SIAM J. OPTIM.
Vol. 7, No. 1, pp. 1–25, February 1997

ON THE CONVERGENCE OF PATTERN SEARCH ALGORITHMS*

VIRGINIA TORCZON†

Abstract. We introduce an abstract definition of pattern search methods for solving nonlinear unconstrained optimization problems. Our definition unifies an important collection of optimization methods that neither compute nor explicitly approximate derivatives. We exploit our characterization of pattern search methods to establish a global convergence theory that does not enforce a notion of sufficient decrease. Our analysis is possible because the iterates of a pattern search method lie on a scaled, translated integer lattice. This allows us to relax the classical requirements on the acceptance of the step, at the expense of stronger conditions on the form of the step, and still guarantee global convergence.

Key words. unconstrained optimization, convergence analysis, direct search methods, globalization strategies, alternating variable search, axial relaxation, local variation, coordinate search, evolutionary operation, pattern search, multidirectional search, downhill simplex search

AMS subject classifications. 49D30, 65K05

PII. S1052623493250780

1. Introduction.

We consider the familiar problem of minimizing a continuously differentiable function $f : \mathbf{R}^n \to \mathbf{R}$. Direct search methods for this problem are methods that neither compute nor explicitly approximate derivatives of f. Our interest is in a particular subset of direct search methods that we will call *pattern search methods*. Our purpose is to generalize these methods and to present a global convergence theory for them. To our knowledge, this is the first convergence result for some of these methods and the first general convergence theory for all of them.

Examples of pattern search methods include such classical direct search algorithms as *coordinate search* with fixed step sizes, *evolutionary operation* using factorial designs (first proposed by G. E. P. Box [2, 3, 13]), and the original *pattern search algorithm* of Hooke and Jeeves [7]. A more recent example is the *multidirectional search algorithm* of Dennis and Torczon [6, 15]. For some time, it has been apparent to us that the unifying theme that distinguishes these algorithms from other direct search methods is that each of them performs a search using a "pattern" of points that is independent of the objective function f. This informal insight is the basis for our general definition of pattern search methods—it turns out that each of the above pattern search methods is an instance of our general model.

Formally, our definition of pattern search methods requires the existence of a lattice T such that if $\{x_1, \ldots, x_N\}$ are the first N iterates generated by a pattern search method, then there exists a scale factor ϕ_N such that the steps $\{x_1 - x_0, x_2 - x_1, \ldots, x_N - x_{N-1}\}$ all lie in the scaled lattice $\phi_N T$. The lattice depends on the pattern that defines the individual method and on the initial choice of the step length control parameter, but it is independent of the objective function f. The scaling

[*] Received by the editors June 23, 1993; accepted for publication (in revised form) September 20, 1995. This research was sponsored by Air Force Office of Scientific Research grants 89-0363, F49620-93-1-0212, and F49620-95-1-0210; United States Air Force grant F49629-92-J-0203; and Department of Energy grant DE-FG005-86ER25017. This research was also supported in part by the Geophysical Parallel Computation Project under State of Texas contract 1059.

http://www.siam.org/journals/siopt/7-1/25078.html

[†] Department of Computer Science, The College of William & Mary, Williamsburg, VA 23187-8795 (va@cs.wm.edu). This work was completed while the author was in the Department of Computational and Applied Mathematics and the Center for Research on Parallel Computation, Rice University, Houston, TX 77251-1892.

depends solely on the sequence of updates that have been applied to the step length control parameter.

Despite isolated convergence results [4, 11, 16] for certain individual pattern search methods, a general theory of convergence for the class of such methods remained elusive for some time. The standard convergence theory for line search and trust region methods depends crucially on some notion of sufficient decrease, but pattern search methods do not enforce any such notion. Therefore, attempts such as [18] to apply the standard theory to pattern search methods arbitrarily introduce some notion of sufficient decrease, thereby modifying the original algorithms. Thus, the challenge was to develop a general convergence theory for pattern search methods without redefining what they are.

Our convergence analysis is guided by that found in Torczon [16] for the multidirectional search algorithm; however, the present level of abstraction makes the important elements of that analysis easier to appreciate. The present paper also includes a correction to the specification of the scaling factors found in [16].

There are three key points to our analysis. First, we show that pattern search methods are descent methods. Second, we prove that pattern search methods are gradient-related methods in the sense of [10]. Finally, we demonstrate that pattern search methods cannot terminate prematurely due to inadequate step length control mechanisms. The crucial element of this analysis is the fact that pattern search methods are able to relax the conditions on accepting a step by enforcing stronger conditions on the step itself. The lattice T, together with the way in which the step length control parameter is updated, prevent a pathological choice of steps: steps of arbitrary lengths along arbitrary search directions are not permitted.

We are able to guarantee that, if the function f is continuously differentiable, then $\liminf_{k \to +\infty} \|\nabla f(x_k)\| = 0$ without an explicit representation of the gradient or the directional derivative. In particular, we prove global convergence for pattern search methods despite the fact that they do not explicitly enforce a notion of sufficient decrease on their iterates, such as fraction of Cauchy decrease, fraction of optimal decrease, or the Armijo–Goldstein–Wolfe conditions. However, our convergence analysis does share certain characteristics with the classical convergence analysis of both line search and trust region methods. This connection is both subtle and unexpected.

Our convergence analysis for pattern search methods makes it clear why these methods are as robust as their proponents have long claimed, while clarifying some of the limitations that have long been ascribed to them. In addition, having identified the common structure of these methods, it is now possible to develop new pattern search methods with guaranteed global convergence.

In section 2 we establish the notation and general specification of pattern search methods. In section 3 we prove that if the function to be minimized is continuously differentiable, then pattern search methods guarantee that $\liminf_{k \to +\infty} \|\nabla f(x_k)\| = 0$. In addition, we identify the modifications that must be made to pattern search methods to obtain the stronger result $\lim_{k \to +\infty} \|\nabla f(x_k)\| = 0$. In section 4 we show that the classical pattern search methods mentioned above, as well as the newer multidirectional search algorithm of Dennis and Torczon, conform to the general specification for pattern search methods. In section 5, we give some concluding remarks; section 6 contains technical results needed for the proofs of section 3.

Notation. We denote by $\mathbf{R}$, $\mathbf{Q}$, $\mathbf{Z}$, and $\mathbf{N}$ the sets of real, rational, integer, and natural numbers, respectively.

All norms are Euclidean vector norms or the associated operator norm. We define

$L(y) = \{x : f(x) \leq f(y)\}$, $C(y) = \{x : f(x) = f(y)\}$, and $X_* = \{x : \nabla f(x) = 0\}$.

2. Pattern search methods.

We begin by introducing the following abstraction of pattern search methods. We defer to section 4 demonstrations that the pattern search methods mentioned above fall comfortably within this abstraction.

2.1. The pattern.

To define a pattern we need two components, a *basis matrix* and a *generating matrix*.

The basis matrix can be any nonsingular matrix $B \in \mathbf{R}^{n\times n}$.

The generating matrix is a matrix $C_k \in \mathbf{Z}^{n\times p}$, where $p > 2n$. We partition the generating matrix into components

$$C_k \quad = \quad [M_k \quad -M_k \quad L_k] \quad = \quad [\Gamma_k \quad L_k]. \tag{1}$$

We require that $M_k \in \mathbf{M} \subset \mathbf{Z}^{n\times n}$, where $\mathbf{M}$ is a finite set of nonsingular matrices, and that $L_k \in \mathbf{Z}^{n\times(p-2n)}$ and contains at least one column, the column of zeros.

A *pattern* P_k is then defined by the columns of the matrix $P_k = BC_k$. Because both B and C_k have rank n, the columns of P_k span $\mathbf{R}^n$. For convenience, we use the partition of the generating matrix C_k given in (1) to partition P_k as follows:

$$P_k \quad = \quad BC_k \quad = \quad [BM_k \quad -BM_k \quad BL_k] \quad = \quad [B\Gamma_k \quad BL_k]. \tag{2}$$

Given $\Delta_k \in \mathbf{R}$, $\Delta_k > 0$, we define a *trial step* s_k^i to be any vector of the form

$$s_k^i = \Delta_k B c_k^i\,, \tag{3}$$

where c_k^i denotes a column of $C_k = [c_k^1 \cdots c_k^p]$. Note that Bc_k^i determines the direction of the step, while Δ_k serves as a step length parameter.

At iteration k, we define a *trial point* as any point of the form $x_k^i = x_k + s_k^i$, where x_k is the current iterate.

2.2. The exploratory moves.

Pattern search methods proceed by conducting a series of *exploratory moves* about the current iterate before declaring a new iterate and updating the associated information. These moves can be viewed as sampling the function about the current iterate x_k in a well-defined deterministic fashion in search of a new iterate $x_{k+1} = x_k + s_k$ with a lower function value. The individual pattern search methods are distinguished, in part, by the manner in which these exploratory moves are conducted. To allow the broadest possible choice of exploratory moves and yet still maintain the properties required to prove convergence for the pattern search methods, we place two requirements on the exploratory moves associated with any particular pattern search method. These requirements are given in the following Hypotheses on exploratory moves. (Please note an abuse of notation that is nonetheless convenient: $y \in A$ means that the vector y is contained in the set of columns of the matrix A.)

HYPOTHESES ON EXPLORATORY MOVES.

1. $s_k \in \Delta_k P_k \equiv \Delta_k BC_k \equiv \Delta_k\,[B\Gamma_k \; BL_k]$.
2. If $\min\{f(x_k + y),\ y \in \Delta_k B\Gamma_k\} < f(x_k)$, then $f(x_k + s_k) < f(x_k)$.

The choice of exploratory moves must ensure two things:

1. The direction of any step s_k accepted at iteration k is defined by the pattern P_k, and its length is determined by Δ_k.
2. If simple decrease on the function value at the current iterate can be found among any of the $2n$ trial steps defined by $\Delta_k B\Gamma_k$, then the exploratory moves must produce a step s_k that also gives simple decrease on the function

value at the current iterate. In particular, $f(x_k + s_k)$ need not be less than or equal to $\min\{f(x_k + y),\ y \in \Delta_k B\Gamma_k\}$.

Thus, a legitimate exploratory moves algorithm would be one that somehow guesses which of the steps defined by $\Delta_k P_k$ will produce simple decrease and then evaluates the function at only one such step. (And that step may be contained in $\Delta_k BL_k$ rather than in $\Delta_k B\Gamma_k$.) At the other extreme, a legitimate exploratory moves algorithm would be one that evaluates all p steps defined by $\Delta_k P_k$ and returns the step that produced the least function value.

These are the properties of the exploratory moves that enable us to prove

$$\liminf_{k\to+\infty} \|\nabla f(x_k)\| = 0,$$

even though we only require simple decrease on f. Thus we avoid the necessity of enforcing either fraction of Cauchy decrease, fraction of optimal decrease, or the Armijo–Goldstein–Wolfe conditions on the iterates. To obtain

$$\lim_{k\to+\infty} \|\nabla f(x_k)\| = 0,$$

we need to place stronger hypotheses on the exploratory moves as well as place a boundedness condition on the columns of the generating matrices. These extensions will be discussed further in section 3.3.2.

2.3. The generalized pattern search method.

2.3. The generalized pattern search method. Algorithm 1 states the generalized pattern search method for unconstrained minimization.

Algorithm 1. The Generalized Pattern Search Method.

```
Let $x_0 \in \mathbf{R}^n$ and $\Delta_0 > 0$ be given.
For $k = 0, 1, \ldots,$
    (a) Compute $f(x_k)$.
    (b) Determine a step $s_k$ using an exploratory moves algorithm.
    (c) Compute $\rho_k = f(x_k) - f(x_k + s_k)$.
    (d) If $\rho_k > 0$ then $x_{k+1} = x_k + s_k$. Otherwise $x_{k+1} = x_k$.
    (e) Update $C_k$ and $\Delta_k$.
```

To define a particular pattern search method, it is necessary to specify the basis matrix B, the generating matrix C_k, the exploratory moves to be used to produce a step s_k, and the algorithms for updating C_k and Δ_k.

2.4. The updates. Algorithm 2 specifies the requirements for updating Δ_k. The aim of the updating algorithm for Δ_k is to force $\rho_k > 0$. An iteration with $\rho_k > 0$ is *successful*; otherwise, the iteration is *unsuccessful.* Again we note that to accept a step we only require *simple*, as opposed to *sufficient*, decrease.

Algorithm 2. Updating Δ_k.

```
Given $\tau \in \mathbf{Q}$, let $\theta = \tau^{w_0}$ and $\lambda_k \in \Lambda = \{\tau^{w_1}, \ldots, \tau^{w_L}\}$, where $\tau > 1$ and
$\{w_0, w_1, \ldots, w_L\} \subset \mathbf{Z}$, $L \equiv |\Lambda| < +\infty$, $w_0 < 0$, and $w_i \geq 0$, $i = 1, \ldots, L$.
    (a) If $\rho_k \leq 0$ then $\Delta_{k+1} = \theta\Delta_k$.
    (b) If $\rho_k > 0$ then $\Delta_{k+1} = \lambda_k\Delta_k$.
```

The conditions on θ and Λ ensure that $0 < \theta < 1$ and $\lambda_i \geq 1$ for all $\lambda_i \in \Lambda$. Thus, if an iteration is successful it may be possible to increase the step length parameter Δ_k, but Δ_k is not allowed to decrease. Not surprisingly, this is crucial to the success of the analysis. Also crucial to the analysis is the relationship (overlooked in [16]) between θ and the elements of Λ.

The algorithm for updating C_k depends on the pattern search method. For theoretical purposes, it is sufficient to choose the columns of C_k so that they satisfy (1) and the conditions we have placed on the matrices $M_k \in \mathbf{M} \subset \mathbf{Z}^{n\times n}$ and $L_k \in \mathbf{Z}^{n\times(p-2n)}$.

3. The convergence theory.

Having set up the machinery to define pattern search methods, we are now ready to analyze these methods. This analysis produces theorems of several types. The first, developed in section 3.1, demonstrates an algebraic fact about the nature of pattern search methods that requires no assumption on the function f. This theorem is critical to the proof of the convergence results for it shows that we only need require simple decrease in f to ensure global convergence. The second theorem, developed in section 3.2, describes the limiting behavior of the step length control parameter Δ_k if we place only a very mild condition on the function f and exploit the interaction of the simple decrease condition for the generalized pattern search method with the algorithm for updating Δ_k. Finally, the third and fourth theorems, developed in section 3.3, give the global convergence results. The first theorem guarantees $\liminf_{k\to+\infty} \|\nabla f(x_k)\| = 0$ for any generalized pattern search method that satisfies the specifications given in section 2. This is significant since the theorem applies to all the pattern search methods we discuss in section 4 without the need to impose any modifications on the methods as originally stated. The second theorem is equivalent to convergence results for line search and trust-region globalization strategies. We can guarantee $\lim_{k\to+\infty} \|\nabla f(x_k)\| = 0$, but to do so requires placing stronger conditions on the specifications for generalized pattern search methods. We could certainly impose these stronger conditions on the pattern search methods presented in section 4—none of them are unreasonable to suggest or to enforce—but we would do so at the expense of attractive algorithmic features found in the original methods.

3.1. The algebraic structure of the iterates.

The results found in this section are purely algebraic facts about the nature of pattern search methods; they are also independent of the function to be optimized. It is the algebraic structure of the iterates that allows us to prove global convergence for pattern search methods without imposing a notion of sufficient decrease on the iterates.

We begin by showing in what sense Δ_k is a step length parameter.

LEMMA 3.1. *There exists a constant $\zeta_* > 0$, independent of k, such that for any trial step $s_k^i \neq 0$ produced by a generalized pattern search method (Algorithm 1) we have*

$$\|s_k^i\| \geq \zeta_* \Delta_k.$$

Proof. From (3) we have $s_k^i = \Delta_k B c_k^i$. The conditions we have placed on the generating matrix C_k ensure that $c_k^i \in \mathbf{Z}^n$.

Let $\sigma_n(B)$ denote the smallest singular value of B. Then

$$\|s_k^i\| \quad = \quad \Delta_k \|B c_k^i\| \quad \geq \quad \Delta_k \sigma_n(B) \|c_k^i\| \quad \geq \quad \Delta_k \sigma_n(B).$$

The last inequality holds because at least one of the components of c_k^i is a nonzero integer, and hence $\|c_k^i\| \geq 1$. □

From Lemma 3.1 we can see that the role of Δ_k as a step length parameter is to regulate backtracking and thus prevent excessively short steps.

THEOREM 3.2. *Any iterate x_N produced by a generalized pattern search method*

(Algorithm 1) can be expressed in the following form:

$$x_N = x_0 + \left(\beta^{r_{LB}} \alpha^{-r_{UB}}\right) \Delta_0 B \sum_{k=0}^{N-1} z_k,$$

where

- x_0 *is the initial guess,*
- $\beta/\alpha \equiv \tau$, *with* $\alpha, \beta \in \mathbf{N}$ *and relatively prime, and* τ *is as defined in the algorithm for updating* Δ_k *(Algorithm 2),*
- r_{LB} *and* r_{UB} depend on N*,*
- Δ_0 *is the initial choice for the step length control parameter,*
- B *is the basis matrix, and*
- $z_k \in \mathbf{Z}^n$*,* $k = 0, \ldots, N-1$*.*

Proof. The generalized pattern search algorithm, as stated in Algorithm 1, guarantees that any iterate x_N is of the form

$$x_N = x_0 + \sum_{k=0}^{N-1} s_k. \tag{4}$$

(We adopt the convention that $s_k = 0$ if iteration k is unsuccessful.) We also know that the step s_k must come from the set of trial steps s_k^i, $i = 1, \ldots, p$. The trial steps are of the form $s_k^i = \Delta_k B c_k^i$.

Consider the step length parameter Δ_k. For any $k \geq 0$, the update for Δ_k given in Algorithm 2 guarantees that Δ_k is of the form

$$\Delta_k = \theta^{q_k^0} \lambda_1^{q_k^1} \lambda_2^{q_k^2} \cdots \lambda_L^{q_k^L} \Delta_0, \tag{5}$$

where $q_k^i \in \mathbf{Z}$ and $q_k^i \geq 0$. (Recall that $L = |\Lambda|$.) We have also placed the following restrictions on the form of θ and λ_i: for a given $\tau \in \mathbf{Q}$, $\tau > 1$, and $\{w_0, w_1, \ldots, w_L\} \subset \mathbf{Z}$, $\theta = \tau^{w_0}$, $w_0 < 0$ and $\lambda_i = \tau^{w_i}$, $w_i \geq 0$, $i = 1, \ldots, L$. We can thus rewrite (5) as:

$$\Delta_k \quad = \quad (\tau^{w_0})^{q_k^0} (\tau^{w_1})^{q_k^1} (\tau^{w_2})^{q_k^2} \cdots (\tau^{w_L})^{q_k^L} \Delta_0 \quad = \quad \tau^{r_k} \Delta_0, \tag{6}$$

where $r_k \in \mathbf{Z}$. Let

$$r_{LB} = \min_{0 \leq k < N} \{r_k\} \qquad r_{UB} = \max_{0 \leq k < N} \{r_k\}. \tag{7}$$

Then from (4) and (6) we have

$$x_N = x_0 + \sum_{k=0}^{N-1} \Delta_k B c_k = x_0 + \Delta_0 B \sum_{k=0}^{N-1} \tau^{r_k} c_k.$$

Since τ is rational, we can express τ as $\tau = \frac{\beta}{\alpha}$, where $\alpha, \beta \in \mathbf{N}$ are relatively prime. Then, using (7),

$$x_N = x_0 + \left(\beta^{r_{LB}} \alpha^{-r_{UB}}\right) \Delta_0 B \sum_{k=0}^{N-1} z_k, \tag{8}$$

where $z_k \in \mathbf{Z}^n$. □

Theorem 3.2 synthesizes the requirements we have placed on the pattern, the definition of the trial steps, and the algorithm for updating Δ_k. Note that this means that for a fixed N, all the iterates lie on a translated integer lattice generated by x_0 and the columns of $\beta^{r_{LB}} \alpha^{-r_{UB}} \Delta_0 B$.

3.2. The limiting behavior of the step length control parameter. The next theorem combines the strict algebraic structure of the iterates with the simple decrease condition of the generalized pattern search algorithm, along with the algorithm for updating Δ_k, to give us a useful fact about the limiting behavior of Δ_k.

THEOREM 3.3. *Assume that $L(x_0)$ is compact. Then* $\liminf_{k\to+\infty} \Delta_k = 0$.

Proof. The proof is by contradiction. Suppose $0 < \Delta_{LB} \leq \Delta_k$ for all k. From (6) we know that Δ_k can be written as $\Delta_k = \tau^{r_k}\Delta_0$, where $r_k \in \mathbf{Z}$.

The hypothesis that $\Delta_{LB} \leq \Delta_k$ for all k means that the sequence $\{\tau^{r_k}\}$ is bounded away from zero. Meanwhile, we also know that the sequence $\{\Delta_k\}$ is bounded above because all the iterates x_k must lie inside the set $L(x_0) = \{x : f(x) \leq f(x_0)\}$, and the latter set is compact; Lemma 3.1 then guarantees an upper bound Δ_{UB} for $\{\Delta_k\}$. This, in turn, means that the sequence $\{\tau^{r_k}\}$ is bounded above. Consequently, the sequence $\{\tau^{r_k}\}$ is a finite set. Equivalently, the sequence $\{r_k\}$ is bounded above and below.

