpha_n)(St_n-t_n)=\alpha_n(t_n-x_n)+(z_n-t_n)$. Then

$$\begin{aligned}(1-c)\|St_n-t_n\| &\le (1-\alpha_n)\|St_n-t_n\|\\ &\le \alpha_n\|t_n-x_n\|+\|z_n-t_n\|\\ &\le (1+\alpha_n)\|t_n-x_n\|+\|z_n-x_n\|\end{aligned}$$

and hence $\|t_n-St_n\|\to 0$. As $\{x_n\}$ is bounded, there is a subsequence $\{x_{n_i}\}$ of $\{x_n\}$ such that $\{x_{n_i}\}$ converges weakly to some u. We can obtain that $u\in F(S)\cap VI(C,A)$. First, we show $u\in VI(C,A)$. Since $x_n-t_n\to 0$ and $x_n-y_n\to 0$, we have $\{t_{n_i}\}\rightharpoonup u$ and $\{y_{n_i}\}\rightharpoonup u$. Let

$$Tv=\begin{cases}Av+N_Cv & \text{if}\quad v\in C,\\ \emptyset & \text{if}\quad v\notin C,\end{cases}$$

where N_Cv is the normal cone to C at $v\in C$. We have already mentioned that in this case the mapping T is maximal monotone, and $0\in Tv$ if and only if $v\in VI(C,A)$; see [23]. Let $G(T)$ be the graph of T and let $(v,w)\in G(T)$. Then, we have $w\in Tv=Av+N_Cv$ and hence $w-Av\in N_Cv$. So, we have $\langle v-t,w-Av\rangle\ge 0$ for all $t\in C$. On the other hand, from $t_n=P_C(x_n-\lambda_nAy_n)$ and $v\in C$ we have

$$\langle x_n-\lambda_nAy_n-t_n,t_n-v\rangle\ge 0$$

and hence

$$\left\langle v-t_n,\frac{t_n-x_n}{\lambda_n}+Ay_n\right\rangle\ge 0.$$

From $\langle v-t, w-Av\rangle \geq 0$ for all $t \in C$ and $t_{n_i} \in C$, we have

$$\begin{aligned}\langle v-t_{n_i}, w\rangle &\geq \langle v-t_{n_i}, Av\rangle \\ &\geq \langle v-t_{n_i}, Av\rangle - \left\langle v-t_{n_i}, \frac{t_{n_i}-x_{n_i}}{\lambda_{n_i}} + Ay_{n_i}\right\rangle \\ &= \langle v-t_{n_i}, Av-At_{n_i}\rangle + \langle v-t_{n_i}, At_{n_i}-Ay_{n_i}\rangle - \left\langle v-t_{n_i}, \frac{t_{n_i}-x_{n_i}}{\lambda_{n_i}}\right\rangle \\ &\geq \langle v-t_{n_i}, At_{n_i}-Ay_{n_i}\rangle - \left\langle v-t_{n_i}, \frac{t_{n_i}-x_{n_i}}{\lambda_{n_i}}\right\rangle .\end{aligned}$$

Hence, we obtain $\langle v-u, w\rangle \geq 0$ as $i \to \infty$. Since T is maximal monotone, we have $u \in T^{-1}0$ and hence $u \in VI\,(C, A)$.

Let us show $u \in F\,(S)$. Assume $u \notin F\,(S)$. From Opial's condition, we have

$$\begin{aligned}\liminf_{i\to\infty} \|t_{n_i}-u\| &< \liminf_{i\to\infty} \|t_{n_i}-Su\| = \liminf_{i\to\infty} \|t_{n_i}-St_{n_i}+St_{n_i}-Su\| \\ &\leq \liminf_{i\to\infty} \|St_{n_i}-Su\| \leq \liminf_{i\to\infty} \|t_{n_i}-u\| .\end{aligned}$$

This is a contradiction. So, we obtain $u \in F\,(S)$. This implies $u \in F\,(S) \cap VI\,(C, A)$.