Let

$$r_{LB} = \min_{0\leq k<+\infty}\{r_k\} \qquad r_{UB} = \max_{0\leq k<+\infty}\{r_k\}. \tag{9}$$

Then (8) now holds for the bounds given in (9), rather than (7), and we see that for all k, x_k lies in the translated integer lattice G generated by x_0 and the columns of $\beta^{r_{LB}}\alpha^{-r_{UB}}\Delta_0 B$.

The intersection of the compact set $L(x_0)$ with the translated integer lattice G is finite. Thus, there must exist at least one point x_* in the lattice for which $x_k = x_*$ for infinitely many k.

We appeal to the simple decrease condition in the generalized pattern search method (Algorithm 1 (d)), which guarantees that a lattice point cannot be revisited infinitely many times since we accept a new step s_k if and only if $f(x_k) > f(x_k + s_k)$. Thus there exists an N such that for all $k \geq N$, $x_k = x_*$, which implies that $\rho_k = 0$.

We now appeal to the algorithm for updating Δ_k (Algorithm 2 (a)) to see that $\Delta_k \to 0$, thus leading to a contradiction. □

3.3. Global convergence. Throughout the discussion in this section, we assume that f is continuously differentiable on a neighborhood of $L(x_0)$; however, this assumption can be weakened, using the same style of argument found in [16].

3.3.1. The general result. To prove Theorem 3.5 we need Proposition 3.4. We defer the proof of Proposition 3.4 to section 6 in part because we wish to discuss there several other issues that are tangential to the proof of Theorem 3.5. It is also the case that the proofs for the results in section 6 are similar to those given for the equivalent results found in [16], though now restated more succinctly in terms of the machinery developed in section 2.

PROPOSITION 3.4. *Assume that $L(x_0)$ is compact, that f is continuously differentiable on a neighborhood of $L(x_0)$, and that* $\liminf_{k\to+\infty}\|\nabla f(x_k)\| \neq 0$. *Then there exists a constant $\Delta_{LB} > 0$ such that for all k, $\Delta_k > \Delta_{LB}$.*

We emphasize that the existence of a positive lower bound Δ_{LB} for Δ_k is guaranteed only under the null hypothesis that $\liminf_{k\to+\infty}\|\nabla f(x_k)\| \neq 0$.

THEOREM 3.5. *Assume that $L(x_0)$ is compact and that f is continuously differentiable on a neighborhood of $L(x_0)$. Then for the sequence of iterates $\{x_k\}$ produced by the generalized pattern search method (Algorithm 1),*

$$\liminf_{k\to+\infty}\|\nabla f(x_k)\| = 0\,.$$

Proof. The proof is by contradiction. Suppose that $\liminf_{k\to+\infty} \|\nabla f(x_k)\| \neq 0$. Then Proposition 3.4 tells us that there exists $\Delta_{LB} > 0$ such that for all k, $\Delta_k \geq \Delta_{LB}$. But this contradicts Theorem 3.3. □

3.3.2. The stronger result. We can strengthen the result given in Theorem 3.5 at the expense of wider applicability. To begin with, we must add three further restrictions: one on the pattern matrix, one on the Hypotheses on exploratory moves, and one on the limiting behavior of the step length control parameter Δ_k.

First, we must ensure that the columns of the generating matrix C_k are bounded in norm, i.e., that there exists a constant $\mathcal{C} > 0$ such that for all k, $\mathcal{C} > \|c_k^i\|$ for all $i = 1, \ldots, p$. Given this bound, we can place an upper bound, in terms of Δ_k, on the norm of any trial step s_k^i.

Lemma 3.6. *Given a constant $\mathcal{C} > 0$ such that for all k, $\mathcal{C} > \|c_k^i\|$ for all $i = 1, \ldots, p$, there exists a constant $\psi_* > 0$, independent of k, such that for any trial step s_k^i produced by a generalized pattern search method (Algorithm 1) we have*

$$\Delta_k \geq \psi_* \|s_k^i\|.$$

Proof. From (3) we have $s_k^i = \Delta_k B c_k^i$. Then $\|s_k^i\| = \Delta_k \|Bc_k^i\| \leq \Delta_k \|B\| \|c_k^i\| \leq \Delta_k \mathcal{C} ||B||$. Set $\psi_* = \frac{1}{\mathcal{C}||B||}$. □

Note that the columns of $M_k \in \mathbf{M}$ are bounded by the assumption that $|\mathbf{M}| < +\infty$; we use this fact in the proof of Proposition 6.4. The stronger boundedness condition on the columns of $C_k = [M_k \; -M_k \; L_k]$ is needed to monitor the behavior of L_k.

Second, we must replace the original Hypotheses on exploratory moves with a stronger version, as given below. Together, Lemma 3.6 and the Strong hypotheses on exploratory moves allow us to tie decrease in f to the norm of the gradient when the step sizes get small enough. This is the import of Corollary 6.5, which is given in section 6.

Strong hypotheses on exploratory moves.

1. $s_k \in \Delta_k P_k \equiv \Delta_k B C_k \equiv \Delta_k \, [B\Gamma_k \; BL_k]$.
2. If $\min\{f(x_k + y),\ y \in \Delta_k B\Gamma_k\} < f(x_k)$, then
   $f(x_k + s_k) \leq \min\{f(x_k + y),\ y \in \Delta_k B\Gamma_k\}$.

Third, we require that $\lim_{k\to+\infty} \Delta_k = 0$. We can use the algorithm for updating Δ_k (Algorithm 2) to ensure that this condition holds. For instance, we can force Δ_k to be nonincreasing by requiring $w_i = 0$, $i = 1, \ldots, L$, which when taken together with Theorem 3.3 guarantees that $\lim_{k\to+\infty} \Delta_k = 0$. All the algorithms we consider in section 4, except the multidirectional search algorithm, enforce this condition by limiting $\Lambda = \{1\} \equiv \{\tau^0\}$. However, it is not necessary to force the steps to be nonincreasing; we need only require that in the limit the step length control parameter goes to zero, which, in conjunction with Lemmas 3.1 and 3.6, has the effect of ultimately forcing the steps to zero.

Theorem 3.7. *Assume that $L(x_0)$ is compact and that f is continuously differentiable on a neighborhood of $L(x_0)$. In addition, assume that the columns of the generating matrices are bounded in norm, that $\lim_{k\to+\infty} \Delta_k = 0$, and that the generalized pattern search method (Algorithm 1) enforces the Strong hypotheses on exploratory moves. Then for the sequence of iterates $\{x_k\}$ produced by the generalized pattern search method,*

$$\lim_{k\to+\infty} \|\nabla f(x_k)\| = 0\,.$$

Proof. The proof is by contradiction. Suppose $\limsup_{k\to+\infty} \|\nabla f(x_k)\| \neq 0$. Let $\varepsilon > 0$ be such that there exists a subsequence $\|\nabla f(x_{m_i})\| \geq \varepsilon$. Since

$$\liminf_{k\to+\infty} \|\nabla f(x_k)\| = 0,$$

given any $0 < \eta < \varepsilon$, there exists an associated subsequence l_i such that

$$\|\nabla f(x_k)\| \quad > \eta \qquad \text{for} \qquad m_i \leq k < l_i, \qquad \|\nabla f(x_{l_i})\| \quad < \quad \eta.$$

Then, since $\Delta_k \to 0$, we can appeal to Corollary 6.5 to obtain for $m_i \leq k < l_i$, i sufficiently large,

$$f(x_k) - f(x_{k+1}) \quad \geq \quad \sigma\|\nabla f(x_k)\|\|s_k\| \quad \geq \quad \sigma\eta\|s_k\|,$$

where $\sigma > 0$. Then the telescoping sum

$$(f(x_{m_i}) - f(x_{m_i+1})) + (f(x_{m_i+1}) - f(x_{m_i+2})) + \cdots + (f(x_{l_i-1}) - f(x_{l_i})) \geq \sum_{k=m_i}^{l_i} \sigma\eta\|s_k\|$$

gives us

$$f(x_{m_i}) - f(x_{l_i}) \quad \geq \quad \textstyle\sum_{k=m_i}^{l_i} \sigma\eta\|s_k\| \quad \geq \quad c'\|x_{m_i} - x_{l_i}\|.$$

Since f is bounded below, $f(x_{m_i}) - f(x_{l_i}) \to 0$ as $i \to +\infty$, so $\|x_{m_i} - x_{l_i}\| \to 0$ as $i \to +\infty$. Then, because ∇f is uniformly continuous,

$$\|\nabla f(x_{m_i}) - \nabla f(x_{l_i})\| < \eta$$

for i sufficiently large. However,

$$\|\nabla f(x_{m_i})\| \quad \leq \quad \|\nabla f(x_{m_i}) - \nabla f(x_{l_i})\| + \|\nabla f(x_{l_i})\| \quad \leq \quad 2\eta. \tag{10}$$

Since equation (10) must hold for any η, $0 < \eta < \varepsilon$, we have a contradiction (e.g., try $\eta = \frac{\varepsilon}{4}$). □

The proof of Theorem 3.7 is almost identical to that of an equivalent result for trust-region methods that was first given by Thomas [14] and which is included, in a more general form, in the survey by Moré [8].

One final note: the hypotheses of Theorem 3.7 suggest that in the absence of any explicit higher-order information about the function to be minimized, it makes sense to terminate a generalized pattern search algorithm when Δ_k is less than some reasonably small tolerance. In fact, this is a common stopping condition for algorithms of this sort and the one implemented for the multidirectional search algorithm [17].

4. The particular pattern search methods.

In section 2 we stated the conditions an algorithm must satisfy to be a pattern search method. We now illustrate these conditions by considering the following specific algorithms:

- coordinate search with fixed step lengths,
- evolutionary operation using factorial designs [2, 3, 13],
- the original pattern search method of Hooke and Jeeves [7], and
- the multidirectional search algorithm of Dennis and Torczon [6, 15].

We will show that these algorithms satisfy the conditions that define pattern search methods and thus are special cases of the generalized pattern search method presented as Algorithm 1. Then we can appeal to Theorem 3.5 to claim global convergence for these methods.

There are other algorithms for which the abstraction and accompanying analysis holds—including various modifications to the algorithms presented—but we shall confine our investigation to these, the best known of the pattern search methods, to illustrate the power of our abstract approach to pattern search methods.

4.1. Coordinate search with fixed step lengths. The method of coordinate search is perhaps the simplest and most obvious of all the pattern search methods. Davidon describes it concisely in the opening of his belated preface to Argonne National Laboratory Research and Development Report 5990 [5]:

> Enrico Fermi and Nicholas Metropolis used one of the first digital computers, the Los Alamos Maniac, to determine which values of certain theoretical parameters (phase shifts) best fit experimental data (scattering cross sections). They varied one theoretical parameter at a time by steps of the same magnitude, and when no such increase or decrease in any one parameter further improved the fit to the experimental data, they halved the step size and repeated the process until the steps were deemed sufficiently small. Their simple procedure was slow but sure....

This simple search method enjoys many names, among them *alternating directions*, *alternating variable search*, *axial relaxation*, and *local variation*. We shall refer to it as *coordinate search*.

Perhaps less obvious is that coordinate search is a pattern search method. To see this, we begin by considering all possible outcomes for a single iteration of coordinate search when $n = 2$, as shown in Fig. 1. We mark the current iterate x_k. The x_k^i's denote trial points considered during the course of the iteration. The next iterate x_{k+1} is marked. Solid circles indicate successful intermediate steps taken during the course of the exploratory moves while open circles indicate points at which the function was evaluated but that did not produce further decrease in the value of the objective function. Thus, in the first scenario shown a step from x_k to x_k^1 resulted in a decrease in the objective function, so the step from x_k^1 to x_{k+1} was tried and led to a further decrease in the objective function value. The iteration was then terminated with a new point x_{k+1} that satisfies the simple decrease condition $f(x_{k+1}) < f(x_k)$. In the worst case, the last scenario shown, $2n$ trial points were evaluated (x_k^1, $x_k^{1'}$, x_k^2, and $x_k^{2'}$) without producing decrease in the function value at the current iterate x_k. In this case, $x_{k+1} = x_k$ and the step size must be reduced for the next iteration.

We now show this algorithm is an instance of a generalized pattern search method.

4.1.1. The matrices. Coordinate search is usually defined so that the basis matrix is the identity matrix; i.e., $B = I$. However, knowledge of the problem may lead to a different choice for the basis matrix. It may make sense to search using a different coordinate system. For instance, if the variables are known to differ by several orders of magnitude, this can be taken into account in the choice of the basis matrix (though, as we will see in section 6.2, this may have a significant effect on the behavior of the method).

The generating matrix for coordinate search is fixed across all iterations of the method. The generating matrix $C_k = C$ contains in its columns all possible combi-

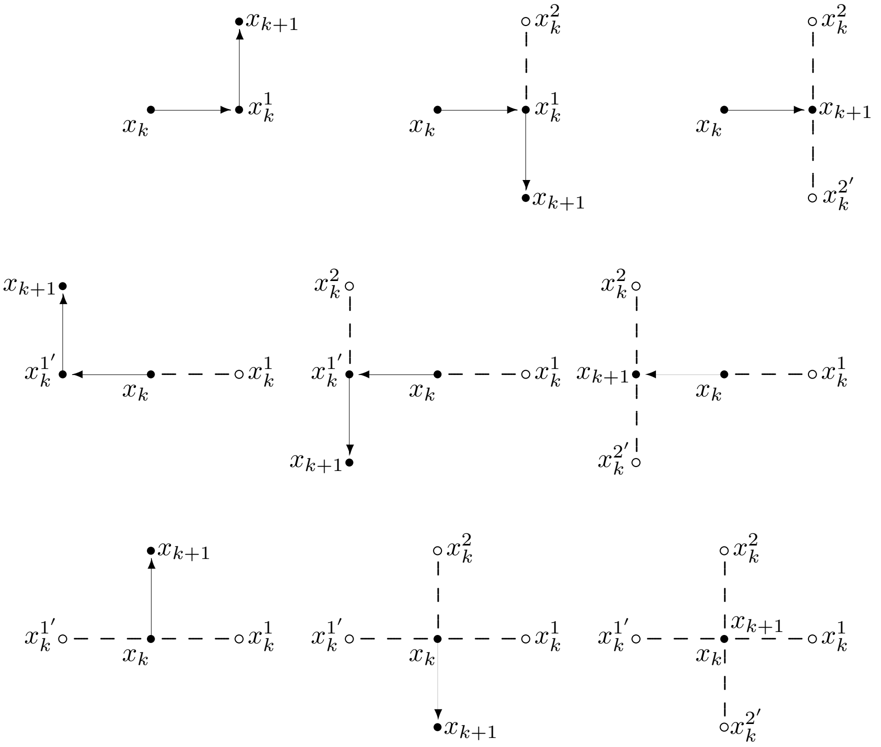


FIG. 1. *All possible subsets of the steps for coordinate search in* $\mathbf{R}^2$.

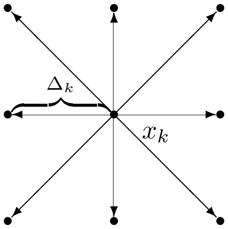


FIG. 2. *The pattern for coordinate search in* $\mathbf{R}^2$ *with a given step length control parameter* Δ_k.

nations of $\{-1, 0, 1\}$. Thus, C has $p = 3^n$ columns. In particular, the columns of C contain both I and $-I$, as well as a column of zeros. We define $M = I$; L consists of the remaining $3^n - 2n$ columns of C. Since C is fixed across all iterations of the method, there is no need for an update algorithm.

For $n = 2$ we have

$$C = \left[\begin{array}{ccccccccc} 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 & 1 & -1 & -1 & 1 & 0 \end{array} \right].$$

Thus, when $n = 2$, all possible trial points defined by the pattern $P = BC$, for a given step length Δ_k, can be seen in Fig. 2. Note that the pattern includes all the possible trial points enumerated in Fig. 1.

4.1.2. The exploratory moves. The exploratory moves for coordinate search are given in Algorithm 3, where the e_i's denote the unit coordinate vectors.

Algorithm 3. Exploratory Moves Algorithm for Coordinate Search.

```
Given $x_k$, $\Delta_k$, $f(x_k)$, and $B$, set $s_k = 0$, $\rho_k = 0$, and min $= f(x_k)$.
For $i = 1, \ldots, n$ do
    (a) $s_k^i = s_k + \Delta_k B e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.
    (b) If $f(x_k^i) <$ min then $\rho_k = f(x_k) - f(x_k^i)$, min $= f(x_k^i)$, and $s_k = s_k^i$.
        Otherwise,
          (i) $s_k^i = s_k - \Delta_k B e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.
         (ii) If $f(x_k^i) <$ min then $\rho_k = f(x_k) - f(x_k^i)$, min $= f(x_k^i)$, and $s_k = s_k^i$.
Return.
```

The exploratory moves are executed sequentially in the sense that the selection of the next trial step is based on the success or failure of the previous trial step. Thus, while there are 3^n possible trial steps, we may compute as few as n trial steps, but we compute no more than $2n$ at any given iteration, as we saw in Fig. 1.

From the perspective of the theory, there are two conditions that need to be met by the exploratory moves algorithm. First, as Figs. 1 and 2 illustrate, all possible trial steps are contained in $\Delta_k P$.

The second condition on the exploratory moves is the more interesting; coordinate search demonstrates the laxity of this second hypothesis. For instance, in the first scenario shown in Fig. 1, decrease in the objective function was realized for the first trial step

$$s_k^1 = \Delta_k I \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

so the second trial step

$$s_k^2 = \Delta_k I \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \Delta_k I \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \Delta_k I \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

was tried and accepted. It is certainly possible that greater decrease in the value of the objective function might have been realized for the trial step

$$s_k^{'} = \Delta_k I \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

which is defined by a column in the matrix M (the step s_k^2 is defined by a column in the matrix L), but $s_k^{'}$ is not tried when simple decrease is realized by the step s_k^1. However, in the worst case, as seen in Fig. 1, the algorithm for coordinate search ensures that all $2n$ steps defined by $\Delta_k B \Gamma = \Delta_k B[M \; -M] = \Delta_k B[I \; -I]$ are tried before returning the step $s_k = 0$. In other words, the exploratory moves given in Algorithm 3 examine all $2n$ steps defined by $\Delta_k B \Gamma$ unless a step satisfying $f(x_k + s_k) < f(x_k)$ is found.

4.1.3. Updating the step length. The update for Δ_k is exactly as given in Algorithm 2. As noted by Davidon, the usual practice is to continue with steps of the same magnitude until no further decrease in the objective function is realized, at which point the step size is halved. This corresponds to setting $\theta = 1/2$ and $\Lambda = \{1\}$. Thus, $\tau = 2$, $w_0 = -1$, and $w_1 = 0$.

This suffices to verify that coordinate search with fixed step length is a pattern search method. Theorem 3.5 thus holds. The exploratory moves algorithm for coordinate search would need to be modified to satisfy the Strong hypotheses on exploratory moves for the conditions of Theorem 3.7 to be met.

4.2. Evolutionary operation using factorial designs. In 1957 G. E. P. Box [2] introduced the notion of evolutionary operation as a method for increasing industrial productivity. The ideas were developed within the context of the on-line management of industrial processes, but Box recognized that the technique had more general applicability. Subsequent authors [3, 13] argued that the basic technique was readily applicable to general unconstrained optimization and it is within this context that we examine the ideas here.

In its simplest form, evolutionary operation is based on using two-level factorial designs: evaluate the function at the vertices of a hypercube centered about the current iterate. (G. E. P. Box refers to this as one of a variety of "pattern of variants" [2].) If simple decrease in the value of the objective function is observed at one of the vertices, it becomes the new iterate. Otherwise, the lengths of the edges in the hypercube are halved and the process is repeated.

4.2.1. The matrices. As with coordinate search, the usual choice for the basis matrix is $B = I$, though, as with coordinate search, other choices may be made to reflect information known about the problem to be solved.

The generating matrix for evolutionary operation is fixed across all iterations of the method. The generating matrix $C_k = C$ contains in its columns all possible combinations of $\{-1, 1\}$; to this we append a column of zeros. Thus C has $p = 2^n + 1$ columns.

We take M to be any linearly independent subset of n columns of C; $-M$ necessarily will be contained in C. Once again, L is fixed and consists of the remaining $(2^n + 1) - 2n$ columns of C.

There is no need for an algorithm to update C since the generating matrix is fixed.

4.2.2. The exploratory moves. The exploratory moves given in Algorithm 4 are simultaneous in the sense that every possible trial step $s_k^i \in \Delta_k P = \Delta_k BC$ is computed at each iteration. It is then the case that every trial step s_k^i is contained in $\Delta_k P$. The second observation of note is that since

$$s_k = \underset{s_k^i \in \Delta_k P}{\arg\min} \{f(x_k + s_k^i)\},$$

then, if $\min\{f(x_k + y), y \in \Delta_k B\Gamma\} < f(x_k)$, we have $f(x_k + s_k) < f(x_k)$, regardless of our choice of M (and thus, by extension, our choice of Γ). Furthermore, we are guaranteed that the Strong hypotheses on exploratory moves are satisfied.

ALGORITHM 4. EXPLORATORY MOVES ALGORITHM FOR EVOLUTIONARY OPERATION.

```
Given $x_k$, $\Delta_k$, $f(x_k)$, $B$, and $C = \left[c^1 \cdots c^p\right]$, set $s_k = 0$, $\rho_k = 0$, and min = $f(x_k)$.
For $i = 1, \ldots, 2^n$ do
    (a) $s_k^i = \Delta_k B c^i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.
    (b) If $f(x_k^i) <$ min then $\rho_k = f(x_k) - f(x_k^i)$, min = $f(x_k^i)$, and $s_k = s_k^i$.
Return.
```

4.2.3. Updating the step length. The algorithm for updating Δ_k is exactly as given in Algorithm 2, with θ usually set to $1/2$ and $\Lambda = \{1\}$.

Since we have shown that evolutionary operation satisfies all the necessary requirements, we can therefore conclude that it, too, is a pattern search method, so Theorem 3.5 holds. The algorithm, as stated above, also satisfies the conditions of Theorem 3.7.

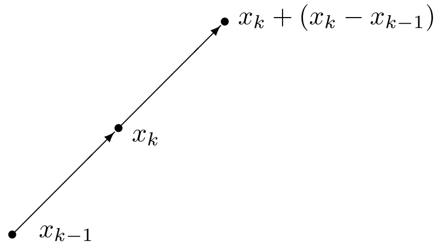


FIG. 3. *The pattern step in* $\mathbf{R}^2$, *given* $x_k \neq x_{k-1}$, $k > 0$.

4.3. Hooke and Jeeves' pattern search algorithm. In addition to introducing the general notion of a "direct search" method, Hooke and Jeeves introduced the pattern search method, a specific kind of search strategy [7]. The pattern search of Hooke and Jeeves is a variant of coordinate search that incorporates a *pattern step* in an attempt to accelerate the progress of the algorithm by exploiting information gained from the search during previous successful iterations.

The Hooke and Jeeves pattern search algorithm is opportunistic. If the previous iteration was successful (i.e., $\rho_{k-1} > 0$), then the current iteration begins by conducting coordinate search about a speculative iterate $x_k + (x_k - x_{k-1})$, rather than about the current iterate x_k. This is the pattern step. The idea is to investigate whether further progress is possible in the general direction $x_k - x_{k-1}$ (since, if $x_k \neq x_{k-1}$, then $x_k - x_{k-1}$ is clearly a promising direction).

To make this a little clearer, we consider the example shown in Fig. 3. Given x_{k-1} and x_k (we assume, for now, that $k > 0$ and that $x_k \neq x_{k-1}$), the pattern search algorithm takes the step $x_k - x_{k-1}$ from x_k. The function is evaluated at this trial step and the trial step is accepted, temporarily, even if $f(x_k + (x_k - x_{k-1})) \geq f(x_k)$. The Hooke and Jeeves pattern search algorithm then proceeds to conduct coordinate search about the temporary iterate $x_k + (x_k - x_{k-1})$. Thus, in $\mathbf{R}^2$, the exploratory moves are exactly as shown in Fig. 1, but with $x_k + (x_k - x_{k-1})$ substituted for x_k.

If coordinate search about the temporary iterate $x_k + (x_k - x_{k-1})$ is successful, then the point returned by coordinate search about the temporary iterate is accepted as the new iterate x_{k+1}. If not, i.e., $f((x_k + (x_k - x_{k-1})) + s_k) \geq f(x_k)$, then the pattern step is deemed unsuccessful, and the method reduces to coordinate search about x_k. For the two dimensional case, then, the exploratory moves would simply resort to the possibilities shown in Fig. 1.

If the previous iteration was not successful, so $x_k = x_{k-1}$ and $(x_k - x_{k-1}) = 0$, then the iteration is limited to coordinate search about x_k. In this instance, though, the updating algorithm for Δ_k will have reduced the size of the step (i.e., $\Delta_k = \theta\Delta_{k-1}$).

The algorithm does not execute the pattern step when $k = 0$.

To express the pattern search algorithm within the framework we have developed, we use all the machinery required for coordinate search. Once again, the basis matrix is usually defined to be $B = I$. We append to the generating matrix another set of 3^n columns to capture the effect of the pattern step and we change the exploratory moves algorithm, as detailed below.

4.3.1. The generating matrix. Recall that the generating matrix for coordinate search consists of all possible combinations of $\{-1, 0, 1\}$ and is never changed. For the Hooke and Jeeves pattern search method, we allow the generating matrix to change from iteration to iteration to capture the effect of the pattern step. We append another set of 3^n columns, consisting of all possible combinations of $\{-1, 0, 1\}$, to the initial generating matrix for coordinate search. Thus C_k has $p = 2 \cdot 3^n$ columns. The additional 3^n columns allow us to express the effect of the pattern step with respect to x_k, rather than with respect to the temporary iterate $x_k + (x_k - x_{k-1})$, which is how the Hooke and Jeeves pattern search method usually is described. The matrix M is unchanged; $M = I$. Now, however, $L_k \in \mathbf{Z}^{n \times (p-2n)}$ is allowed to vary, though only in the 3^n columns associated with the pattern step. For $n = 2$,

$$C_0 = \left[\begin{array}{rrrrrrrrr} 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 & 1 & -1 & -1 & 1 & 0 \end{array} \right.$$
$$\left. \begin{array}{rrrrrrrrr} 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 & 1 & -1 & -1 & 1 & 0 \end{array} \right]. \tag{11}$$

For notational convenience, we require that the last column of C_0, which we denote as c_0^p, be the column of zeros. In both the algorithm for updating C_k (Algorithm 5) and the algorithm for the exploratory moves (Algorithm 6), we use the column c_k^p to measure the accumulation of a sequence of successful pattern steps. This can be seen, in (12), for our example from Fig. 3. In this example, we have the generating matrix

$$C_k = \left[\begin{array}{rrrrrrrrr} 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 & 1 & -1 & -1 & 1 & 0 \end{array} \right.$$
$$\left. \begin{array}{rrrrrrrrr} 2 & 1 & 0 & 1 & 2 & 2 & 0 & 0 & 1 \\ 1 & 2 & 1 & 0 & 2 & 0 & 0 & 2 & 1 \end{array} \right]. \tag{12}$$

The pattern step $(x_k - x_{k-1})$ is represented by the vector $(1\ 1)^T$, seen in the last column of C_k. Note that the only difference between the columns of C_0 given in (11) and the columns of C_k given in (12) is that $(1\ 1)^T$ has been added to the last 3^2 columns of C_k.

The algorithm for updating the generating matrix updates the last 3^n columns of C_k; the first 3^n columns remain unchanged, as in coordinate search. The purpose of the updating algorithm is to incorporate the result of the search at the current iteration into the pattern for the next iteration. This is done using Algorithm 5. Note the distinguished role of c_k^p, the last column of C_k, which represents the pattern step $(x_k - x_{k-1})$.

```
ALGORITHM 5. UPDATING $C_k$.
For $i = 3^n + 1, \ldots, 2 \cdot 3^n$ do
        $c_{k+1}^i = c_k^i + (1/\Delta_k) s_k - c_k^p$.
Return.
```

Since $(1/\Delta_k) s_k$ is necessarily a column of C_k and $C_0 \in \mathbf{Z}^{n \times p}$, an argument by induction shows that the update algorithm for C_k ensures that the columns of C_k always consist of integers.

4.3.2. The exploratory moves. In Algorithm 6, the e_i's denote the unit coordinate vectors and c_k^p denotes the last column of C_k. We set $\rho_{-1} = 0$ so that ρ_{k-1} is defined when $k = 0$.

A useful example for working through the logic of the algorithm can be found in [1], though the presentation and notation differ somewhat from that given here.

```
ALGORITHM 6. EXPLORATORY MOVES ALGORITHM FOR HOOKE AND JEEVES.
Given $x_k$, $\Delta_k$, $f(x_k)$, $B$, and $\rho_{k-1}$, set $\rho_k = \rho_{k-1}$ and min $= f(x_k)$.
If $\rho_k > 0$ then set $s_k = \Delta_k B c_k^p$, $\rho_k = f(x_k) - f(x_k + s_k)$, and min $= f(x_k + s_k)$.
```

```
    For $i = 1, \ldots, n$ do
        (a)$s_k^i = s_k + \Delta_k B e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.
        (b)If $f(x_k^i) <$ min then $\rho_k = f(x_k) - f(x_k^i)$, min $= f(x_k^i)$, and $s_k = s_k^i$.
            Otherwise,
                (i) $s_k^i = s_k - \Delta_k B e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.
                (ii)If $f(x_k^i) <$ min then $\rho_k = f(x_k) - f(x_k^i)$, min $= f(x_k^i)$, and $s_k = s_k^i$.
If $\rho_k \leq 0$ then set $s_k = 0$, $\rho_k = 0$, and min $= f(x_k)$.
    For $i = 1, \ldots, n$ do
        (a)$s_k^i = s_k + \Delta_k B e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.
        (b)If $f(x_k^i) <$ min then $\rho_k = f(x_k) - f(x_k^i)$, min $= f(x_k^i)$, and $s_k = s_k^i$.
            Otherwise,
                (i) $s_k^i = s_k - \Delta_k B e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.
                (ii)If $f(x_k^i) <$ min then $\rho_k = f(x_k) - f(x_k^i)$, min $= f(x_k^i)$, and $s_k = s_k^i$.
Return.
```

All possible steps are contained in $\Delta_k P_k$ since C_k contains columns that represent the "pattern steps" tried at the beginning of the iteration. And, once again, the exploratory moves given in Algorithm 6 examine all $2n$ steps defined by $\Delta_k B \Gamma$ unless a step satisfying $f(x_k + s_k) < f(x_k)$ is found.

Since we have shown that the pattern search algorithm of Hooke and Jeeves satisfies all the necessary requirements, we can therefore conclude that it, too, is a special case of the generalized pattern search method and Theorem 3.5 holds.

4.4. Multidirectional search.

The multidirectional search algorithm was introduced by Dennis and Torczon in 1989 [15] as a first step towards a general purpose optimization algorithm with promising properties for parallel computation. While subsequent work led to a class of algorithms (based on the multidirectional search algorithm) that allows for more flexible computation [6, 17], one of the unanticipated results of the original research was a global convergence theorem for the multidirectional search algorithm [16].

The multidirectional search algorithm is a simplex-based algorithm. The pattern of points can be expressed as a simplex (i.e., $n + 1$ points or vertices) based at the current iterate; as such, multidirectional search owes much in its conception to its predecessors, the simplex design algorithm of Spendley, Hext, and Himsworth [12] and the simplex algorithm of Nelder and Mead [9]. However, multidirectional search is a different algorithm—particularly from a theoretical standpoint. Convergence for the Spendley, Hext, and Himsworth algorithm can be shown only with some modification of the original algorithm, and then only under the additional assumption that the function f is convex. There are numerical examples to demonstrate that the Nelder–Mead simplex algorithm may fail to converge to a stationary point of the function because the uniform linear independence property (discussed in section 6.2), which plays a key role in the convergence analysis, cannot be guaranteed to hold [15].

The multidirectional search algorithm is described in detail in both [6] and [16]. The formulation given here is different and, in fact, introduces some redundancy that can be eliminated when actually implementing the algorithm. However, the way of expressing the algorithm that we use here allows us to make clear the similarities between this and other pattern search methods.

4.4.1. The matrices.

It is most natural to express multidirectional search in terms of multiple basis matrices B_k and a fixed generating matrix C, which is at odds with our definition for generalized pattern search methods. As we shall see, however, it is possible to convert the more natural specification to one that conforms to our requirements for a pattern search method.

The multidirectional search algorithm centers around a family of basis matrices $\mathbf{B}$ that consists of all matrices representing the edges adjacent to each vertex in a nondegenerate n-dimensional simplex that the user is allowed to specify. Since the ordering of the columns is not unique and typically not preserved in the implementation of the method, we consider all possible representations of the columns of the matrices associated with the edges adjacent to the $(n+1)$ vertices of the simplex. We then add the negatives of these $(n+1)!$ basis matrices to account for the effect of the *reflection* step allowed by the multidirectional search algorithm. Thus the cardinality of the set $\mathbf{B}$ is $|\mathbf{B}| = 2(n+1)!$.

Fortunately, there is no need to construct this unwieldy number of basis matrices to initialize the method. We can update the basis matrix after each iteration k by reconstructing the new basis matrix B_{k+1}, given the outcome of the exploratory moves, from the trial points x_k^i, $i = 1, \ldots, n$, considered during the course of the exploratory moves. This procedure is given in Algorithm 7. The scalar *scale* is chosen during the course of the exploratory moves (see Algorithm 8) to ensure that $B_{k+1} \in \mathbf{B}$ by factoring out any change in the size of the simplex introduced by a change in Δ_k. This has the further effect of preserving the role of Δ_k as a step length parameter.

ALGORITHM 7. UPDATING B_k.

```
Given $B_k = [b_k^1 \cdots b_k^i \cdots b_k^n]$, $scale$, $best$, and $x_k^i$ for $i = 0, \ldots, n$,
If $\rho_k > 0$ then
   For $i = 0, \ldots, (best - 1)$ do
      $b_{k+1}^{i+1} = scale * (x_k^i - x_k^{best})$.
   For $i = (best + 1), \ldots, n$ do
      $b_{k+1}^{i} = scale * (x_k^i - x_k^{best})$.
Otherwise
   For $i = 1, \ldots, n$ do
      $b_{k+1}^i = b_k^i$.
Return.
```

Given this use of a family of basis matrices to help define the multidirectional search algorithm, the generating matrix is then the fixed matrix $C = [I \ -I \ -\mu I \ \ 0]$. Thus, C contains $p = 3n + 1$ columns, with $M = I$. To ensure that $C \in \mathbf{Z}^{n \times p}$, we require $\mu \in \mathbf{Z}$. Furthermore, to ensure that the role of Δ_k as a step length parameter is not lost with the introduction of the *expansion* step represented by $-\mu I$, we require $\mu \in \Lambda$. The algorithm is defined so that $\Lambda = \{\tau^{w_1}, \tau^{w_2}\}$, with $\mu = \tau^{w_2}$. This requires the further restriction that $\tau \in \mathbf{N}$. Again, this is not an onerous restriction. Multidirectional search usually is specified so that $\tau = 2$, $w_2 = 1$, and thus $\mu = 2$.

Now, to bring this notation into conformity with our definition for a generalized pattern search method, observe that we can represent all possible basis matrices $B_\nu \in \mathbf{B}$ in terms of a single reference matrix $B \in \mathbf{B}$ so that

$$B_\nu \quad = \quad B\hat{B}_\nu, \qquad \nu \quad = \quad 1, \ldots, |\mathbf{B}|.$$

A convenient feature of using the edges of a simplex to form the set of basis matrices is that the matrices $\hat{B}_\nu$ consist only of elements from the set $\{-1, 0, 1\}$. The matrices $\hat{B}_\nu$ are necessarily nonsingular because of the nondegeneracy of the simplex. We use $\hat{\mathbf{B}}$ to represent the set of matrices $\hat{B}_\nu$ and observe that since $\mathbf{B}$ is a finite set, the set $\hat{\mathbf{B}}$ is also finite.

We then observe that

$$\begin{aligned} P_k = \quad B_k C \quad &= \quad B_k \quad [I \quad -I \quad -\mu I \quad 0] \\ &\equiv \quad B \quad [\hat{B}_k \quad -\hat{B}_k \quad -\mu \hat{B}_k \quad 0] \quad = \quad BC_k. \end{aligned}$$

Thus we can define the pattern in terms of the single reference matrix B and the redefined generating matrix

$$C_k \quad \equiv \quad [\hat{B}_k \quad -\hat{B}_k \quad -\mu \hat{B}_k \quad 0],$$

with $M_k \equiv \hat{B}_k$ and $\mathbf{M} \equiv \hat{\mathbf{B}}$. We also have $L_k \equiv [-\mu \hat{B}_k \;\; 0]$ and since $\mu \in \mathbf{Z}$, $L_k \in \mathbf{Z}^{n \times (n+1)}$, as required.

4.4.2. The exploratory moves. The exploratory moves for the multidirectional search method are given in Algorithm 8; the e_i's denote the unit coordinate vectors. We use the notion of $B_k \in \mathbf{B}$ for consistency with the update algorithm given in Algorithm 6, but we could just as easily substitute $B\hat{B}_k$ for B_k in the algorithm given below.

Algorithm 8. Exploratory Moves Algorithm for Multidirectional Search.

```
Given $x_k$, $\Delta_k$, $f(x_k)$, $B_k$, and $\mu = \tau^{w_2} \in \mathbf{N}$, set $s_k = 0$, $\rho_k = 0$, min $= f(x_k)$,
$\lambda_k = 1$, $scale = 1/\Delta_k$, $best = 0$, and $x_k^0 = x_k$.
For $i = 1, \ldots, n$ do
  (a) $s_k^i = \Delta_k B_k e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.
  (b) If $f(x_k^i) <$ min then $\rho_k = f(x_k) - f(x_k^i)$, min $= f(x_k^i)$, $s_k = s_k^i$, and $best = i$.
If $\rho_k \leq 0$ then
  For $i = 1, \ldots, n$ do
    (a) $s_k^i = -\Delta_k B_k e_i$ and $x_k^i = x_k + s_k^i$. Compute $f(x_k^i)$.
    (b) If $f(x_k^i) <$ min then $\rho_k = f(x_k) - f(x_k^i)$, min $= f(x_k^i)$, $s_k = s_k^i$, and $best = i$.
  If $\rho_k > 0$ then set $scale = 1/\mu\Delta_k$.
    For $i = 1, \ldots, n$ do
      (a) $s_k^i = -\mu\Delta_k B_k e_i$ and $x_k^i = x_k + s_k$. Compute $f(x_k^i)$.
      (b) If $f(x_k^i) <$ min then $\rho_k = f(x_k) - f(x_k^i)$, min $= f(x_k^i)$, $s_k = s_k^i$, $best = i$,
          and $\lambda_k = \mu$.
Return.
```

Clearly, $s_k \in \Delta_k P_k$. Since the exploratory moves algorithm considers all steps of the form $\Delta_k B \Gamma_k$, unless simple decrease is found after examining only the steps defined by $\Delta_k B M_k$, this guarantees we satisfy the condition that if $\min\{f(x_k+y), y \in \Delta_k B \Gamma_k\} < f(x_k)$, then $f(x_k + s_k) < f(x_k)$.

4.4.3. Updating the step length. The algorithm for updating Δ_k is that given in Algorithm 2. In this case, while θ usually is set to $1/2$ so that $\tau = 2$, $w_0 = -1$, and $w_1 = 0$, we also include an expansion factor $\mu = \tau^{w_2}$, where w_2 usually equals one. Thus $\Lambda = \{1, \mu\}$, where μ is usually 2. The choice of $\lambda_k \in \Lambda$ is made during the execution of the exploratory moves.

Since we have shown that the multidirectional search algorithm satisfies all the necessary requirements, we conclude that it is also a pattern search method and thus Theorem 3.5 applies. Note that since we allow $\mu > 1$, which is a useful algorithmic feature, we cannot guarantee that $\lim_{k \to +\infty} \Delta_k = 0$ and so Theorem 3.7 does not automatically apply.

5. Conclusions.

We have presented a framework in which one can analyze pattern search methods. This framework abstracts and quantifies the similarities of the classical pattern search methods and enables us to prove $\liminf_{k\to+\infty} \|\nabla f(x_k)\| = 0$ for this class of algorithms. We also specify the conditions under which the limit $\lim_{k\to+\infty} \|\nabla f(x_k)\| = 0$ can be shown to hold.

These convergence results are perhaps surprising, given the simplicity of pattern search methods, but derive from the algebraic rigidity imposed on the iterates produced by pattern search methods. This is gratifying, since while this rigidity originally was introduced as a heuristic for directing the exploratory moves, it turns out to be the key to proving convergence as well. This analysis also highlights just how weak the conditions on the acceptance of the step can be and yet still allow a global convergence analysis, an observation that may prove useful in the analysis of other classes of optimization methods.

6. Technical results.

We deferred the proof of Proposition 3.4 for several reasons. First, many of the results in this section are generalizations of similar results to be found in [16]. The abstraction in section 2 leads to more succinct proofs. Second, the proof of Proposition 3.4 is closely related to that of several other results presented in this section and requires us to introduce several additional notions.

We return to our definition of the pattern as $P_k = BC_k$ to show that the pattern contains at least one direction of descent whenever $\nabla f(x_k) \neq 0$.

Recall that we require the columns of C_k to contain both M_k and $-M_k$. Thus, P_k can be partitioned as follows:

$$P_k \quad = \quad BC_k \quad = \quad B[M_k \quad -M_k \quad L_k] \quad = \quad B[\Gamma_k \quad L_k].$$

We now elaborate on these requirements. Since M_k is an $n \times n$ nonsingular matrix and B is nonsingular, we are guaranteed that BM_k forms a basis for $\mathbf{R}^n$. Further, we are guaranteed that at any iteration k, if $\nabla f(x_k) \neq 0$, $x_k - Bc_k^i$ is a direction of descent for at least one column c_k^i contained in the block Γ_k.

6.1. Descent methods.

Of course, the existence of a trial step in a descent direction is not sufficient to guarantee that decrease in the value of the objective function will be realized. To guarantee that a pattern search method is a descent method, we need to guarantee that in a finite number of iterations the method produces a positive step size Δ_k that achieves decrease on the objective function at the current iterate. We now show that this is the case.

LEMMA 6.1. *Suppose that f is continuously differentiable on a neighborhood of $L(x_0)$. If $\nabla f(x_k) \neq 0$, then there exists $q \in \mathbf{Z}$, $q \geq 0$ such that $\rho_{k+q} > 0$ (i.e., the $(k+q)$th iteration is* successful*).*

Proof. A key hypothesis placed on the exploratory moves is that if descent can be found for one of the trial steps defined by $\Delta_k B\Gamma_k$, then the exploratory moves returns a step that produces descent.