From $l_0 = P_{F(S)\cap VI(C,A)}x$, $u \in F\,(S) \cap VI\,(C, A)$, and (3.3), we have

$$\|l_0 - x\| \leq \|u-x\| \leq \liminf_{i\to\infty} \|x_{n_i}-x\| \leq \limsup_{i\to\infty} \|x_{n_i}-x\| \leq \|l_0-x\| .$$

So, we obtain

$$\lim_{i\to\infty} \|x_{n_i}-x\| = \|u-x\| .$$

From $x_{n_i}-x \rightharpoonup u-x$ we have $x_{n_i}-x \to u-x$ and hence $x_{n_i} \to u$. Since $x_n = P_{Q_n}x$ and $l_0 \in F\,(S) \cap VI\,(C, A) \subset C_n \cap Q_n \subset Q_n$, we have

$$-\|l_0-x_{n_i}\|^2 = \langle l_0-x_{n_i}, x_{n_i}-x\rangle + \langle l_0-x_{n_i}, x-l_0\rangle \geq \langle l_0-x_{n_i}, x-l_0\rangle .$$

As $i \to \infty$, we obtain $-\|l_0-u\|^2 \geq \langle l_0-u, x-l_0\rangle \geq 0$ by $l_0 = P_{F(S)\cap VI(C,A)}x$ and $u \in F\,(S) \cap VI\,(C, A)$. Hence we have $u = l_0$. This implies that $x_n \to l_0$. It is easy to see $y_n \to l_0$ and $z_n \to l_0$. □

4. Applications.

Using Theorem 3.1, we prove some strong convergence theorems in a real Hilbert space.

THEOREM 4.1. *Let C be a closed convex subset of a real Hilbert space H. Let A be a monotone and k-Lipschitz-continuous mapping of C into H such that $VI\,(C, A)$ is nonempty. Let $\{x_n\}$, $\{y_n\}$, and $\{z_n\}$ be sequences generated by*

$$\begin{cases} x_0 = x \in C, \\ y_n = P_C\,(x_n - \lambda_n Ax_n)\,, \\ z_n = P_C\,(x_n - \lambda_n Ay_n)\,, \\ C_n = \{z \in C : \|z_n - z\| \leq \|x_n - z\|\}\,, \\ Q_n = \{z \in C : \langle x_n - z, x - x_n\rangle \geq 0\}\,, \\ x_{n+1} = P_{C_n\cap Q_n}x \end{cases}$$

for every $n = 0, 1, 2, \ldots,$ where $\{\lambda_n\} \subset [a, b]$ for some $a, b \in (0, 1/k)$. Then the sequences $\{x_n\}$, $\{y_n\}$, and $\{z_n\}$ converge strongly to $P_{VI(C,A)}x$.

Proof. Putting $S = I$, $\alpha_n = 0$ for all $n = 0, 1, 2, \ldots$, by Theorem 3.1 we obtain the desired result. □

Remark. *See Iiduka, Takahashi, and Toyoda* [13] *for the case when the mapping A is α-inverse-strongly-monotone.*

Theorem 4.2. *Let C be a closed convex subset of a real Hilbert space H and let S be a nonexpansive mapping of C into itself such that $F(S)$ is nonempty. Let $\{x_n\}$ and $\{y_n\}$ be sequences generated by*

$$\begin{cases} x_0 = x \in C, \\ y_n = \alpha_n x_n + (1 - \alpha_n) S P_C x_n, \\ C_n = \{z \in C : \|y_n - z\| \leq \|x_n - z\|\}, \\ Q_n = \{z \in C : \langle x_n - z, x - x_n \rangle \geq 0\}, \\ x_{n+1} = P_{C_n \cap Q_n} x \end{cases}$$

for every $n = 0, 1, 2, \ldots$, where $\alpha_n \subset [0, c]$ for some $c \in [0, 1)$. Then the sequences $\{x_n\}$ and $\{y_n\}$ converge strongly to $P_{F(S)} x$.

Proof. Putting $A = 0$, by Theorem 3.1 we obtain the desired result. □

Remark. *Originally Theorem 4.2 is the result of Nakajo and Takahashi* [18].

Theorem 4.3. *Let H be a real Hilbert space. Let A be a monotone and k-Lipschitz-continuous mapping of H into itself and let S be a nonexpansive mapping of H into itself such that $F(S) \cap A^{-1}0 \neq \emptyset$. Let $\{x_n\}$ and $\{y_n\}$ be sequences generated by*

$$\begin{cases} x_0 = x \in H, \\ y_n = \alpha_n x_n + (1 - \alpha_n) S (x_n - \lambda_n A (x_n - \lambda_n A x_n)), \\ C_n = \{z \in H : \|y_n - z\| \leq \|x_n - z\|\}, \\ Q_n = \{z \in H : \langle x_n - z, x - x_n \rangle \geq 0\}, \\ x_{n+1} = P_{C_n \cap Q_n} x \end{cases}$$

for every $n = 0, 1, 2, \ldots$, where $\{\lambda_n\} \subset [a, b]$ for some $a, b \in (0, 1/k)$ and $\alpha_n \subset [0, c]$ for some $c \in [0, 1)$. Then the sequences $\{x_n\}$ and $\{y_n\}$ converge strongly to $P_{F(S) \cap A^{-1}0} x$.