Because BC_k has rank n, if $\nabla f(x_k) \neq 0$, then there exists at least one trial direction $d_k^i = x_k - Bc_k^i$, where $c_k^i \in \Gamma_k$, such that $\nabla f(x_k)^T d_k^i \neq 0$. But, since $-c_k^i \in \Gamma_k$, $\nabla f(x_k)^T d_k^i < 0$ without loss of generality. Thus, there exists an $h_k > 0$ such that for $0 < h \leq h_k$, $f(x_k + hd_k^i) < f(x_k)$.

If at iteration k, $\Delta_k > h_k$, then the iteration may be unsuccessful; that is, $\rho_k = f(x_k) - f(x_k + s_k) \leq 0$. When the iteration is unsuccessful, the generalized pattern search method sets $x_{k+1} = x_k$ and the updating algorithm sets $\Delta_{k+1} = \theta\Delta_k$. Since θ is strictly less than one, there exists $q \in \mathbf{Z}$, $q \geq 0$, such that $\theta^q \Delta_k \leq h_k$. Thus we are guaranteed descent, i.e., a successful iteration, in at most q iterations. □

6.2. Uniform linear independence. The pattern P_k guarantees the existence of at least one direction of descent whenever $\nabla f(x_k) \neq 0$. We now want to guarantee the existence of a bound on the angle between the direction of descent contained in $B\Gamma_k$ and the negative gradient at x_k (whenever $\nabla f(x_k) \neq 0$). We will show, in fact, that this bound is uniform across all iterations of the pattern search algorithm. To do so, we use the notion of *uniform linear independence* [10].

LEMMA 6.2. *For a pattern search algorithm, there exists a constant $\xi > 0$ such that for all $k \geq 0$ and $x \neq 0$,*

$$\max \left\{ \frac{|x^T(x_k^i - x_k)|}{\|x\| \|x_k^i - x_k\|}, i = 1, \ldots, p \right\} \geq \xi. \tag{13}$$

Proof. To demonstrate the existence of ξ, we first consider the simplest possible case, $B = I$ and $C = [M \ -M \ \ 0] = [I \ -I \ \ 0]$, and use this to derive a bound for any choice of B and C_k that satisfies the conditions we have imposed.

LEMMA 6.3. *Suppose $\|y\| = 1$. Define $\theta(y) \in [0, \pi/2]$ by*

$$\cos\theta(y) = \max_{1 \leq j \leq n} \left\{ |y^T e_j| \right\},$$

where the e_j's are the unit coordinate vectors.

If $B = I$ and $C = [I \ -I \ \ 0]$, then

$$\min_{y \in \mathbf{R}^n} \cos\theta(y) = \frac{1}{\sqrt{n}}.$$

Proof. We have $|y^T e_j| = |y_j|$, where $y = (y_1, \ldots, y_n)^T$. Since $\sum_{j=1}^n |y_j|^2 = 1$, we are guaranteed that $|y_j| \geq 1/\sqrt{n}$ for some j, so $|y^T e_j| \geq 1/\sqrt{n}$ for some j. Thus $\cos\theta(y) \geq 1/\sqrt{n}$.

Now note that $\cos\theta(y)$ attains this lower bound for any $y = \alpha_1 e_1 + \alpha_2 e_2 + \cdots + \alpha_n e_n$, where $\alpha_j = \pm 1/\sqrt{n}$. $\square$

Thus, if the pattern search is restricted to the coordinate directions defined by $P = [I \ -I \ 0]$, $\xi = 1/\sqrt{n}$ gives the lower bound on the absolute value of the cosine of the angle between the gradient and a guaranteed direction of descent. We now use the bound for this particular case to derive a bound for the general case.

Assume a general basis matrix B and a general matrix $M_k \in \mathbf{M}$, where $|\mathbf{M}| < +\infty$. We adopt the notation $BM_k = [y_k^1 \cdots y_k^n]$. Then for any $x \neq 0$ we have the following:

$$|\cos\theta| = \frac{\left|x^T y_k^j\right|}{\|x\| \|y_k^j\|} = \frac{|x^T BM_k e_j|}{\|x\| \|BM_k e_j\|} = \frac{\left|\left((BM_k)^T x\right)^T e_j\right|}{\|x\| \|BM_k e_j\|}.$$

If we set $w = (BM_k)^T x$ so that $x = (BM_k)^{-T} w$, we have

$$\begin{aligned}
|\cos\theta| &= \frac{|w^T e_j|}{\|(BM_k)^{-T} w\| \|BM_k e_j\|} \geq \frac{|w^T e_j|}{\|(BM_k)^{-T}\| \|w\| \|BM_k\| \|e_j\|} \\
&= \frac{1}{\|(BM_k)^{-T}\| \|BM_k\|} \left(\frac{|w^T e_j|}{\|w\| \|e_j\|} \right) = \frac{1}{\|(BM_k)^{-1}\| \|BM_k\|} \left(\frac{|w^T e_j|}{\|w\| \|e_j\|} \right) \\
&\geq \frac{1}{\kappa(BM_k)} \frac{1}{\sqrt{n}},
\end{aligned}$$

where $\kappa(BM_k)$ is the condition number of the matrix BM_k. Thus, we have

$$|\cos\theta| \geq \frac{1}{\kappa(BM_k)\sqrt{n}} > 0\,.$$

To ensure a bound ξ that is independent of the choice of any particular matrix $M \in \mathbf{M}$, we simply observe that the set $\mathbf{M}$ is required to be finite. Thus, ξ is taken to be

$$\xi = \min_{M\in\mathbf{M}} \left\{ \frac{1}{\kappa(BM)\sqrt{n}} \right\}. \qquad \square \tag{14}$$

The bound given in (14) points to two features that explain much about the behavior of pattern search methods. Since we never explicitly calculate—or approximate—the gradient, we are dependent on the fact that in the worst case at least one of our search directions is not orthogonal to the gradient; ξ gives us a bound on how far away we can be. Thus, as either the condition number of the product BM_k increases, or the dimension of the problem increases, our bound on the angle between the search direction and the gradient deteriorates. This suggests two things. First, we should be very careful in our choice of B and $\mathbf{M}$ for any particular pattern search method. Second, we should not be surprised that these methods become less effective as the dimension of the problem increases.

Nevertheless, even though pattern search methods neither require nor explicitly approximate the gradient of the function, the uniform linear independence condition demonstrates that the pattern search methods are, in fact, *gradient-related methods*, as defined by Ortega and Rheinboldt [10], which is one reason why we can establish global convergence.

6.3. The descent condition. Having introduced the notion of uniform linear independence with the bound ξ, we are now ready to show that pattern search methods reduce Δ_k only when necessary to find descent. To do this we will show that once the steps $s_k^i \equiv (x_k^i - x_k)$ are small enough, then a successful step must be returned by the exploratory moves algorithm. Lemma 3.1 allows us to restate this condition in terms of Δ_k. We use the result to prove Proposition 3.4.

PROPOSITION 6.4. *Suppose that $L(x_0)$ is compact and f is continuously differentiable on a neighborhood of $L(x_0)$. Given $\epsilon > 0$, let*

$$\Omega_\epsilon = \{x \in L(x_0) : dist(x, X_*) \geq \epsilon\}.$$

Suppose also that $x_0 \in \Omega_\epsilon$. Then there exists $\delta > 0$, independent of k, such that if $x_k \in \Omega_\epsilon$ and $\Delta_k < \delta$, then the kth iteration of a generalized pattern search method (see Algorithm 1) will be successful (i.e., $\rho_k = f(x_k) - f(x_k + s_k) > 0$) and thus $\Delta_{k+1} \geq \Delta_k$.

Proof. We restrict our attention to the steps defined by the columns of $\Delta_k B\Gamma_k$. This is sufficient since the Hypotheses on exploratory moves ensure that a step s_k satisfying the simple decrease condition $\rho_k > 0$ must be returned if a trial step defined by a column of $\Delta_k B\Gamma_k$ satisfies the simple decrease condition.

If s_k^i, $i = 1, \ldots, 2n$, is a step defined by $\Delta_k B\Gamma_k$ (we assume that P_k is partitioned as in (2) so that the first $2n$ columns of P_k contain the columns of $B\Gamma_k \equiv [BM_k\ -BM_k]$), then for some $\zeta^* > 0$, independent of k,

$$\|s_k^i\| \;=\; \|\Delta_k B c_k^i\| \;\leq\; \|B\|\|c_k^i\|\Delta_k \;\leq\; \zeta^*\Delta_k, \qquad i = 1, \ldots, 2n, \tag{15}$$

since $M_k \in \mathbf{M} \subset \mathbf{Z}^{n\times n}$ and $\mathbf{M}$ is a finite set of matrices. Together, (15) and Lemma 3.1 yield

$$\zeta_* \Delta_k \quad \leq \quad \|s_k^i\| \leq \quad \zeta^* \Delta_k, \qquad i = 1, \ldots, 2n.$$

Since $x_0 \in \Omega_\epsilon$, Lemma 6.1 allows us to define $N = \min\{k : x_k \neq x_0\}$. Define $d = \operatorname{dist}(L(x_N), C(x_0))$. Because $L(x_N)$ and $C(x_0)$ are compact and disjoint, we know that $d > 0$. If $\Delta_k < d/2\zeta^*$, then $\|s_k^i\| \leq \zeta^* \Delta_k < d/2$ for all $i = 1, \ldots, 2n$. Thus x_k^i lies in the interior of $L(x_0)$ for all $i = 1, \ldots, 2n$. More precisely, for all $i = 1, \ldots, 2n$, x_k^i lies in the ball $B(x_k, d/2) \subset L(x_0)$.

Let $\alpha = \min_{x\in\Omega_\epsilon} \|\nabla f(x)\|$. By design, $\alpha > 0$. Since ∇f is continuous on a neighborhood of $L(x_0)$, ∇f is uniformly continuous on a neighborhood of $L(x_0)$. Thus, there exists a constant $r > 0$, depending only on α and the ξ from (13), such that

$$\|\nabla f(x) - \nabla f(x_k)\| \quad \leq \quad \tfrac{\xi\alpha}{2} \qquad \text{whenever} \qquad \|x - x_k\| \quad \leq \quad r \qquad (\text{and } x \in L(x_0)).$$

We define

$$\delta = \frac{1}{\zeta^*} \min\left\{\frac{d}{2},\ r\right\}. \tag{16}$$

We are now assured that if

$$\Delta_k < \delta \tag{17}$$

then

$$x_k^i \in B\left(x_k, \frac{d}{2}\right) \subset L(x_0),\ i = 1, \ldots, 2n, \tag{18}$$

and

$$\|\nabla f(x_k^i) - \nabla f(x_k)\| \leq \tfrac{\xi\alpha}{2},\ i = 1, \ldots, 2n. \tag{19}$$

We are ready to argue that if at any iteration $k \geq N$, $x_k \in \Omega_\epsilon$ and (17) is satisfied, then an acceptable step will be found.

Choose a trial point x_k^i, $i = 1, \ldots, 2n$, that satisfies both $\nabla f(x_k)^T(x_k^i - x_k) < 0$ and

$$\frac{|\nabla f(x_k)^T(x_k^i - x_k)|}{\|\nabla f(x_k)\|\|x_k^i - x_k\|} \geq \xi.$$

The definitions of Ω_ϵ and the pattern P_k, together with Lemma 6.2, guarantee the existence of at least one such x_k^i.

Since (17) holds by assumption, (18) also holds. We can apply the mean value theorem to obtain $f(x_k^i) - f(x_k) = \nabla f(\omega)^T(x_k^i - x_k)$ for some $\omega \in (x_k, x_k^i)$, where

$$f(x_k^i) - f(x_k) = \nabla f(x_k)^T(x_k^i - x_k) + (\nabla f(\omega) - \nabla f(x_k))^T(x_k^i - x_k). \tag{20}$$

Consider the first term on the right-hand side of (20). Our choice of x_k^i gives us

$$\left|\nabla f(x_k)^T(x_k^i - x_k)\right| \geq \xi\|\nabla f(x_k)\|\|x_k^i - x_k\|.$$

Furthermore, since $\nabla f(x_k)^T(x_k^i - x_k) < 0$, we have

$$\nabla f(x_k)^T(x_k^i - x_k) \leq -\xi \|\nabla f(x_k)\| \|x_k^i - x_k\|. \tag{21}$$

Now consider the second term on the right-hand side of (20). The Cauchy–Schwarz inequality gives us

$$\left|(\nabla f(\omega) - \nabla f(x_k))^T(x_k^i - x_k)\right| \leq \|\nabla f(\omega) - \nabla f(x_k)\| \|x_k^i - x_k\|. \tag{22}$$

Combine (21) and (22) to rewrite (20) as

$$\begin{aligned} f(x_k^i) - f(x_k) &\leq -\xi \|\nabla f(x_k)\| \|x_k^i - x_k\| + \|\nabla f(\omega) - \nabla f(x_k)\| \|x_k^i - x_k\| \\ &= (-\xi \|\nabla f(x_k)\| + \|\nabla f(\omega) - \nabla f(x_k)\|) \|x_k^i - x_k\|. \end{aligned}$$

Since $\omega \in (x_k, x_k^i)$ and (17) holds by assumption, (19) also holds. We then have

$$f(x_k^i) - f(x_k) \leq (-\xi \|\nabla f(x_k)\| + \frac{\xi}{2} \|\nabla f(x_k)\|) \|x_k^i - x_k\| < 0. \tag{23}$$

Thus, when $\Delta_k < \delta$, $f(x_k^i) \equiv f(x_k + s_k^i) < f(x_k)$ for at least one s_k^i defined by $\Delta_k B c_k^i$, $i = 1, \ldots, 2n$. The Hypotheses on exploratory moves guarantee that if $\min\{f(x_k + y), y \in \Delta_k B \Gamma_k\} < f(x_k)$, then $f(x_k + s_k) < f(x_k)$. Thus, $\rho_k = f(x_k) - f(x_k + s_k) > 0$ and the algorithm for updating Δ_k (Algorithm 2) ensures that $\Delta_{k+1} \geq \Delta_k$. □

Proposition 6.4 guarantees that if Δ_k is small enough, a generalized pattern search method realizes simple decrease because there exists at least one step among the $2n$ steps defined by $\Delta_k B \Gamma_k$ that gives decrease as a function of the norm of the gradient at the current iterate, as shown in (23); the Hypotheses on exploratory moves then ensure that the exploratory moves algorithm must return a step that satisfies at least simple decrease. However, there are no guarantees that the step returned by an exploratory moves algorithm satisfies more than the simple decrease condition.

To tie the amount of actual decrease to the norm of the gradient, we must place much stronger conditions on the generalized pattern search method, as discussed in section 3.3.2. Once we have done so, Corollary 6.5 follows more or less immediately from Proposition 6.4.

COROLLARY 6.5. *Suppose that $L(x_0)$ is compact and f is continuously differentiable on a neighborhood of $L(x_0)$. Suppose that the columns of the generating matrix are bounded in norm and that the generalized pattern search method (Algorithm 1) enforces the Strong hypotheses on exploratory moves. Given $\epsilon > 0$, let*

$$\Omega_\epsilon = \{x \in L(x_0) : dist(x, X_*) \geq \epsilon\}.$$

Suppose also that $x_0 \in \Omega_\epsilon$. Then there exist $\delta > 0$ and $\sigma > 0$, independent of k, such that for all but finitely many k, if $x_k \in \Omega_\epsilon$ and $\Delta_k < \delta$, then

$$f(x_{k+1}) \quad \leq \quad f(x_k) - \sigma \|\nabla f(x_k)\| \|s_k\| \quad < \quad f(x_k).$$

Proof. From Proposition 6.4, (23) says that for $k \geq N = \min\{k : x_k \neq x_0\}$ (Lemma 6.1 guarantees the existence of N), there exists at least one trial step $s_k^i \in \Delta_k B \Gamma_k$ such that once $\Delta_k < \delta$, where δ is as defined in (16), we have

$$f(x_k^i) \quad \leq \quad f(x_k) - \tfrac{\xi}{2} \|\nabla f(x_k)\| \|s_k^i\| \quad < \quad f(x_k).$$

The Strong hypotheses on exploratory moves give us

$$f(x_{k+1}) \quad \leq \quad f(x_k) - \tfrac{\xi}{2}\|\nabla f(x_k)\|\|s_k^i\| \quad < \quad f(x_k).$$

Lemma 3.1 ensures that

$$f(x_{k+1}) \quad \leq \quad f(x_k) - \tfrac{\xi}{2}\zeta_* \Delta_k \|\nabla f(x_k)\| \quad < \quad f(x_k).$$

Lemma 3.6, which holds only when the columns of the generating matrix are bounded in norm, gives us

$$f(x_{k+1}) \quad \leq \quad f(x_k) - \tfrac{\xi}{2}\zeta_* \psi_* \|\nabla f(x_k)\|\|s_k\| \quad < \quad f(x_k).$$

We define $\sigma = \frac{\xi}{2}\zeta_*\psi_*$ to complete the proof. □

We now prove Proposition 3.4.

Proof. By assumption, $\liminf_{k\to+\infty} \|\nabla f(x_k)\| \neq 0$. Then we can find N_1 and $\epsilon > 0$ such that for all $k \geq N_1$, $x_k \in \Omega_\epsilon = \{x \in L(x_0) : \mathrm{dist}(x, X_*) \geq \epsilon\}$. Lemma 6.1 guarantees the existence of $N_2 = \min\{k : x_k \neq x_0\}$. Let $N = \max(N_1, N_2)$.

From Proposition 6.4 we are assured of $\delta > 0$ such that if $\Delta_k \leq \delta$, then the iteration will be successful. Given Δ_0, there exists a constant $q \in \mathbf{Z}$, $q \geq 0$, such that $\theta^q \Delta_0 \leq \delta$, where $\theta \in (0,1)$ and is as defined in the algorithm for updating Δ_k (Algorithm 2). Thus, for $k \geq N$, $\theta^{q+1}\Delta_0 < \Delta_k$.

Set $\Delta_{LB} = \theta \min(\theta^q \Delta_0,\ \Delta_1, \ldots,\ \Delta_{N-1})$. Then for all k, $\Delta_{LB} < \Delta_k$. □

Acknowledgments. This paper benefited from conversations with J. E. Dennis, Stephen Nash, Michael Trosset, Luís Vicente, and especially Michael Lewis. In particular, discussions with Michael Lewis were critical in the distillation of the abstraction for pattern search methods found in section 2 and in the development of the analytic arguments for the algebraic structure of the iterates found in section 3.1.

The review and comments made by an anonymous referee, Danny Ralph (the second referee, who agreed to reveal his identity for this acknowledgment), and Jorge Moré, the editor, are gratefully acknowledged. In particular, the observation by Danny Ralph that the pattern contains only vectors in a fixed lattice led to a more general result and a much more elegant presentation.

REFERENCES

[1] M. Avriel, *Nonlinear Programming: Analysis and Methods*, Prentice–Hall, Englewood Cliffs, NJ, 1976.

[2] G. E. P. Box, *Evolutionary operation: A method for increasing industrial productivity*, Appl. Statist., 6 (1957), pp. 81–101.

[3] M. J. Box, D. Davies, and W. H. Swann, *Non-Linear Optimization Techniques*, ICI Monograph No. 5, Oliver & Boyd, Edinburgh, 1969.

[4] J. Céa, *Optimisation: Théorie et algorithmes*, Dunod, Paris, 1971.

[5] W. C. Davidon, *Variable metric method for minimization*, SIAM J. Optim., 1 (1991), pp. 1–17. Originally published without the preface as Argonne National Laboratory Research and Development Report 5990, May, 1959.

[6] J. E. Dennis, Jr. and V. Torczon, *Direct search methods on parallel machines*, SIAM J. Optim., 1 (1991), pp. 448–474.

[7] R. Hooke and T. A. Jeeves, *"Direct search" solution of numerical and statistical problems*, J. Assoc. Comput. Mach., 8 (1961), pp. 212–229.

[8] J. J. Moré, *Recent developments in algorithms and software for trust region methods*, in Math. Programming, The State of the Art, A. Bachem, M. Grötschel, and G. Korte, eds., Springer-Verlag, Berlin, New York, 1983, pp. 256–287.

[9] J. A. Nelder and R. Mead, *A simplex method for function minimization*, Comput. J., 7 (1965), pp. 308–313.

[10] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[11] E. Polak, *Computational Methods in Optimization: A Unified Approach*, Academic Press, New York, 1971.

[12] W. Spendley, G. R. Hext, and F. R. Himsworth, *Sequential application of simplex designs in optimisation and evolutionary operation*, Technometrics, 4 (1962), pp. 441–461.

[13] W. H. Swann, *Direct search methods*, in Numerical Methods for Unconstrained Optimization, W. Murray, ed., Academic Press, New York, 1972, pp. 13–28.

[14] S. W. Thomas, *Sequential Estimation Techniques for Quasi-Newton Algorithms*, Ph.D. thesis, Cornell University, Ithaca, NY, 1975.

[15] V. Torczon, *Multi-Directional Search: A Direct Search Algorithm for Parallel Machines*, Ph.D. thesis, Department of Mathematical Sciences, Rice University, Houston, TX, 1989.

[16] V. Torczon, *On the convergence of the multidirectional search algorithm*, SIAM J. Optim., 1 (1991), pp. 123–145.

[17] V. Torczon, *PDS: Direct Search Methods for Unconstrained Optimization on Either Sequential or Parallel Machines*, Tech. Report 92–9, Department of Mathematical Sciences, Rice University, Houston, TX, 1992.

[18] Y. Wen-ci, *Positive basis and a class of direct search techniques*, Scientia Sinica, Special Issue of Mathematics, 1 (1979), pp. 53–67.

SIAM J. Optim.
Vol. 7, No. 1, pp. 26–33, February 1997

THE BARZILAI AND BORWEIN GRADIENT METHOD FOR THE LARGE SCALE UNCONSTRAINED MINIMIZATION PROBLEM*

MARCOS RAYDAN†

Abstract. The Barzilai and Borwein gradient method for the solution of large scale unconstrained minimization problems is considered. This method requires few storage locations and very inexpensive computations. Furthermore, it does not guarantee descent in the objective function and no line search is required. Recently, the global convergence for the convex quadratic case has been established. However, for the nonquadratic case, the method needs to be incorporated in a globalization scheme. In this work, a nonmonotone line search strategy that guarantees global convergence is combined with the Barzilai and Borwein method. This strategy is based on the nonmonotone line search technique proposed by Grippo, Lampariello, and Lucidi [*SIAM J. Numer. Anal.*, 23 (1986), pp. 707–716]. Numerical results to compare the behavior of this method with recent implementations of the conjugate gradient method are presented. These results indicate that the global Barzilai and Borwein method may allow some significant reduction in the number of line searches and also in the number of gradient evaluations.

Key words. unconstrained optimization, nonmonotone line search, Barzilai and Borwein method, conjugate gradient method

AMS subject classifications. 49M07, 49M10, 90C06, 65K

PII. S1052623494266365

1. Introduction.

In this paper we consider the Barzilai and Borwein gradient method for the large scale unconstrained minimization problem

$$\min_{x \in R^n} f(x), \tag{1}$$

where $f : R^n \to R$. The method is defined by

$$x_{k+1} = x_k - \frac{1}{\alpha_k} g_k, \tag{2}$$

where g_k is the gradient vector of f at x_k and the scalar α_k is given by

$$\alpha_k = \frac{s_{k-1}^t y_{k-1}}{s_{k-1}^t s_{k-1}}, \tag{3}$$

where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$.

Every iteration of the Barzilai and Borwein method requires only $O(n)$ floating point operations and a gradient evaluation. No matrix computations and no line searches are required during the process. The search direction is always the negative gradient direction, but the choice of steplength is not the classical choice of the steepest descent method. In fact, Barzilai and Borwein [1] observed that this new choice of steplength required less computational work and greatly speeded up the convergence of the gradient method for quadratics.

* Received by the editors April 15, 1994; accepted for publication (in revised form) May 16, 1995. This research was supported in part by CDCH-UCV project 03.13.0034.93 and by NSF grant CHE-9301120.

http://www.siam.org/journals/siopt/7-1/26636.html

† Facultad de Ciencias, Universidad Central de Venezuela, Ap. 47002, Caracas 1041-A, Venezuela (mraydan@conicit.ve).

More interesting from a theoretical point of view is that the method does not guarantee descent in the objective function. Barzilai and Borwein [1] presented a convergence analysis in the two-dimensional quadratic case. They established, for that particular case, R-superlinear convergence. However, Fletcher [5] argued that, in general, only R-linear convergence should be expected. Later, Raydan [18] established global convergence for the strictly convex quadratic case with any number of variables. This result has been recently extended to the (not necessarily strictly) convex quadratic case by Friedlander, Martinez, and Raydan [6] to incorporate the method in a box constrained optimization technique.

Glunt, Hayden, and Raydan [9] established a relationship with the shifted power method that adds understanding to the significant improvement obtained with the choice of steplength given by (3). In particular, they applied the Barzilai and Borwein gradient method to find local minimizers of nonquadratic functions that appear in the determination of molecular structures from nuclear magnetic resonance data. For this application, it was possible to choose good starting values and convergence was observed. However, in general, for the nonquadratic case the method needs to be incorporated in a globalization scheme.

The object of this work is to embed the Barzilai and Borwein gradient method in a globalization strategy that accepts the steplength given by (3) as frequently as possible and that only requires storage of first-order information during the process.

This paper is organized as follows. In section 2 we present a globalization strategy suitable for the Barzilai and Borwein method. This strategy is based on the nonmonotone line search technique of Grippo, Lampariello, and Lucidi [10] for Newton's method. We discuss the properties of this new algorithm and establish global convergence under mild assumptions. In section 3 we present some preliminary numerical results to compare the behavior of our global new method with recent implementations of the conjugate gradient method for the nonquadratic case. Finally, in section 4 we present some concluding remarks.

2. Globalization strategy. Standard methods for the solution of (1) usually generate a sequence of iterates for which a sufficient decrease in the objective function f is enforced at every iteration. In many cases, the global strategy consists of accepting the steplength, in the search direction, if it satisfies the well-known Armijo–Goldstein–Wolfe conditions. Practical line search schemes have been developed to enforce these conditions when combined with Newton, quasi-Newton, and conjugate gradient methods. For a complete discussion on this topic see [3], [4], and [14].

There are some disadvantages to forcing the Armijo–Goldstein–Wolfe conditions when combined with the Barzilai and Borwein gradient method. One of the disadvantages is that forcing decrease at every iteration will destroy some of the local properties of the method. As it is argued in Fletcher [5] and Glunt, Hayden, and Raydan [9], the choice of steplength (3) is related to the eigenvalues of the Hessian at the minimizer and not to the function value. Moreover, since the search direction is always the negative gradient direction, forcing decrease at every iteration will reduce the method to the steepest descent method, which is known for being slow.

Therefore, we will enforce a much weaker condition of the form

$$f(x_{k+1}) \leq \max_{0\leq j\leq M} f(x_{k-j}) + \gamma g_k^t(x_{k+1} - x_k), \tag{4}$$

where M is a nonnegative integer and γ is a small positive number. This type of condition (4) was introduced by Grippo, Lampariello, and Lucidi [10] and successfully applied to Newton's method for a set of test functions. Recently, the same type of nonmonotone line search technique has been incorporated into a variety of optimization algorithms. See, for instance, [11], [15], and [16]. Condition (4) allows the objective function to increase at some iterations and still guarantees global convergence. This feature fits nicely with the nonmonotone behavior of the Barzilai and Borwein gradient method. We now present the proposed algorithm.

Global Barzilai and Borwein (GBB) Algorithm.

```
Given $x_0$, $\alpha_0$, integer $M \geq 0$, $\gamma \in (0,1)$, $\delta > 0$,
$0 < \sigma_1 < \sigma_2 < 1$,   $0 < \epsilon < 1$. Set $k = 0$.

Step 1: If $\|g_k\|$=0 stop

Step 2: If $\alpha_k \leq \epsilon$ or $\alpha_k \geq 1/\epsilon$ then set $\alpha_k = \delta$

Step 3: Set $\lambda = 1/\alpha_k$

Step 4: (nonmonotone line search)
If $f(x_k - \lambda g_k) \leq \max_{0 \leq j \leq \min(k,M)}(f_{k-j}) - \gamma\lambda g_k^t g_k$
then set $\lambda_k = \lambda$, $x_{k+1} = x_k - \lambda_k g_k$, and go to Step 6

Step 5: Choose $\sigma \in [\sigma_1, \sigma_2]$, set $\lambda = \sigma\lambda$, and go to Step 4

Step 6: Set $\alpha_{k+1} = -(g_k^t y_k)/(\lambda_k g_k^t g_k)$, $k = k+1$, and go to Step 1.
```

Remarks. (1) The object of Step 2 is to avoid uphill directions and to keep the sequence $\{\lambda_k\}$ uniformly bounded. In fact, for all k

$$0 < \min\left(\epsilon, \frac{1}{\delta}\right) \leq \lambda_k \leq \max\left(\frac{1}{\epsilon}, \frac{1}{\delta}\right).$$

(2) The GBB algorithm cannot cycle indefinitely between Steps 4 and 5. Indeed, since $\lambda g_k^t g_k > 0$ and $\gamma < 1$, for sufficiently small values of λ the condition in Step 4 is satisfied.

(3) Since $s_k = -\lambda_k g_k$, then the definition of α_{k+1} given in Step 6 is equivalent to the one given in (3). The advantage of the definition used in the algorithm is that it avoids the storage of the vector s_k and reduces to $3n$ locations the storage requirements of the GBB algorithm.

(4) For $k = 0$ the condition in Step 4 reduces to the Armijo α condition. For $k > 0$ the objective function might increase at some iterations. However, $f(x_k) \leq f(x_0)$ for all k, and so the level set $\{x : f(x) \leq f(x_0)\}$ contains the entire sequence of iterates $\{x_k\}$.

The convergence properties of the GBB algorithm are stated in the following theorem.

Theorem 2.1. *Assume that $\Omega_0 = \{x : f(x) \leq f(x_0)\}$ is a bounded set. Let $f : R^n \to R$ be continuously differentiable in some neighborhood N of Ω_0. Let $\{x_k\}$ be the sequence generated by the GBB algorithm. Then either $g(x_j) = 0$ for some finite j, or the following properties hold:*

(i) $\lim_{k\to\infty} \|g_k\| = 0$;

(ii) *no limit point of $\{x_k\}$ is a local maximum of f;*

(iii) *if the number of stationary points of f in Ω_0 is finite, then the sequence $\{x_k\}$ converges.*

Proof. In order to establish (i), we make use of the first part of the proof of the convergence theorem in [10, p. 709].

Let us define $m(k) = \min(k, M)$. Clearly, $m(0) = 0$ and

$$0 \leq m(k) \leq \min(m(k-1)+1, M) \quad \text{for } k \geq 1.$$

Moreover, there exists a positive constant a such that $0 < \lambda_k \leq a$ for all k. Indeed, in the GBB algorithm $a = \max(\epsilon^{-1}, \delta^{-1})$. Finally, there exist positive numbers c_1 and c_2 such that the search direction d_k satisfies $g_k^t d_k \leq -c_1 \|g_k\|_2^2$ and $\|d_k\|_2 \leq c_2 \|g_k\|_2$. In fact, in the GBB algorithm, the search direction d_k is $-g_k$ for all k and so $c_1 = c_2 = 1$. Therefore, we obtain equation (14) in [10, p. 711] that in our case reduces to

$$\lim_{k \to \infty} \lambda_k \|g_k\|_2 = 0.$$

Since $\lambda_k \geq \min(\epsilon, \frac{1}{\delta})$ for all k, then part (i) holds. Assertions (ii) and (iii) follow directly from the convergence theorem in [10]. □

Notice that, forcing the weak condition (4), the sequence $\{x_k\}$ generated by the GBB algorithm has the following property:

$$\lim_{k \to \infty} \|g_k\| = 0.$$

This is in sharp contrast to the conjugate gradient methods (Fletcher–Reeves, Polak–Ribière, etc.) for which much stronger conditions have to be imposed to obtain the weaker result:

$$\liminf_{k \to \infty} \|g_k\| = 0.$$

For a further discussion on the convergence properties of the conjugate gradient methods see Nocedal [14] and Gilbert and Nocedal [8].

3. Numerical results.

In this section we present numerical results to compare the behavior of the GBB algorithm with two different implementations of the conjugate gradient method for the nonquadratic case. In particular, we compare the GBB algorithm with the well-known routine CONMIN of Shanno and Phua [19], which includes automatic restarts and requires $7n$ storage locations. We also compare the GBB algorithm with the Polak–Ribière implementation of Gilbert and Nocedal [8] (PR^+) that requires $4n$ storage locations and for which global convergence was established under mild assumptions. The line search for the PR^+ method is based on the algorithm of Moré and Thuente [13] and is fully described in [8]. It is worth mentioning that for the classical Polak–Ribière method no satisfactory global convergence result has been found and a negative convergence result has been established; see Powell [17]. For this lack of theoretical support we have decided to compare the new algorithm with PR^+ instead of the classical Polak–Ribière method, although they behave similarly in practice.

The problems used in our tests include well-known large functions and two new strictly convex functions. Table 1 lists the problems and the references for descriptions of the test functions and the starting points. For the problems of Moré, Garbow, and Hillstrom [12] we use the standard starting vector. In this paper, we only describe the new functions:

Strictly Convex 1:

$$f(x) = \sum_{i=1}^{n} (e^{x_i} - x_i); \qquad x_0 = \left(\frac{1}{n}, \ldots, \frac{i}{n}, \ldots, 1\right)^t.$$

Table 1
Test problems.

Problem	Name	Reference
1	Strictly Convex 1	
2	Strictly Convex 2	
3	Brown almost linear	Moré et al. [12]
4	Trigonometric	Moré et al. [12]
5	Broyden tridiagonal	Moré et al. [12]
6	Oren's power	Garg and Tapia [7]
7	Extended Rosenbrock	Moré et al. [12]
8	Penalty 1	Moré et al. [12]
9	Tridiagonal 1	Buckley and LeNir [2]
10	Variably dimensioned	Moré et al. [12]
11	Extended Powell	Moré et al. [12]
12	Generalized Rosenbrock	Moré et al. [12]
13	Extended ENGLV1	Toint [20]
14	Extended Freudenstein and Roth	Toint [20]
15	Wrong Extended Wood	Toint [20]

Strictly Convex 2:

$$f(x) = \sum_{i=1}^{n} \frac{i}{10}(e^{x_i} - x_i); \quad x_0 = (1, 1, \ldots, 1)^t.$$

Clearly, the unique minimizer of Strictly Convex 1 and Strictly Convex 2 is given by $x_\star = (0, \ldots, 0)^t$. The Hessian of Strictly Convex 1 at $x_\star$ is the identity matrix, and the Hessian of Strictly Convex 2 at $x_\star$ has n distinct eigenvalues.

All the experiments were run on a SparcStation 1 in double precision FORTRAN with a machine epsilon of about 2×10^{-16}. For the GBB algorithm we used $\gamma = 10^{-4}$, $\epsilon = 10^{-10}$, $\sigma_1 = 0.1$, $\sigma_2 = 0.5$, $\alpha_0 = 1$, and $M = 10$. We have chosen the parameter ϵ to be a very small number in order to accept the Barzilai and Borwein step as many times as possible. However, if the condition in Step 2 was satisfied at iteration k, then the parameter δ was chosen in the following way:

$$\delta = \begin{cases} 1 & \text{if} \quad \|g_k\|_2 > 1, \\ \|g_k\|_2^{-1} & \text{if} \quad 10^{-5} \leq \|g_k\|_2 \leq 1, \\ 10^5 & \text{if} \quad \|g_k\|_2 < 10^{-5}. \end{cases}$$

Notice that, with this choice of δ, the sequence $\{\lambda_k\}$ remains uniformly bounded. In Step 5, σ is chosen by means of a quadratic interpolation described in [3, p. 127]. All runs were stopped when

$$\|g_k\|_2 \leq 10^{-6}(1 + |f(x_k)|),$$

and we verified that the three methods converged to the same solution point.

The numerical results are shown in Table 2. We report number of iterations (IT), CPU time in seconds (Time), number of function evaluations (f), number of gradient evaluations (g), and number of line searches (LS) required by the GBB method during the process, i.e., number of iterations for which the GBB algorithm goes through Step 5 at least once. Every time GBB requires a line search, it needs additional function evaluations and no additional gradient evaluations. On the other hand, CONMIN and PR^+ require a line search at every iteration and as many function

Table 2
Results for GBB, CONMIN, and PR^+.

	GBB					CONMIN			PR^+		
P(n)	IT	f	g	LS	Time	IT	f-g	Time	IT	f-g	Time
1(100)	8	8	8	0	0.04	15	38	0.16	6	17	0.14
1(1000)	8	8	8	0	0.28	15	38	1.23	7	22	0.92
1(10000)	8	8	8	0	2.8	15	38	12.57	7	22	8.45
2(100)	52	57	52	4	0.29	40	81	0.55	33	69	0.45
2(500)	74	80	74	5	1.61	63	127	4.1	44	92	2.34
2(1000)	82	91	82	7	3.49	71	145	9.13	40	84	4.17
3(100)	3	3	3	0	0.01	3	7	0.03	2	4	0.03
3(1000)	4	4	4	0	0.1	15	38	0.94	F	F	F
3(10000)	57	72	57	10	25.5	17	41	22.3	F	F	F
4(100)	76	81	76	4	0.86	51	108	1.33	54	121	1.1
4(1000)	93	106	93	13	9.6	53	112	13.1	58	132	10.1
4(10000)	89	99	89	10	83.3	59	126	134.	61	133	97.
5(100)	34	34	34	0	0.13	33	67	0.34	31	70	0.42
5(1000)	40	40	40	0	1.1	38	75	3.8	32	75	3.47
5(3000)	44	45	44	1	3.7	35	71	10.8	31	71	9.82
6(100)	105	112	105	7	0.32	49	99	0.3	39	87	0.25
6(1000)	310	378	310	54	6.8	158	320	10.8	114	236	6.42
6(10000)	1351	1750	1351	263	325.	464	937	365.	355	719	207.
7(100)	69	91	69	15	0.22	19	47	0.12	25	73	0.24
7(1000)	93	118	93	20	1.73	30	73	1.92	23	70	1.45
7(10000)	70	92	70	11	14.3	28	69	16.3	20	64	13.9
8(100)	48	49	48	1	0.16	27	65	0.23	53	204	0.75
8(1000)	57	57	57	0	1.22	25	55	1.44	40	164	4.15
8(10000)	62	62	62	0	14.2	25	55	15.	40	164	43.2
9(100)	167	191	167	18	0.55	80	161	0.8	78	158	0.7
9(1000)	878	1152	878	186	21.3	306	613	25.8	295	593	18.3
10(100)	38	38	38	0	0.13	13	29	0.1	7	39	0.16
10(1000)	54	54	54	0	1.22	27	62	1.54	F	F	F
11(100)	740	988	740	136	3.5	47	95	0.48	190	434	1.6
11(1000)	815	1125	815	163	32.6	43	87	4.1	99	238	10.6
12(100)	1429	1869	1429	342	4.85	254	516	2.3	258	533	2.63
12(500)	4452	5622	4452	1087	51.	1082	2280	45.3	1072	2162	39.7
13(100)	26	26	26	0	0.1	13	27	0.15	17	43	0.21
13(1000)	23	23	23	0	0.54	12	25	0.81	13	45	1.32
13(10000)	21	21	21	0	5.16	11	23	7.3	9	32	9.3
14(100)	438	560	438	102	2.0	13	27	0.14	14	39	0.3
14(1000)	288	377	288	69	10.3	12	25	1.13	19	50	2.4
14(10000)	119	151	119	21	44.2	11	23	11.1	8	30	16.8
15(100)	76	85	76	8	0.3	25	53	0.25	54	127	0.62
15(1000)	80	87	80	5	2.32	34	70	2.83	29	66	2.1

evaluations as gradient evaluations during the process. Hence, for CONMIN and PR^+, we report function and gradient evaluations under the label (f-g). The letter F that appears under the multicolumn PR^+ means that the run was stopped because the line search procedure failed to find a steplength. In those cases, we were not able to report any information for the PR^+ method. The results of Table 2 are summarized in Table 3. We report in Table 3 the number of problems for which each method was a winner in number of iterations, number of gradient evaluations, and CPU time.

We observe that the GBB method out performs CONMIN and PR^+ in number of gradient evaluations and CPU time, except for problems with a very ill-conditioned Hessian at the solution. For some of these problems, GBB is still competitive in

Table 3
Number of problems for which a method was a winner.

Method	IT	g	Time
GBB	1	19	22
CONMIN	17	12	10
PR^+	22	9	8

CPU time. However, if the Hessian is singular at the solution as in problem 11, then CONMIN and PR^+ clearly out perform GBB.

CONMIN and PR^+ out perform GBB in number of iterations, except for problems with a well-conditioned Hessian at the solution, in which case the number of iterations is quite similar. In some of those cases (problems 1 and 5), the difference in computing time is remarkable.

4. Concluding remarks. The Barzilai and Borwein method can be incorporated in a globalization strategy that preserves the good features of the method and only requires $3n$ storage locations. Since the search direction is always the negative gradient direction, it is trivial to ensure that descent directions are generated at every iteration. This is in sharp contrast to the conjugate gradient methods, for which a very accurate line search has to be performed at every iteration to generate descent directions.

Our numerical experiments seem to indicate that the global Barzilai and Borwein algorithm is competitive and sometimes preferable to recent and well-known implementations of the conjugate gradient method. However, further numerical investigation needs to be done to establish this conclusively.

We observe that the GBB algorithm requires few line searches. In the worst case (problem 12), it requires 1 line search out of every 5 iterations. Moreover, we have observed that near the solution the GBB method does not require any line search. At this point, we would like to stress that no local convergence analysis has been presented to support this observation. All we can say, so far, to explain the local behavior of the GBB method is that the Barzilai and Borwein method, given by (2) and (3), is globally convergent for convex quadratic functions.

Finally, we would like to comment on the choice of the parameter M in the GBB algorithm. We have tested the same set of problems with different values of M ranging from 5 to 20. In general, we observed similar results to the ones presented in Tables 2 and 3, except for problems with a singular or very ill-conditioned Hessian at the solution. For these problems, the behavior of the method is very sensitive to the choice of M. For example, using $M = 20$ in problem 11 with $n = 1000$, convergence is obtained after 365 iterations, 36 line searches, 451 function evaluations, and 12.8 seconds of execution time. These results represent a significant improvement over the ones reported in Table 2 with $M = 10$. On the other hand, using $M = 5$ the results obtained are worse than the ones in Table 2. Therefore, for the singular or very ill-conditioned case, the choice of the parameter M is a delicate issue and merits further investigation.