Proof. We have $A^{-1}0 = VI(H, A)$ and $P_H = I$. By Theorem 3.1 we obtain the desired result. □

Let $B : H \to 2^H$ be a maximal monotone mapping. Then, for any $x \in H$ and $r > 0$, consider $J_r x = \{z \in H : z + rBz \ni x\}$. Such $J_r x$ is called the *resolvent* of B and is denoted by $J_r = (I + rB)^{-1}$.

Theorem 4.4. *Let H be a real Hilbert space. Let A be a monotone and k-Lipschitz-continuous mapping of H into itself and let $B : H \to 2^H$ be a maximal monotone mapping such that $A^{-1}0 \cap B^{-1}0 \neq \emptyset$. Let J_r be the resolvent of B for each $r > 0$. Let $\{x_n\}$ and $\{y_n\}$ be sequences generated by*

$$\begin{cases} x_0 = x \in H, \\ y_n = \alpha_n x_n + (1 - \alpha_n) J_r (x_n - \lambda_n A (x_n - \lambda_n A x_n)), \\ C_n = \{z \in H : \|y_n - z\| \leq \|x_n - z\|\}, \\ Q_n = \{z \in H : \langle x_n - z, x - x_n \rangle \geq 0\}, \\ x_{n+1} = P_{C_n \cap Q_n} x \end{cases}$$

for every $n = 0, 1, 2, \ldots$, where $\{\lambda_n\} \subset [a, b]$ for some $a, b \in (0, 1/k)$ and $\alpha_n \subset [0, c]$ for some $c \in [0, 1)$. Then the sequences $\{x_n\}$ and $\{y_n\}$ converge strongly to $P_{A^{-1}0 \cap B^{-1}0} x$.

Proof. We know that J_r^B is nonexpansive; see [28]. We also have $A^{-1}0 = VI\left(H, A\right)$ and $F\left(J_r^B\right) = B^{-1}0$. Putting $P_H = I$, by Theorem 3.1 we obtain the desired result. □

We also know one more definition of a pseudocontractive mapping, which is equivalent to the definition given in the introduction. A mapping T from C to C is called *pseudocontractive* if

$$\langle Tx - Ty, x - y\rangle \le \|x - y\|^2 \tag{4.1}$$

for all $x, y \in C$; see [6]. Obviously, the class of pseudocontractive mappings is more general than the class of nonexpansive mappings. Let us introduce two examples of Lipschitz pseudocontractive mappings. A linear operator $A : H \to H$ is called *positive* if $\langle Ax, x\rangle \ge 0$ for all $x \in H$. Let A be a bounded linear positive operator from H to H. Then the linear operator $T = I - A$ is Lipschitz-continuous and pseudocontractive. Let $B : H \to 2^H$ be a maximal monotone operator and let J_λ be the resolvent of B for $\lambda > 0$. We can also define the following operator, which is called the *Yosida approximation*: $B_\lambda = \frac{1}{\lambda}\left(I - J_\lambda\right)$. The operator $T = I - B_\lambda$ is also Lipschitz-continuous and pseudocontractive (see, e.g., [28]).

In the following theorem we introduce an iterative process that converges strongly to a common fixed point of two mappings, one of which is nonexpansive and the other Lipschitz-continuous and pseudocontractive.