Acknowledgments. The author thanks Jorge Nocedal for providing all the required routines to test the PR^+ method. He also thanks Jose Mario Martinez and two anonymous referees for their constructive suggestions.

REFERENCES

[1] J. Barzilai and J. M. Borwein, *Two point step size gradient methods*, IMA J. Numer. Anal., 8 (1988), pp. 141–148.

[2] A. Buckley and A. LeNir, *QN-like variable storage conjugate gradients*, Math. Programming, 27 (1983), pp. 155–175.

[3] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[4] R. Fletcher, *Practical Methods of Optimization*, John Wiley, New York, 1987.

[5] R. Fletcher, *Low storage methods for unconstrained optimization*, in Lectures in Applied Mathematics, Vol. 26, American Mathematical Society, Providence, RI, 1990, pp. 165–179.

[6] A. Friedlander, J. M. Martinez, and M. Raydan, *A new method for large-scale box constrained convex quadratic minimization problems*, Optim. Methods and Software, 5 (1995), pp. 57–74.

[7] N. K. Garg and R. A. Tapia, *QDN: A Variable Storage Algorithm for Unconstrained Optimization*, Technical report, Department of Mathematical Sciences, Rice University, Houston, TX, 1977.

[8] J. C. Gilbert and J. Nocedal, *Global convergence properties of conjugate gradient methods for optimization*, SIAM J. Optim., 2 (1992), pp. 21–42.

[9] W. Glunt, T. L. Hayden, and M. Raydan, *Molecular conformations from distance matrices*, J. Comput. Chem., 14 (1993), pp. 114-120.

[10] L. Grippo, F. Lampariello, and S. Lucidi, *A nonmonotone line search technique for Newton's method*, SIAM J. Numer. Anal., 23 (1986), pp. 707–716.

[11] L. Grippo, F. Lampariello, and S. Lucidi, *A class of nonmonotone stabilization methods in unconstrained optimization*, Numer. Math., 59 (1991), pp. 779–805.

[12] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, *Testing unconstrained optimization software*, ACM Trans. Math. Software, 7 (1981), pp. 17–41.

[13] J. J. Moré and D. J. Thuente, *On line search algorithms with guaranteed sufficient decrease*, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1990, preprint MCS-P153-0590.

[14] J. Nocedal, *Theory of algorithms for unconstrained optimization*, Acta Numerica, 1 (1992), pp. 199–242.

[15] J. S. Pang, S. P. Han, and N. Rangaraj, *Minimization of locally lipschitzian functions*, SIAM J. Optim., 1 (1991), pp. 57–82.

[16] E. R. Panier and A. L. Tits, *Avoiding the Maratos effect by means of a nonmonotone line search* I. *General constrained problems*, SIAM J. Numer. Anal., 28 (1991), pp. 1183–1195.

[17] M. J. D. Powell, *Nonconvex minimization calculations and the conjugate gradient method*, in Lecture Notes in Mathematics, Vol. 1066, Springer-Verlag, Berlin, 1984, pp. 122–141.

[18] M. Raydan, *On the Barzilai and Borwein choice of steplength for the gradient method*, IMA J. Numer. Anal., 13 (1993), pp. 321–326.

[19] D. F. Shanno and K. H. Phua, *Remark on algorithm 500: Minimization of unconstrained multivariate functions*, ACM Trans. Math. Software, 6 (1980), pp. 618–622.

[20] Ph. L. Toint *Test Problems for Partially Separable Optimization and Results for the Routine PSPMIN*, Report Nr 83/4, Department of Mathematics, Facultés Universitaires de Namur, Namur, Belgium, 1983.

SIAM J. OPTIM.
Vol. 7, No. 1, pp. 34–46, February 1997

THE AFFINE SCALING ALGORITHM FAILS FOR STEPSIZE 0.999*

WALTER F. MASCARENHAS[†]

Abstract. We present two examples in which the dual affine scaling algorithm converges to a vertex that is not optimal if at each iteration we move 0.999 of the step to the boundary of the feasible region.

Key words. convergence, degeneracy, affine scaling algorithm

AMS subject classifications. 90C05, 68C25, 73K40

PII. S1052623493258404

1. Introduction.

This work is about the convergence of the affine scaling algorithm. We assume that the reader is familiar with this algorithm, which is a variation of the affine scaling algorithm proposed by Dikin in [Dk]. Dikin [Dk2] and Tsuchiya and Muramatsu [TM] have shown that $\bar{x}$ is optimal if $\lambda \leq 1/2$ and $\lambda \leq 2/3$, respectively, regardless of degeneracy, where λ is the fraction of the step to the boundary taken at each iteration. Saigal [S] has shown convergence to optimality for $\lambda \leq 2q/(3q-1)$, where q is the number of nonzeros in the limiting solution. In this work we present two examples in which the dual affine scaling algorithm, with $\lambda = 0.999$, converges to a vertex that is not optimal if we choose the starting point properly. The first example is simpler and contains the essence of how convergence to the wrong solution happens. However, it does not have an optimal solution and for completeness we present the second example, which has an optimal solution.

We are interested in linear programs $\Pi(A,b,c)$ of the form

$$\text{minimize } z(x) = c^t x, \qquad \text{subject to } A^t x \geq b. \tag{1}$$

The *feasible set* of Π is $\mathcal{F}_\Pi = \{x \in \mathbb{R}^n \text{ s.t. } A^t x \geq b\}$. Its interior is called $\mathcal{F}_\Pi^+$. Usually, we look at programs that have an optimal solution and for which $c \neq 0, \mathcal{F}_\Pi^+ \neq \varnothing$, and A has full rank. It follows from [MTW] that for any such programs, $\lambda \in (0,1)$, and interior starting point x^0, the dual affine scaling algorithm converges to $\bar{x}(\Pi, \lambda, x^0)$ in the relative interior of some face φ of $\mathcal{F}_\Pi$. We say that φ is nondegenerate if the restrictions active at its relative interior are linearly independent. If φ is nondegenerate, then $\bar{x}$ is optimal [Dk2].

Our examples are presented in (2) and (3). The proofs that show that the dual affine scaling algorithm with $\lambda = 0.999$ fails for these problems are rather technical, but the examples themselves are simple and we encourage the reader to perform numerical experiments with them. The experiments will show that if $x^0 = t(1, \bar{s}_2, \bar{s}_3)^t$, with $\bar{s}_2$ and $\bar{s}_3$ given by (4) and t small in the second example, then $x^{2k} \approx (0.001108633)^k x^0$ for several values of k. For a verification in higher precision, use Mathematica with decimal numbers represented as rationals ($0.999 = 999/1000$, etc.) to avoid the rounding errors

* Received by the editors October 27, 1993; accepted for publication (in revised form) June 8, 1995.

http://www.siam.org/journals/siopt/7-1/25840.html.

[†] Departamento de Matematica, Universidade Estadual de Campinas, Caixa Postal 6065, Barao Geraldo, CEP 13081, Campinas S.P., Brazil.

on the conversion to base 2. We hope that these experiments convince the practical-minded reader that $x^k \to 0$ for properly chosen x^0. In the last sections of this work we present rigorous arguments showing how to turn this empirical evidence into theorems.

The paper is organized as follows. In section 2 we present the examples and explain how we found them. In section 3 we introduce some notation. In section 4 we prove two theorems which formalize the statements above. In section 5 we prove two technical lemmas. In section 6 we prove a weak version of the stable manifold theorem, used for the rigorous analysis of the second example.

2. The examples. In this section we present the two examples and state two theorems saying that the dual affine scaling algorithm with $\lambda = 0.999$ converges to a nonoptimal solution if we choose the starting point properly. The section ends with an explanation of how we found the examples. The first example is the program Π given by

$$A = \begin{pmatrix} 0 & 0 & -1 & -1 \\ 1 & 0 & \alpha & \beta \\ 0 & 1 & \beta & \alpha \end{pmatrix}, \qquad b = (0,0,0,0)^t, \qquad c = (1,0,0)^t \tag{2}$$

with $\alpha = 0.39574487$ and $\beta = 0.91836049$. Program Π does not have an optimal solution. To fix that, we add a restriction, getting the second example, $\tilde{\Pi}$, given by

$$\tilde{A} = \begin{pmatrix} 0 & 0 & -1 & -1 & 1 \\ 1 & 0 & \alpha & \beta & -1 \\ 0 & 1 & \beta & \alpha & -1 \end{pmatrix}, \qquad \tilde{b} = (0,0,0,0,-1)^t, \qquad c = (1,0,0)^t. \tag{3}$$

The vertex $(-1,0,0)$ is the optimal solution of $\tilde{\Pi}$. (The proofs below work as long as $\bar{\Pi}$ is not empty and the corresponding face does not contain (0, 0, 0), but the notation for the proof of the general case would be horrendous.)

The main results of this paper are the following theorems.

Theorem 1. *Let Π be given by* (2). *There exists $s = (1, s_2, s_3) \in \mathcal{F}_\Pi^+$ and $0 < \mu < 1$ such that if $x^k = ts$ for $t > 0$ and $\lambda = 0.999$ then $x^{k+2} = \mu x^k$.*

Theorem 2. *Let $\tilde{\Pi}$ be given by* (3). *There exists $\varepsilon > 0$, a curve σ: $(0,\varepsilon) \to \mathcal{F}_{\tilde{\Pi}}^+$, and a function ϕ: $(0,\varepsilon) \to (0,\varepsilon)$ with $0 < \phi(t) < t/4$ such that if $\lambda = 0.999$ and $x^k = \sigma(t)$ then $x^{k+2} = \sigma(\phi(t))$.*

In other words, there exists a half line $\mathcal{L}$ in $\mathcal{F}_\Pi^+$ such that if the algorithm starts from $x^0 \in \mathcal{L}$ then all the even iterates will lie on $\mathcal{L}$ and converge to 0, which is not optimal. The behavior of the second example is a nonlinear version of the behavior of the first one, with the half line $\mathcal{L}$ replaced by the curve σ. The nonlinearity is introduced when we turn Π into the acceptable program $\tilde{\Pi}$ by adding one constraint. Theorem 2 holds because this nonlinearity has a negligible effect.

The values of s_2 and s_3 in Theorem 1 are close to $\bar{s}_2$ and $\bar{s}_3$ given, respectively, by

$$\begin{aligned} &2.373875277831879570815871749315245119314258024281969559631585343413865, \\ &0.105896780064483343718545069960983375540459740821398318758336306497571. \end{aligned} \tag{4}$$

These values were computed by Mathematica with high precision and rounded to 70 digits.

We leave the proofs of Theorems 1 and 2 to section 4 and explain now how we found Π and $\tilde{\Pi}$. For two variables, we proved that the iterates converge to an optimal

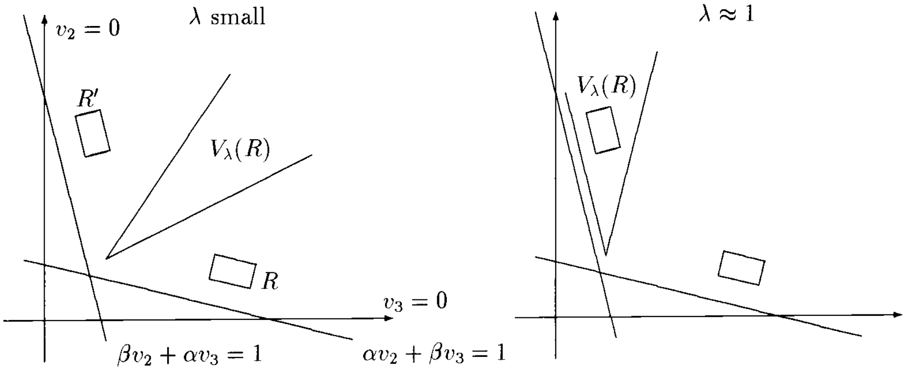


FIG. 1.

solution for every $\lambda \in (0,1)$ and feasible x^0 [M]. Therefore, we needed three variables. Using Farkas's lemma and scaling, we concluded that the minimal program for which the vertex 0 is degenerate and not optimal looks like $\bar{\Pi}$ given by

$$\bar{A} = \begin{pmatrix} 0 & 0 & -1 & -1 \\ 1 & 0 & \sin\theta_1 & \cos\theta_1 \\ 0 & 1 & \sin\theta_2 & \cos\theta_2 \end{pmatrix}, \qquad \bar{b} = (0,0,0,0)^t, \qquad \bar{c} = (1,0,0)^t, \tag{5}$$

where θ_1 and θ_2 are free parameters. To simplify, we took $\sin\theta_1 = \cos\theta_2 = \alpha$ and $\cos\theta_1 = \sin\theta_2 = \beta$ equal to the numerical value of $\sqrt{1-\alpha^2}$, getting program Π in (2). This choice makes Π symmetric with respect to the permutation $x_2 \leftrightarrow x_3$. We leave to the reader the verification that if $P \in \mathbb{R}^{3\times 3}$ is the permutation matrix such that

$$P(x_1, x_2, x_3)^t = (x_1, x_3, x_2)^t, \tag{6}$$

then the iterates for programs (2) and (3) satisfy

$$y^k = Px^k \Rightarrow y^{k+1} = Px^{k+1}. \tag{7}$$

Since Π is homogeneous, for $x_1 > 0$, we looked at

$$v^t(x) = (v_1(x), v_2(x), v_3(x))^t = \left(x_1, \frac{x_2}{x_1}, \frac{x_3}{x_1}\right)^t. \tag{8}$$

If $v_1(x) > 0$ then $x \in \mathcal{F}_\Pi^+$ if and only if $(v_2(x), v_3(x))$ belongs to the set

$$\mathcal{V}_\alpha = \{(v_2, v_3) \in \mathbb{R}^2 \text{ s.t. } v_2, v_3 > 0, \alpha v_2 + \beta v_3 > 1, \beta v_2 + \alpha v_3 > 1\}.$$

The evolution of $v_2(x)$ and $v_3(x)$ is independent from $v_1(x)$ and there exists a differentiable function $V_{\alpha,\lambda}$ with domain containing $\mathcal{V}_\alpha$ such that

$$(v_2(x^{k+1}), v_3(x^{k+1})) = V_{\alpha,\lambda}(v_2(x^k), v_3(x^k)).$$

As long as $v_1^k > 0$, we can recover x from v using $x^k = v_1^k(1, v_2^k, v_3^k)$. We used graphics routines to study $V_{\alpha,\lambda}$ and found that for $\alpha \approx 0.4$ and $V_\lambda = V_{\alpha,\lambda}$, the iterations are described by Figure 1.

In this figure, R is a region in the (v_2, v_3) plane and $V_\lambda(R)$ is its image by V_λ. We use a prime ($'$) to denote "symmetrical with respect to the diagonal $v_2 = v_3$." The intersection of R' and $V_\lambda(R)$ is empty if λ is small. However, as λ increases, $V_\lambda(R)$ moves to the left. When λ is about 0.999, $V_\lambda(R)$ contains R'. By the symmetry of $\Pi, V_\lambda(R') = V_\lambda(R)'$ and $R \subset V_\lambda(R') \subset V_\lambda(V_\lambda(R))$. Since $V_\lambda^{(2)}$ is continuous and one-to-one in $R, R \subset V_\lambda^{(2)}(R)$ and R is convex and bounded, there exists (s_2, s_3) in R such that $V_\lambda^{(2)}(s_2, s_3) = (s_2, s_3)$. (This is a version of Brower's fixed point theorem and can be proven by applying the standard argument, presented on p. 194 of [Sp], to the inverse of V_λ^2.) Thus, if $x^0 = (1, s_2, s_3)$ then $x^2 = (x^2)_1(1, s_2, s_3) = \mu x^0$. According to Mathematica, $\mu \approx 0.001$. Since Π is homogeneous, if $x^k = (x^k)_1(1, s_2, s_3)$, then $x^{k+2} = \mu(x^k)_1(1, s_2, s_3) = \mu x^k$. Geometrically, if $x^k \in \mathcal{L} = \{t(1, s_2, s_3), t > 0\}$, then x^{k+2} is also in $\mathcal{L}$. Moreover, x^{k+2} is closer to 0 than x^k and $\lim_{k\to\infty} x^k = 0$. To get an example with an optimal solution we add a constraint to bound $\mathcal{F}_\Pi$, getting $\tilde{\Pi}$ in (3). With this new restriction, the evolution of v_2 and v_3 depends on v_1. This dependence is weak if $v_1 \approx 0$ and the new example behaves like Π near 0.

3. Notation. Let Π be the linear program in (1). The *slack* ξ is defined as $\xi(\Pi, x) = A^t x - b$. If $v \in \mathbb{R}^k$, then (v) is the diagonal matrix with diagonal v and $\max(v)$ is the value of its biggest entry. The dual affine scaling algorithm steps in the direction

$$d(\Pi, x) = (A[\xi(\Pi, x)]^{-2}A^t)^{-1}c, \tag{9}$$

normalized by

$$\chi(\Pi, x) = \max([\xi(\Pi, x)]^{-1}A^t d(\Pi, x)). \tag{10}$$

The next iterate of the dual affine scaling algorithm is given by

$$x^{k+1} = N(\Pi, \lambda, x^k) = x^k - \frac{\lambda}{\chi(\Pi, x^k)}\, d(\Pi, x^k). \tag{11}$$

The 2-norm of the matrix M is $\|M\|$. If F is a function, then $F^{(k)}$ is its kth iterate; that is, $F^{(0)}(x) = x$ and $F^{(k+1)}(x) = F(F^{(k)}(x))$. Analogously, we define $N^{(0)}(\Pi, \lambda, x) = x$ and $N^{(k+1)}(\Pi, \lambda, x) = N(\Pi, \lambda, N^{(k)}(\Pi, \lambda, x))$. We use $C^k(D, S)$ to denote the set of k times continuously differentiable functions from D to S. The jacobian of F at x is $JF(x)$. The open ball of center x and radius ρ in $\mathbb{R}^k$ is

$$B^k(x, \rho) = \{y \in \mathbb{R}^k \text{ s.t. } \|x - y\| < \rho\}.$$

4. Proofs of Theorems 1 and 2. In this section we prove Theorems 1 and 2. Motivated by the graphical arguments in section 2, we express the slacks of problem $\tilde{\Pi}$ in terms of the normalized variable v, defined in (8), and get

$$\xi^t(x) = x_1\left(v_2, v_3, \alpha v_2 + \beta v_3 - 1, \beta v_2 + \alpha v_3 - 1, \frac{1 + v_1 - v_1 v_2 - v_1 v_3}{v_1}\right)^t. \tag{12}$$

For the program Π in (2) the slacks are given by the first four components of this vector. This motivates the introduction of the functions

$$\psi^t(v) = \left(\frac{1}{v_2}, \frac{1}{v_3}, \frac{1}{\alpha v_2 + \beta v_3 - 1}, \frac{1}{\beta v_2 + \alpha v_3 - 1}, \frac{v_1}{1 + v_1 - v_1 v_2 - v_1 v_3} \right)^t, \tag{13}$$

$$\omega(v) = (\tilde{A}[\psi(v)]^2 \tilde{A}^t)^{-1}, \tag{14}$$

$$\tau(v) = [\psi(v)]\tilde{A}^t \omega(v) c, \tag{15}$$

$$\kappa(v) = \max(\tau(v)), \tag{16}$$

$$\delta(v) = \kappa(v) v - \lambda \omega(v) c, \tag{17}$$

$$\theta(v) = \kappa(v) - \lambda(\omega(v)c)_1. \tag{18}$$

Notice that the functions above are well defined for all $v \in \mathbb{R}^3$ for which $\psi(v)$ and $\omega(v)$ are defined, even if $v_1 = 0$. The following lemma states facts about the functions in (16)–(18) which hold even for $v_1 \le 0$, and we will be careful and require that $v_1 > 0$ when we want to conclude something about x. Let

$$\bar{s} = (0, \bar{s}_2, \bar{s}_3) \tag{19}$$

with $\bar{s}_2$ and $\bar{s}_3$ given in (4). We have the following lemma.

Lemma 1. *The functions κ, δ, and θ are rational in $B^3(\bar{s}, 10^{-50})$ and*

$$10^{-2} < \kappa(v), \|\delta(v)\|_\infty, \theta(v) < 1, \tag{20}$$

$$\left| \frac{\partial \kappa}{\partial v_i}(v) \right|, \left\| \frac{\partial \delta}{\partial v_i}(v) \right\|, \left| \frac{\partial \theta}{\partial v_i}(v) \right| < 10^{20}, \tag{21}$$

$$\left\| \frac{\partial^2 \delta}{\partial v_i \partial v_j}(v) \right\|, \left| \frac{\partial^2 \theta}{\partial v_i \partial v_j}(v) \right| < 10^{30}, \tag{22}$$

$$\frac{\partial \theta}{\partial v_1}(0, v_2, v_3) = \frac{\partial \delta_2}{\partial v_1}(0, v_2, v_3) = \frac{\partial \delta_3}{\partial v_1}(0, v_2, v_3) = 0. \tag{23}$$

The proofs of Theorems 1 and 2 use the functions above to describe the evolution of the iterates. For $\tilde{\Pi}$, we get from (9), (10), and (13)–(16) that $d(x) = (x_1)^2 \omega(v(x))c, \chi(x) = x_1 \kappa(v(x))$, and

$$x^{k+1} = x^k - x_1^k \frac{\lambda}{\kappa(v(x^k))}\, \omega(v(x^k))c. \tag{24}$$

It follows from (17)–(18) and (24) that if $i = 2, 3, x_1^k > 0$ and $x_1^{k+1} > 0$, then

$$v_1(x^{k+1}) = x_1^{k+1} = v_1(x^k) \frac{\theta(v(x^k))}{\kappa(v(x^k))},$$

$$v_i(x^{k+1}) = \frac{x_i^{k+1}}{x_1^{k+1}} = \frac{\delta_i(v(x^k))}{\theta(v(x^k))}.$$

In other words, $v(x^{k+1}) = G(v(x^k))$ for

$$G(v) = \left(v_1 \frac{\theta(v)}{\kappa(v)}, \frac{\delta_2(v)}{\theta(v)}, \frac{\delta_3(v)}{\theta(v)} \right). \tag{25}$$

Notice that G is a function of v and may be defined even when $v_1 = 0$.