THEOREM 4.5. *Let C be a nonempty closed convex subset of a real Hilbert space H. Let T be a pseudocontractive, m-Lipschitz-continuous mapping of C into itself and let S be a nonexpansive mapping of C into itself such that $F\left(T\right) \cap F\left(S\right) \ne \emptyset$. Let $\{z_n\}$ be a sequence generated by*

$$\begin{cases} x_0 = x \in C, \\ y_n = x_n - \lambda_n\left(x_n - Tx_n\right), \\ z_n = \alpha_n x_n + \left(1 - \alpha_n\right) SP_C\left(x_n - \lambda_n\left(y_n - Ty_n\right)\right), \\ C_n = \left\{z \in C : \|z_n - z\| \le \|x_n - z\|\right\}, \\ Q_n = \left\{z \in C : \langle x_n - z, x - x_n\rangle \ge 0\right\}, \\ x_{n+1} = P_{C_n \cap Q_n} x \end{cases}$$

for every $n = 0, 1, 2, \dots,$ where $\{\lambda_n\} \subset [a, b]$ for some a, b with $0 < a < b < \frac{1}{m+1}$ and $\alpha_n \subset [0, c]$ for some $c \in [0, 1)$. Then the sequences $\{x_n\}$, $\{y_n\}$, and $\{z_n\}$ converge strongly to $P_{F(T)\cap F(S)}x$.

Proof. Let $A = I - T$. Let us show the mapping A is monotone and $(m + 1)$-Lipschitz-continuous. From the definition of the mapping A and (4.1), we have

$$\begin{aligned} \langle Ax - Ay, x - y\rangle &= \langle x - y - Tx + Ty, x - y\rangle \\ &= \|x - y\|^2 - \langle Tx - Ty, x - y\rangle \ge \|x - y\|^2 - \|x - y\|^2 = 0. \end{aligned}$$

So, A is monotone. We also have

$$\begin{aligned} \|Ax - Ay\|^2 &= \|\left(I - T\right)x - \left(I - T\right)y\|^2 \\ &= \|x - y\|^2 + \|Tx - Ty\|^2 - 2\left\langle x - y, Tx - Ty\right\rangle \\ &\le \|x - y\|^2 + m^2\|x - y\|^2 + 2\|x - y\|\,\|Tx - Ty\| \\ &\le \|x - y\|^2 + m^2\|x - y\|^2 + 2m\|x - y\|^2 = \left(m + 1\right)^2\|x - y\|^2. \end{aligned}$$

So, we have $\|Ax - Ay\| \leq (m+1)\|x - y\|$ and A is $(m+1)$-Lipschitz-continuous.

Now let us show $F(T) = VI(C, A)$. In fact, we have, for $\lambda > 0$,

$$\begin{aligned} u \in VI(C, A) &\Leftrightarrow \langle y - u, Au\rangle \geq 0 \quad \forall y \in C \\ &\Leftrightarrow \langle u - y, u - \lambda Au - u\rangle \geq 0 \quad \forall y \in C \\ &\Leftrightarrow u = P_C(u - \lambda Au) \\ &\Leftrightarrow u = P_C(u - \lambda u + \lambda Tu) \\ &\Leftrightarrow \langle u - \lambda u + \lambda Tu - u, u - y\rangle \geq 0 \quad \forall y \in C \\ &\Leftrightarrow \langle u - Tu, u - y\rangle \leq 0 \quad \forall y \in C \\ &\Leftrightarrow u = Tu. \end{aligned}$$

By Theorem 3.1 we obtain the desired result. □

Acknowledgment. The authors thank the anonymous referees and Professor Heinz Bauschke for careful reading and constructive suggestions which led to considerable improvement of the paper.

REFERENCES

[1] A. S. Antipin, *Methods for solving variational inequalities with related constraints*, Comput. Math. Math. Phys., 40 (2000), pp. 1239–1254.

[2] A. S. Antipin and F. P. Vasiliev, *Regularized prediction method for solving variational inequalities with an inexactly given set*, Comput. Math. Math. Phys., 44 (2004), pp. 750–758.

[3] H. H. Bauschke and P. L. Combettes, *A weak-to-strong convergence principle for Fejér-monotone methods in Hilbert spaces*, Math. Oper. Res., 26 (2001), pp. 248–264.

[4] H. H. Bauschke and P. L. Combettes, *Construction of best Bregman approximations in reflexive Banach spaces*, Proc. Amer. Math. Soc., 131 (2003), pp. 3757–3766.

[5] R. S. Burachik, J. O. Lopes, and B. F. Svaiter, *An outer approximation method for the variational inequality problem*, SIAM J. Control Optim., 43 (2005), pp. 2071–2088.

[6] F. E. Browder and W. V. Petryshyn, *Construction of fixed points of nonlinear mappings in Hilbert space*, J. Math. Anal. Appl., 20 (1967), pp. 197–228.