The evolution of the x^k in program Π is similar; we only need to notice that Π is homogeneous and replace v_1 by 0 in the fifth component of ψ in (13). The reader can verify that in this case

$$v(x^{k+1}) = \left(v_1^k \frac{\theta(\bar{v})}{\kappa(\bar{v})}, G_2(\bar{v}), G_3(\bar{v}) \right) \tag{26}$$

for $\bar{v} = (0, v_2(x^k), v_3(x^k))$.

Now consider the permutation P in (6) and define

$$H(v) = PG(v) = (G_1(v), G_3(v), G_2(v)). \tag{27}$$

It follows from Lemma 1 (especially (23)) that if $v \in B^3(\bar{s}, 10^{-50})$ and $v_1 = 0$ then

$$JH(v) = \begin{pmatrix} \frac{\theta(v)}{\kappa(v)} & 0 & 0 \\ 0 & \frac{\partial G_3}{\partial v_2}(v) & \frac{\partial G_3}{\partial v_3}(v) \\ 0 & \frac{\partial G_2}{\partial v_3}(v) & \frac{\partial G_2}{\partial v_3}(v) \end{pmatrix}.$$

The last result we need to prove Theorems 1 and 2 is the following lemma.

Lemma 2. *There exists $s = (0, s_2, s_3)$ such that $s_2, s_3 > 0, \alpha s_2 + \beta s_3 > 1, \beta s_2 + \alpha s_3 > 1, H$ in (27) belongs to $C^1(B^3(s, 10^{-60}), \mathbb{R}^3), H(s) = s, |\theta(s)/\kappa(s) - 0.03| < 10^{-2}$, and the eigenvalues of the right lower corner of $JH(s)$ satisfy $\|(\mu_2, \mu_3) - (-1.1, 15.06)\| < 10^{-2}$.*

We now prove Theorems 1 and 2 and finish this section.

Proof of Theorem 1. If s is as in Lemma 2 then $G_2(s) = H_3(s) = s_3$ and $G_3(s) = H_2(s) = s_2$. Therefore, if $x^k = t(1, s_2, s_3)$ then $v^k = (t, s_2, s_3)$ and (26) imply that

$$v^{k+1} = v(x^{k+1}) = \left(t\, \frac{\theta(s)}{\kappa(s)}, s_3, s_2 \right).$$

Taking $s' = (0, s_3, s_2)$, we get by the symmetry of Π that $\kappa(s') = \kappa(s)$ and $\theta(s') = \theta(s)$. Notice that $v_1^{k+1} > 0$. Therefore, (26) and the symmetry of Π lead to

$$v^{k+2} = v(x^{k+2}) = \left(t \left(\frac{\theta(s)}{\kappa(s)} \right)^2, G_2(s'), G_3(s') \right) = (\mu t, s_2, s_3)$$

for $\mu = \theta(s)^2/\kappa(s)^2$. Since $\mu t > 0$, we conclude that $x^{k+2} = \mu t(1, s_2, s_3) = \mu x^k$. Lemma 2 shows that $0 < \mu < 1$ and the proof of Theorem 1 is complete. □

Proof of Theorem 2. We will apply the *stable manifold theorem* [HP] to H and s from Lemma 2. Since this theorem is not widely known in the mathematical programming community and in order to make the paper self contained, we state only a weak version of it, which we will prove in section 6.

The Stable Manifold Theorem. *Let $a > 0$ and F in $C^1((-a, a) \times B^n(0, a), \mathbb{R}^{1+n})$ be given by*

$$F(x, y) = (\mu_0 x + R(x, y), Ay + S(x, y))$$

with $R(0, 0) = 0, S(0, 0) = 0, \nabla R(0, 0) = 0$, and $JS(0, 0) = 0$. Assume that $|\mu_0| < 1$ and the eigenvalues μ_i of A satisfy $|\mu_i| > 1$ for all i. Then, given $\varepsilon > 0$, there exits $\delta > 0, \mu$ in $C^0((-\delta, \delta), \mathbb{R})$, and γ in $C^0((-\delta, \delta), \mathbb{R}^n)$ such that

$$F(t, \gamma(t)) = (\mu(t), \gamma(\mu(t))) \tag{28}$$

and

$$|\mu(t) - \mu_0 t| + \|\gamma(t)\| \leq \varepsilon |t|. \tag{29}$$

The curve $\mathcal{C}(t) = (t, \gamma(t))$ is called *the stable manifold* of F at (0, 0) because if $x \in \mathcal{C}$ then $F^{(k)}(x)$ converges to (0, 0) as $k \to \infty$. Equation (28) means that if $x \in \mathcal{C}$ then $F(x)$ is also in $\mathcal{C}$, and (29) means that $(t, \gamma(t))$ is tangent to $(t, 0)$. If F were linear, then θ would be the straight line $(t, 0)$ and $\mu(t)$ would equal $\mu_0 t$. The stable manifold theorem says that near (0, 0) F behaves as its linearization.

Lemma 2 and the stable manifold theorem applied to $F(v) = H(v + s) - s$ with

$$\varepsilon = \frac{1}{2} \min \left\{ \mu_0, \frac{1}{2} - \mu_0, s_2, s_3, \alpha s_2 + \beta s_3 - 1, \beta s_2 + \alpha s_3 - 1 \right\} > 0 \tag{30}$$

and $0 < \mu_0 = \theta(s)/\kappa(s) < 1/2$ imply that there exist $\delta \in (0, 10^{-60})$, $\mu \in C^0((-\delta, \delta), \mathbb{R})$, and $\gamma \in C^0((-\delta, \delta), \mathbb{R}^2)$, with

$$H(t, s_2 + \gamma_2(t), s_3 + \gamma_3(t)) = (\mu(t), s_2 + \gamma_2(\mu(t)), s_3 + \gamma_3(\mu(t))), \tag{31}$$

where we have written $\gamma(t) = (\gamma_2(t), \gamma_3(t))$ and such that (29) holds.

To complete the proof we show that

$$\sigma(t) = (t, s_2 t + \gamma_2(t) t, s_3 t + \gamma_3(t) t)$$

and $\phi(t) = \mu(\mu(t))$ satisfy

$$0 < t < \delta \Rightarrow 0 < \mu(t) < t/4, \tag{32}$$

$$0 < t < \delta \Rightarrow \sigma(t) \in \mathcal{F}_{\tilde{\Pi}}^{+}, \tag{33}$$

$$x^k = \sigma(t) \Rightarrow x^{k+2} = \sigma(\phi(t)). \tag{34}$$

Let us start with (32). Inequality (29) implies that if $0 < t < \delta$ then $(\mu_0 - \varepsilon)t < \mu(t) < (\mu_0 + \varepsilon)t < t$. Equation (30) shows that $\mu_0 - \varepsilon > 0$ and $\mu_0 + \varepsilon < 1/2$. Therefore, $0 < \mu(t) < t < \delta$ and, replacing t by $\mu(t)$ in the argument above, we get $0 < \mu(\mu(t)) < (\mu_0 + \varepsilon)\mu(t) < (\mu_0 + \varepsilon)^2 t \leq t/4$.

If $0 < t < \delta$ then $\sigma_1(t) > 0$ and, as in (8), we can define

$$\nu(t) = v(\sigma(t)) = (t, s_2 + \gamma_2(t), s_2 + \gamma_3(t)),$$

and it follows from (30) and (29) that the slacks in (12) are all positive and (33) holds.

We now use the symmetry of $\mathcal{F}_{\tilde{\Pi}}$ and (7) to show (34). By the definition of G and (25) we get $v(x^{k+1}) = G(v(x^k))$. It follows from (31) that $H(\nu(t)) = \nu(\mu(t))$ and

$$v(N(\sigma(t))) = G(v(\sigma(t))) = G(\nu(t)) = PH(\nu(t)) = P\nu(\mu(t)).$$

Since $\nu_1(\mu(t)) = \mu(t) > 0$ and $v(Px) = Pv(x)$, it follows from the last equation that $N(\sigma(t)) = P\sigma(\mu(t))$. Therefore, from (7) and $PP = I$, we get

$$N^{(2)}(\sigma(t)) = N(N(\sigma(t))) = N(P\sigma(\mu(t))) = PN(\sigma(\mu(t))) = PP\sigma(\mu^{(2)}(t)) = \sigma(\phi(t)),$$

as required by Theorem 2. □

5. Technical lemmas. In this section we will prove Lemmas 1 and 2 from the last section. We used Mathematica to evaluate rational functions in these proofs. By "Mathematica shows that $x < y$," we mean that the symbolically computed x and y warrant such a conclusion.

Proof of Lemma 2. We will show that Lemma 2 is satisfied by s close to $\bar{s}$ in (19). Mathematica shows that $\|H(\bar{s})-\bar{s}\| < 10^{-66}$, which is strong evidence of the existence of s close to $\bar{s}$ such that $H(s) = s$. It also shows that $JH(\bar{s})$ satisfies the conditions required from $JH(s)$ in the thesis of Lemma 2. This is evidence that $JH(s)$ is fine. This proof contains estimates showing that the evidence is correct.

Using Lemma 1 and (25), for $i = 2, 3$, we get

$$\frac{\partial^2 G_i}{\partial v_j \partial v_k} = \frac{1}{\theta}\frac{\partial^2 \delta_i}{\partial v_j \partial v_k} - \frac{1}{\theta^2}\left(\frac{\partial \delta_i}{\partial v_j}\frac{\partial \theta}{\partial v_k} + \frac{\partial \delta_i}{\partial v_k}\frac{\partial \theta}{\partial v_j} + \delta_i \frac{\partial^2 \theta}{\partial v_i \partial v_j}\right) + 2\,\frac{\delta_i}{\theta^3}\frac{\partial \theta}{\partial v_j}\frac{\partial \theta}{\partial v_k}$$

and

$$\left|\frac{\partial^2 G_i}{\partial v_j \partial v_k}(v)\right| < 10^{32} + 10^{44} + 10^{44} + 10^{34} + 10^{46} < 10^{47}. \tag{35}$$

Equation (25) shows that $H(0, s_2, s_3) = (0, s_2, s_3)$ if and only if

$$F(s_2, s_3) = (G_3(0, s_2, s_3) - s_2, G_2(0, s_2, s_3) - s_3) = (0, 0).$$

The jacobian of F is

$$\begin{pmatrix} \frac{\partial G_3}{\partial v_2} - 1 & \frac{\partial G_3}{\partial v_3} \\ \frac{\partial G_2}{\partial v_2} & \frac{\partial G_2}{\partial v_3} - 1 \end{pmatrix}.$$

It follows from (35) that, for all $v, v' \in B^3(\bar{s}, 10^{-50})$, we have

$$\|JF(v_2, v_3) - JF(v_2', v_3')\| \leq 10^{49}\|v - v'\|. \tag{36}$$

According to Lemma 1, the functions κ, δ, and θ and their derivatives are rational. We evaluated them symbolically at $\bar{s}$ and found that

$$\left\|\frac{\partial G}{\partial v_i}(\bar{s})\right\| < 10^3, \tag{37}$$

$$\|JF^{-1}(\bar{s}_2, \bar{s}_3)\| < 10, \tag{38}$$

$$\|JF^{-1}(\bar{s}_2, \bar{s}_3)F(\bar{s}_2, \bar{s}_3)\| < 10^{-65}. \tag{39}$$

We now apply Kantorovich's theorem, as stated on p. 92 of [DS]. In [DS]'s notation, (36), (38), and (39) correspond to $\beta = 10, \gamma = 10^{49}, \eta = 10^{-65}$, and $\alpha = 10^{-15}$. It follows from Kantorovich's theorem that F has a zero (s_2, s_3) in $B^2((\bar{s}_2, \bar{s}_3), r_0)$ with $r_0 = (1 - \sqrt{1 - 2 \times 10^{-15}})/10^{50} < 10^{-60}$.

Since $\|\bar{s} - s\| < 10^{-60}, B^3(s, 10^{-60}) \subset B^3(\bar{s}, 10^{-50})$ and Lemma 1 show that κ, δ, and θ are positive rational functions in $B^3(s, 10^{-60})$. Equation (25) shows that $H = PG$ is in $C^1(B^3(\bar{s}, 10^{-60}), \mathbb{R}^3)$. Since $s_1 = 0$, it follows from (23) that $\frac{\partial G_2}{\partial v_1}(s) = \frac{\partial G_3}{\partial v_1}(s) = 0$ and, for $i = 2, 3$, $\frac{\partial G_1}{\partial v_i}(s) = s_1\big(\partial \frac{\theta}{\kappa}/\partial v_i\big)(s) = 0$ and $\frac{\partial G_1}{\partial v_1}(s) = \theta(s)/\kappa(s)$.

Mathematica shows that $|\theta(\bar{s})/\kappa(\bar{s}) - 0.033| < 10^{-3}$. Lemma 1 and $\|s - \bar{s}\| < 10^{-50}$ imply that $|\theta(s)/\kappa(s) - 0.03| < 10^{-2}$. Let $p(\mu, v)$ be the characteristic polynomial of the right lower corner of $JH(v)$:

$$p(\mu, v) = \left(\frac{\partial G_3}{\partial v_2}(v) - \mu\right)\left(\frac{\partial G_2}{\partial v_3}(v) - \mu\right) - \frac{\partial G_2}{\partial v_3}(v)\frac{\partial G_3}{\partial v_3}(v).$$

Bounds (37) and (35) show that $\left\|\frac{\partial G}{\partial v_i}(v)\right\| < 10^4$ in $B(\bar{s}, 10^{-50})$. It follows that if $|\mu| < 10^2$ then $|p(\mu, s) - p(\mu, \bar{s})| < 10^{-5}$. Mathematica shows that $p(-1.11, \bar{s}) > 0.1, p(-1.09, \bar{s}) < -0.1, p(15.05, \bar{s}) < -0.1$, and $p(15.07, \bar{s}) > 0.1$. Thus, $p(-1.11, s)$ $p(-1.09, s)$ and $p(15.05, s)p(15.07, s)$ are negative. It follows that the right lower corner of $JH(s)$ has eigenvalues μ_2, μ_3 such that $\|(\mu_2, \mu_3) - (-1.1, 15.06)\|_\infty < 10^{-2}$, and the proof of Lemma 2 is complete. □

Proof of Lemma 1. Let us bound ψ (see (13)) and its derivatives. Notice that $\alpha\bar{s}_2 + \beta\bar{s}_3 - 1 > 0.035$. Since $0 < \alpha, \beta < 1$, and $v \in B^3(\bar{s}, 10^{-50})$, we have

$$\alpha v_2 + \beta v_3 - 1 > 0.035 - 2 \times 10^{-50} > 10^{-2}.$$

Therefore, $|\psi(v)_3| < 10^2$. Similar computations show that $\|\psi(v)\|_\infty \leq 10^2$ and

$$\|[\psi(v)]\| \leq 10^2. \tag{40}$$

Differentiating (13) with respect to v, we get

$$\frac{\partial \psi^t}{\partial v_1}(v) = \left(0, 0, 0, 0, \frac{1}{(1 - v_1 v_2 - v_1 v_3 + v_1)^2}\right)^t = (0, 0, 0, 0, \psi_5^2)^t, \tag{41}$$

$$\frac{\partial \psi}{\partial v_2}(v) = (-\psi_1^2, 0, -\alpha\psi_3^2, -\beta\psi_4^2, v_1\psi_5^2)^t. \tag{42}$$

Since $0 < |v_1|, \alpha, \beta < 1$, equations (41) and (42) and the analogous equation to v_3 show that

$$\left\|\left[\frac{\partial \psi}{\partial v_i}(v)\right]\right\| \leq 10^4. \tag{43}$$

Since $J\psi$ is 5×3, we have

$$\|J\psi(v)\| \leq \sqrt{15} \times 10^4 < 10^5. \tag{44}$$

Using (41) and (42) and noticing that $|v_1| < 10^{-50}$ and $0 < \alpha, \beta < 1$, we get

$$\left|\frac{\partial^2 \psi_k}{\partial v_i \partial v_j}(v)\right| \leq \left|2\psi_k \frac{\partial \psi_k}{\partial v_i}\right| + \psi_5^2 < 2 \times 10^2 \times 10^4 + 10^4 < 10^7. \tag{45}$$

Let us now bound ω and its derivatives. From (14) we get

$$\omega(v) = (\tilde{A}[\psi(\bar{s})]^2\tilde{A}^t - \tilde{A}([\psi(\bar{s})]^2 - [\psi(v)]^2)\tilde{A}^t)^{-1} = (I - C(v))^{-1}\omega(\bar{s}), \tag{46}$$

where $C(v) = \omega(\bar{s})\tilde{A}([\psi(\bar{s}) - \psi(v)])([\psi(\bar{s}) + \psi(v)])\tilde{A}^t$. Notice that $\|\tilde{A}\| < 10$ and (44) implies that $\|[\psi(\bar{s}) - \psi(v)]\| < 10^5 \times \|v - \bar{s}\| < 10^{-45}$. Mathematica shows that

$\|\omega(\bar{s})\| < 10$, and (4) implies that $\|C(v)\| < 10 \times 10 \times 10^{-45} \times (2 \times 10^2) \times 10 < 10^{-10}$. It follows from (46) that

$$\|\omega(v)\| \leq \frac{\|\omega(\bar{s})\|}{1 - \|C(v)\|} < 10^2. \tag{47}$$

Differentiating ω, we get

$$\frac{\partial \omega}{\partial v_i}(v) = -2\omega(v)\tilde{A}\left[\frac{\partial \psi}{\partial v_i}(v)\right][\psi]\tilde{A}^t\omega(v) \tag{48}$$

and

$$\begin{aligned}\frac{\partial^2 \omega}{\partial v_j \partial v_i}(v) = &- 2\,\frac{\partial \omega}{\partial v_j}(v)\tilde{A}\left[\frac{\partial \psi}{\partial v_i}(v)\right][\psi]\tilde{A}^t\omega(v) - 2\omega(v)\tilde{A}\left[\frac{\partial^2 \psi}{\partial v_i \partial v_j}(v)\right][\psi]\tilde{A}^t\omega(v)\\ &- 2\omega(v)\tilde{A}\left[\frac{\partial \psi}{\partial v_i}(v)\right]\left[\frac{\partial \psi}{\partial v_j}(v)\right]\tilde{A}^t\omega(v) - 2\omega(v)\tilde{A}\left[\frac{\partial \psi}{\partial v_i}(v)\right][\psi]\tilde{A}^t\frac{\partial \omega}{\partial v_j}(v).\end{aligned}$$

Using (40)–(49), we get

$$\left\|\frac{\partial \omega}{\partial v_i}(v)\right\| < 2 \times 10^2 \times 10 \times 10^4 \times 10^2 \times 10 \times 10^2 < 10^{13}, \tag{49}$$

$$\left\|\frac{\partial^2 \omega}{\partial v_i \partial v_j}(v)\right\| < 2 \times (10^{23} + 10^{15} + 10^{14} + 10^{23}) < 10^{24}. \tag{50}$$

Differentiating (15), we obtain

$$\frac{\partial \tau}{\partial v_i}(v) = \left[\frac{\partial \psi}{\partial v_i}(v)\right]\tilde{A}^t\omega(v)c + [\psi(v)]\tilde{A}^t\frac{\partial \omega}{\partial v_i}(v)c \tag{51}$$

and

$$\begin{aligned}\frac{\partial^2 \tau}{\partial v_j \partial v_i}(v) = &\left[\frac{\partial^2 \psi}{\partial v_i \partial v_j}(v)\right]\tilde{A}^t\omega(v)c + \left[\frac{\partial \psi}{\partial v_i}(v)\right]\tilde{A}^t\frac{\partial \omega}{\partial v_j}(v)c\\ &+ \left[\frac{\partial \psi}{\partial v_j}(v)\right]\tilde{A}^t\frac{\partial \omega}{\partial v_i}(v)c + [\psi(v)]\tilde{A}^t\frac{\partial^2 \omega}{\partial v_j \partial v_i}(v)c.\end{aligned}$$

Since $\|c\| = 1$, the estimates above show that $\left|\frac{\partial \tau_k}{\partial v_i}\right| < 2\ times\ 10^{16}$. The jacobian of τ is 5×3 and

$$\|J\tau(v)\| < \sqrt{15} \times 2 \times 10^{16} < 10^{17}. \tag{52}$$

We also have

$$\left\|\frac{\partial^2 \tau}{\partial v_j \partial v_i}(v)\right\| \leq 10^{10} + 10^{18} + 10^{18} + 10^{27} < 10^{28}. \tag{53}$$

Mathematica shows that $\tau_4(\bar{s}) > \tau_i(\bar{s}) + 10^{-4}$ for $i \neq 4$, and (52) implies that

$$\begin{aligned}\tau_4(v) &> \tau_4(\bar{s}) - 10^{17} \times 10^{-50} > \tau_i(\bar{s}) + 10^{-4} - 10^{-33}\\ &> \tau_i(v) + 10^{-5} - 10^{-33} > \tau_i(v) + 10^{-6}.\end{aligned}$$

Therefore, for all $v \in B^3(\bar{s}, 10^{-50}), \kappa(v) = \tau_4(v)$, which is a rational function of v. This shows that δ (see (18)) is a rational function in $B^3(\bar{s}, 10^{-50})$ and

$$\frac{\partial \delta}{\partial v_i}(v) = \frac{\partial \tau_4}{\partial v_i}(v)v + \tau_4(v)\frac{\partial v}{\partial v_i}(v) - \lambda \frac{\partial \omega}{\partial v_i}(v)c,$$

$$\frac{\partial^2 \delta}{\partial v_i \partial v_j}(v) = \frac{\partial^2 \tau_4}{\partial v_j \partial v_i}(v)v + \frac{\partial \tau_4}{\partial v_i}(v)\frac{\partial v}{\partial v_j} + \frac{\partial \tau_4}{\partial v_j}(v)\frac{\partial v}{\partial v_i} - \lambda \frac{\partial^2 \omega}{\partial v_j \partial v_i}(v)c$$

because $\frac{\partial^2 v}{\partial v_i \partial v_j} = 0$. The derivatives of θ are similar, and (21) and (22) follow from the estimates above. Mathematica shows that $|\theta(\bar{s}) - 0.015| < 10^{-3}, |\kappa(\bar{s}) - 0.4| < 10^{-1}$, and $\|\delta(\bar{s})\|_\infty < 0.5$. Bound (20) follows from (21), and $\|v - \bar{s}\| < 10^{-50}$.