[7] F. E. Browder, *Nonlinear monotone and accretive operators in Banach spaces*, Proc. Nat. Acad. Sci. USA, 61 (1968), pp. 388–393.

[8] P. L. Combettes, *Strong convergence of block-iterative outer approximation methods for convex optimization*, SIAM J. Control Optim., 38 (2000), pp. 538–565.

[9] R. Gárciga Otero and A. Iuzem, *Proximal methods with penalization effects in Banach spaces*, Numer. Funct. Anal. Optim., 25 (2004), pp. 69–91.

[10] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, New York, 1984.

[11] B.-S. He, Z.-H. Yang, and X.-M. Yuan, *An approximate proximal-extragradient type method for monotone variational inequalities*, J. Math. Anal. Appl., 300 (2004), pp. 362–374.

[12] H. Iiduka and W. Takahashi, *Strong convergence theorem by a hybrid method for nonlinear mappings of nonexpansive and monotone type and applications*, Adv. Nonlinear Var. Inequal., 9 (2006), pp. 1–10.

[13] H. Iiduka, W. Takahashi, and M. Toyoda, *Approximation of solutions of variational inequalities for monotone mappings*, Panamer. Math. J., 14 (2004), pp. 49–61.

[14] D. Kinderlehrer and G. Stampacchia, *An Introduction to Variational Inequalities and Their Applications*, Academic Press, New York, 1980.

[15] G. M. Korpelevich, *The extragradient method for finding saddle points and other problems*, Matecon, 12 (1976), pp. 747–756.

[16] J. L. Lions, *Quelques Méthodes de Résolution des Problèmes aux Limites Non Linéaires*, Dunod, Paris, 1969.

[17] F. Liu and M. Z. Nashed, *Regularization of nonlinear ill-posed variational inequalities and convergence rates*, Set-Valued Anal., 6 (1998), pp. 313–344.

[18] K. Nakajo and W. Takahashi, *Strong convergence theorems for nonexpansive mappings and nonexpansive semigroups*, J. Math. Anal. Appl., 279 (2003), pp. 372–379.

[19] M. A. Noor, *New extragradient-type methods for general variational inequalities*, J. Math. Anal. Appl., 277 (2003), pp. 379–394.

[20] J. T. Oden, *Quantitative Methods on Nonlinear Mechanics*, Prentice-Hall, Englewood Cliffs, NJ, 1986.

[21] Z. Opial, *Weak convergence of the sequence of successive approximations for nonexpansive mappings*, Bull. Amer. Math. Soc., 73 (1967), pp. 591–597.

[22] L. D. Popov, *On a one-stage method for solving lexicographic variational inequalities*, Izv. Vyssh. Uchebn. Zaved. Mat., 12 (1998), pp. 71–81.

[23] R. T. Rockafellar, *On the maximality of sums of nonlinear monotone operators*, Trans. Amer. Math. Soc., 149 (1970), pp. 75–88.

[24] M. V. Solodov, *Convergence rate analysis of iteractive algorithms for solving variational inequality problem*, Math. Program., 96 (2003), pp. 513–528.

[25] M. V. Solodov and B. F. Svaiter, *Forcing strong convergence of proximal point iterations in a Hilbert space*, Math. Program., 87 (2000), pp. 189–202.

[26] M. V. Solodov and B. F. Svaiter, *An inexact hybrid generalized proximal point algorithm and some new results on the theory of Bregman functions*, Math. Oper. Res., 25 (2000), pp. 214–230.

[27] W. Takahashi, *Convex Analysis and Approximation of Fixed Points*, Yokohama Publishers, Yokohama, Japan, 2000.

[28] W. Takahashi, *Nonlinear Functional Analysis*, Yokohama Publishers, Yokohama, Japan, 2000.

[29] W. Takahashi and M. Toyoda, *Weak convergence theorems for nonexpansive mappings and monotone mappings*, J. Optim. Theory Appl., 118 (2003), pp. 417–428.

[30] I. Yamada, *The hybrid steepest-descent method for the variational inequality problem over the intersection of fixed-point sets of nonexpansive mappings*, in Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications, D. Butnariu, Y. Censor, and S. Reich, eds., Kluwer Academic, Dordrecht, The Netherlands, 2001, pp. 473–504.

[31] E. Zeidler, *Nonlinear Functional Analysis and Its Applications*, Springer-Verlag, New York, 1985.