To prove (23), notice that if $v_1 = 0$ then (41) implies that $\frac{\partial [\psi]^2}{\partial v_1}(v) = 0$. Therefore, (48) implies that $\frac{\partial \omega}{\partial v_1}(v) = 0$, and (51) shows that $\frac{\partial \tau}{\partial v_1}(v) = 0$. Since $\kappa(v) = \tau_4(v)$, (18) shows that $\frac{\partial \theta}{\partial v_1}(v) = 0$. Since $\frac{\partial v_i}{\partial v_1} = 0$ if $i = 2, 3$, (18) implies (23), and the proof of Lemma 1 is complete. □

6. The stable manifold theorem. In this section we prove our weak version of the stable manifold theorem. Our proof is an adaptation of the traditional one, presented in [HP]. The idea is to characterize γ as the fixed point of a contraction on a complete metric space and take $\mu(t) = \mu_\gamma(t)$, where

$$\mu_\phi(t) = \mu_0 t + R(t, \phi(t)). \tag{54}$$

Since all the eigenvalues of A have absolute value bigger than 1, there exists a norm $\|\cdot\|_A$ in $\mathbb{R}^n$ such that the subordinated operator norm $\|\cdot\|_A$ satisfies $\|A^{-1}\|_A < 1$. Moreover, there exists K such that $\|y\| \leq K\|y\|_A$ for all $y \in \mathbb{R}^n$. Since $\|A^{-1}\|_A$ and $|\mu_0|$ are strictly less than 1, there exists $\theta > 0$ such that

$$\begin{aligned} &\theta \leq \frac{a}{K}, \qquad |\mu_0| + \theta^2 + \theta^3 < 1, \qquad (K + \theta + \theta^2)\theta < \varepsilon, \\ &\text{and} \qquad \|A^{-1}\|_A(1 + \theta + \theta^2 + \theta^3) < 1. \end{aligned} \tag{55}$$

Since R and S are C^1 and their derivatives vanish at $(0, 0)$, there exists $\delta \in (0, 1)$, with $\delta < a$, such that if $|x_1|, |x_2|, \|y_1\|_A, \|y_2\|_A \leq \delta$ then

$$|R(x_1, y_1) - R(x_2, y_2)| < \theta^2(|x_1 - x_2| + \|y_1 - y_2\|_A) \tag{56}$$

and

$$\|S(x_1, y_1) - S(x_2, y_2)\|_A < \theta^2(|x_1 - x_2| + \|y_1 - y_2\|_A). \tag{57}$$

Consider the complete metric space

$$\mathcal{H} = \{\phi \in C^0((-\delta, \delta), \mathbb{R}^n) \text{ s.t. } \phi(0) = 0, \ \|\phi(t_1) - \phi(t_2)\|_A \leq \theta|t_1 - t_2|\} \tag{58}$$

with the metric $d(\phi, \psi) = \sup_{-\delta \leq t \leq \delta} \|\phi(t) - \psi(t)\|_A$. We will show that the operator $T : \mathcal{H} \to \mathcal{H}$ given by

$$T[\phi](t) = A^{-1}(\phi(\mu_\phi(t)) - S(t, \phi(t))), \tag{59}$$

with μ_ϕ as in (54), is a contraction in $\mathcal{H}$. The proof proceeds in four steps. First we show that, given $\phi \in \mathcal{H}$, the function $T[\phi]$ is well defined; i.e., $(t, \phi(t))$ is in the domain of R and S and $\mu_\phi(t)$ is in the domain of ϕ. Next we show that $T[\phi] \in \mathcal{H}$. Then we show that T is a contraction. Finally we show that the fixed point γ of T and $\mu = \mu_\gamma$ are as required by the stable manifold theorem.

We now demonstrate that $T[\phi](x)$ is well defined for every $\phi \in \mathcal{H}$ and $x \in [-\delta, \delta]$. Since $\|\phi(x)\| \le K\|\phi(x)\|_A < K\theta\delta < a, S(t, \phi(x))$ and $R(t, \phi(x))$ are well defined. If $|y| \le \delta$ then (56) and (58) show that

$$|R(x,\phi(x)) - R(y,\phi(y))| \le \theta^2(|x-y| + \|\phi(x)-\phi(y)\|_A) \le (\theta^2+\theta^3)|x-y|. \tag{60}$$

(Analogously, $\|S(x,\phi(x)) - S(y,\phi(y))\| \le (\theta^2+\theta^3)|x-y|$.) Using (54) and (55), we get

$$\begin{aligned}|\mu_\phi(x) - \mu_\phi(y)| &\le |\mu_0(x-y) + R(x,\phi(x)) - R(y,\phi(y))| \\ &\le (|\mu_0| + \theta^2 + \theta^3)|x-y| < |x-y|.\end{aligned} \tag{61}$$

In particular, if $y = 0, |\mu_\phi(x)| < |x| < \delta$ and $\mu_\phi(x)$ is in the domain of ϕ.

Let us now demonstrate that $\psi = T[\phi] \in \mathcal{H}$. Since $\phi \in \mathcal{H}$ in (58), it is clear from (59) that $\psi(0) = 0$. Moreover,

$$\begin{aligned}\|\psi(x) - \psi(y)\|_A &\le \|A^{-1}\|_A(\|\phi(\mu_\phi(x)) - \phi(\mu_\phi(y))\| + \|S(x,\phi(x)) - S(y,\phi(y))\|_A) \\ &\le \|A^{-1}\|_A\theta|\mu_\phi(x)-\mu_\phi(y)| + \|A^{-1}\|_A(\theta^2+\theta^3)|x-y| \\ &\le \|A^{-1}\|_A(1+\theta+\theta^2)\theta|x-y|\end{aligned}$$

because of (61), and (55) shows that $\|\psi(x)-\psi(y)\|_A \le \theta|x-y|$. Therefore, $T[\phi] \in \mathcal{H}$.

Let us now show that T is a contraction. Equations (54) and (56) demonstrate that

$$|\mu_\phi(t) - \mu_\psi(t)| = |R(t,\phi(t)) - R(t,\psi(t))| \le \theta^2\|\phi(t)-\psi(t)\|_A \le \theta^2 d(\phi,\psi)$$

and

$$\|S(t,\phi(t)) - S(t,\psi(t))\|_A \le \theta^2\|\phi(t)-\psi(t)\|_A \le \theta^2 d(\phi,\psi).$$

Therefore,

$$\begin{aligned}\|T[\phi](t) - T[\psi](t)\|_A &\le \|A^{-1}\|_A(\|\phi(\mu_\phi(t)) - \psi(\mu_\psi(t))\|_A + \|S(t,\phi(t)) - S(t,\psi(t))\|_A) \\ &\le \|A^{-1}\|_A(\|\phi(\mu_\phi(t)) - \phi(\mu_\psi(t))\|_A + \|\phi(\mu_\psi(t)) - \psi(\mu_\psi(t))\|_A + \theta^2 d(\phi,\psi)) \\ &\le \|A^{-1}\|_A(\theta|\mu_\phi(t)-\mu_\psi(t)| + d(\phi,\psi) + \theta^2 d(\phi,\psi)) \le \|A^{-1}\|_A(1+\theta^2+\theta^3)d(\phi,\psi),\end{aligned}$$

and (55) shows that T is a contraction.

Finally, let us demonstrate that the fixed point γ of T and $\mu = \mu\gamma$ are as required by the stable manifold theorem. Notice that since $\gamma \in \mathcal{H}, \|\gamma(t)\|_A \le \theta|t|$ and

$$\begin{aligned}\|\gamma(t)\| + |\mu_\gamma(t) - \mu_0 t| &\le K\|\gamma(t)\|_A + |R(t,\gamma(t))| \\ &\le (K+\theta^2)\|\gamma(t)\|_A + \theta^2|t| \le (K+\theta+\theta^2)\theta|t| \le \varepsilon|t|.\end{aligned}$$

Since γ is a fixed point of T,

$$A\gamma(t) = \gamma(\mu_\gamma(t)) - S(t, \gamma(t))$$

and

$$F(t, \gamma(t)) = (\mu_\gamma(t), A\gamma(t) + S(t, \gamma(t))) = (\mu_{\gamma(t)}, \gamma(\mu_\gamma(t))),$$

and the proof is complete. □

REFERENCES

[DS] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1983.

[Dk] I. I. Dikin, *Iterative solutions of problems of linear and quadratic programming*, Soviet Math. Dokl., 8 (1967), pp. 674–675.

[Dk1] I. I. Dikin, *On the convergence of an iterative process*, Upravlyaemye Sistemy, 12 (1974), pp. 54–60. (In Russian.) (Suggested by the referee.)

[Dk2] I. I. Dikin, *The Convergence of Dual Variables*, Tech. report, Siberian Energy Institute, Russia, 1991.

[HP] M. Hirsch and C. Pugh, *Stable manifolds and hyperbolic sets*, in Global Analysis, Proc. Symp. in Pure Math. Vol. XIV, Berkeley, CA, July 1968, Amer. Math. Soc., 1970.

[M] W. Mascarenhas, *New Proofs of Convergence for the Dual Affine Scaling Algorithm*, RP 69/93, Universidade Estadual de Campinas, Campinas S.P., Brazil, 1993.

[MTW] R. D. C. Monteiro, T. Tsuchiya, and Y. Wang, *A simplified global convergence proof of the affine scaling algorithm*, Ann. Oper. Res., 47 (1993), pp. 443–482.

[S] R. Saigal, *A simple proof of the primal affine scaling method*, Ann. Oper. Res., to appear.

[Sp] E. Spanier, *Algebraic Topology*, McGraw–Hill, New York, 1971.

[TM] T. Tsuchiya and M. Muramatsu, *Global convergence of a long-step affine scaling algorithm for degenerate linear programming problems*, SIAM J. Optim., 5 (1995), pp. 525–551.

[W] S. Wolfram, *Mathematica: A System for Doing Mathematics by Computer*, 2nd ed., Addison–Wesley, Redwood City, CA, 1991.

SIAM J. Optim.
Vol. 7, No. 1, pp. 47–65, February 1997

ON THE CONVERGENCE OF THE MIZUNO–TODD–YE ALGORITHM TO THE ANALYTIC CENTER OF THE SOLUTION SET*

CLOVIS C. GONZAGA[†] AND RICHARD A. TAPIA[‡]

Abstract. In this work we demonstrate that the Mizuno–Todd–Ye predictor-corrector primal-dual interior-point method for linear programming generates iteration sequences that converge to the analytic center of the solution set.

Key words. linear programming, primal-dual interior-point algorithm, predictor-corrector algorithm, analytic center

AMS subject classifications. 49M, 65K, 90C

PII. S1052623493243557

1. Introduction and preliminaries.

The basic primal-dual interior-point method for linear programming was originally proposed by Kojima, Mizuno, and Yoshise [6] based on earlier work of Megiddo [11]. This algorithm can be viewed as perturbed (centered) and damped Newton's method applied to the first order conditions for a particular standard form linear program. They established linear convergence of the duality-gap sequence to zero and an iteration complexity of $O(nL)$ for their basic algorithm. Immediately Kojima, Mizuno, and Yoshise in a second paper [7] and Monteiro and Adler [15] proposed algorithms that fit in the original Kojima–Mizuno–Yoshise framework and established linear convergence of the duality-gap sequence to zero and a superior iteration complexity of $O(\sqrt{n}L)$ for their versions of the algorithm. Soon after Mizuno, Todd, and Ye [14] considered a predictor-corrector variant of the Kojima–Mizuno–Yoshise basic algorithm. In their algorithm, the predictor step is a damped Newton step and the corrector step is a perturbed (centered) Newton step. Mizuno, Todd, and Ye also established linear convergence of the duality-gap sequence to zero and an iteration complexity of $O(\sqrt{n}L)$ for their predictor-corrector algorithm.

The literature now abounds with papers concerned with issues related to primal-dual interior-point methods. Moreover, when we discuss convergence or convergence attributes (including complexity) of one of these algorithms, we are in general discussing convergence of the duality-gap to zero. This interpretation has become standard in the area even though convergence of the duality-gap sequence does not imply convergence of the iteration sequence. The convergence of the iteration sequence is certainly an important issue in its own right. Indeed, the earlier works on fast (super-

* Received by the editors January 27, 1993; accepted for publication (in revised form) August 3, 1995. This research was initiated in February 1992 while the first author was visiting the Center for Research on Parallel Computation, Rice University. A preliminary version of this paper was presented at the 4th SIAM Meeting on Optimization, Chicago, IL, May 1992.

http://www.siam.org/journals/siopt/7-1/24355.html

[†] Department of Systems Engineering and Computer Science, COPPE-Federal University of Rio de Janeiro, Cx. Postal 68511, 21945-970 Rio de Janiero, RJ Brazil (clovis@mtm.ufsc.br). Part of this work was done while the author was visiting the Cornell Center for Applied Mathematics, with support from NSF, AFOSR, and ONR through NSF grant DMS-8920550 and from the Center for Applied Mathematics.

[‡] Department of Computational and Applied Mathematics and Center for Research on Parallel Computation, Rice University, Houston, TX 77251-1892 (tapia@masc8.rice.edu). The research of this author was supported in part by NSF cooperative agreements CCR-8809615, AFOSR 89-0363, DOE DEFG05-86ER25017, and ARO 9DAAL03-90-G-0093.

linear) convergence of the duality-gap sequence to zero, i.e., Zhang, Tapia, and Dennis [26], Zhang, Tapia, and Potra [27], Zhang and Tapia [23], Ye, Tapia, and Zhang [21], and McShane [10], all made the assumption that the iteration sequence converged.

In some applications, see, e.g., Charnes, Cooper, and Thrall [2], it is important to obtain a solution that is not near the boundary of the solution set. Hence there is significant value in designing a primal-dual interior-point method for linear programming that converges to the analytic center of the solution set.

Tapia, Zhang, and Ye [17] derived conditions under which the iteration sequence generated by the Kojima–Mizuno–Yoshise primal-dual interior-point method converged. These conditions were essentially the conditions for fast (superlinear) convergence established by Zhang, Tapia, and Dennis [26] (see also Zhang and Tapia [24]). Zhang and Tapia [25] derived conditions under which this iteration sequence converged to the analytic center, assuming that the sequence converged. However, these conditions are not completely compatible with the Tapia–Zhang–Ye conditions for the convergence of the iteration sequence.

Ye et al. [20] and, independently, Mehrotra [13], based on the work of Ye, Tapia, and Zhang [21], demonstrated that the Mizuno–Todd–Ye predictor-corrector algorithm in all cases gives quadratic convergence of the duality-gap sequence to zero. A highlight of this contribution was that the assumption of iteration sequence convergence was not needed (for the first time). Soon after, Zhang and Tapia [24] removed this assumption from the Zhang–Tapia–Dennis theory for superlinear convergence. Quite recently, Zhang and El-Bakry [22] were able to show that a modified version of the Mizuno–Todd–Ye predictor-corrector algorithm had the property that the iteration sequence that it generated converged to the analytic center. Their modified algorithm dynamically chose the steplength in the Newton predictor step so that the corrector step would asymptotically enforce arbitrary close proximity to the central path.

In this paper we show that the predictor-corrector algorithm as originally stated by Mizuno, Todd, and Ye has the property that the iteration sequences (predictor-step sequence and corrector-step sequence) it generates converge to the analytic center of the solution set.

The paper is organized as follows. In the remainder of this section we introduce our notation and several fundamental background notions. In section 2 we discuss the primal-dual Newton step and establish some properties concerning this step. Some mathematical tools concerning projections and scalings are derived in section 3. Central path issues are discussed in section 4. The Mizuno–Todd–Ye predictor-corrector algorithm and some of its properties are presented in section 5. In section 6 we combine all our previous discussions and in Theorem 6.3 demonstrate that the Mizuno–Todd–Ye algorithm generates sequences that converge to the analytic center of the solution set.

Given a vector x, d, ϕ, the corresponding upper case symbols denote (as usual) the diagonal matrix X, D, Φ, defined by the vector.

We denote component-wise operations on vectors by the usual notations for real numbers. Thus, given two vectors u, v of the same dimension, uv, u/v, etc. denotes the vectors with components $u_i v_i$, u_i/v_i, etc. This notation is consistent as long as component-wise operations are given precedence over matrix operations. Note that $uv \equiv Uv$ and if A is a matrix, then $Auv \equiv AUv$, but in general $Auv \neq (Au)v$.

We frequently use the $O(\cdot)$ and $\Omega(\cdot)$ notation to express a relationship between functions. Our most common usage will be associated with a sequence $\{x^k\}$ of vectors and a sequence $\{\mu^k\}$ of positive real numbers. In this case $x = O(\mu)$ or $x^k = O(\mu^k)$ means that there is a constant K (dependent on problem data) such that for every $k \in \mathbb{N}$, $\|x^k\| \leq K\mu^k$. Similarly, $x = \Omega(\mu)$ or $x^k = \Omega(\mu^k)$ means that there is $\epsilon > 0$ such that for every $k \in \mathbb{N}$, $\|x^k\| \geq \epsilon\mu^k$.

The primal and dual linear programming problems are as follows:

$$(LP) \qquad \begin{array}{rrcl} \text{minimize} & c^T x & & \\ \text{subject to} & Ax & = & b, \\ & x & \geq & 0 \end{array}$$

and

$$(LD) \qquad \begin{array}{rrcl} \text{maximize} & b^T y & & \\ \text{subject to} & A^T y + s & = & c, \\ & s & \geq & 0, \end{array}$$

where $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and $A \in \mathbb{R}^{m\times n}$. We assume that both problems have optimal solutions and that the sets of optimal solutions are bounded. This is equivalent to the requirement that both feasible sets contain points satisfying all inequalities strictly.

Given any feasible primal-dual pair $(\tilde{x}, \tilde{s})$, the problems can be rewritten as

$$(LP) \qquad \begin{array}{rrcl} \text{minimize} & \tilde{s}^T x & & \\ \text{subject to} & Ax & = & b, \\ & x & \geq & 0 \end{array}$$

and

$$(LD) \qquad \begin{array}{rrcl} \text{minimize} & \tilde{x}^T s & & \\ \text{subject to} & Bs & = & Bc, \\ & s & \geq & 0, \end{array}$$

where B^T is a matrix whose columns span the null space of A. Popular choices for B^T are an orthonormal basis for the null space of A and $B = P_A$, the projection matrix into the null space of A.

The feasible sets for (LP) and (LD) will be denoted, respectively, by $\mathcal{P}$ and $\mathcal{D}$. Their relative interiors will be, respectively, $\mathcal{P}^0$ and $\mathcal{D}^0$.

The set of optimal solutions for the primal-dual pair of problems constitutes a face $F = F_P \times F_D$ of the polyhedron of feasible solutions, where F_P and F_D are, respectively, the primal and dual optimal faces. By hypothesis, this face is a compact set. It is well known that this face is characterized by a partition $\{B, N\}$ of the set of indices $\{1, \dots, n\}$ such that $F_P = \{x \in \mathcal{P} \mid x_N = 0\}$ and $F_D = \{s \in \mathcal{D} \mid s_B = 0\}$. In the relative interior of the face, $x_B > 0$ and $s_N > 0$.

We study algorithms that generate sequences that converge to the optimal face. Our main concern is with the behavior of the iterates as they approach the optimal face. We want this to happen in such a manner that all limit points are in the relative interior of the optimal face. We shall see later how this condition can be enforced.

Given $\mu > 0$, $\mu \in \mathbb{R}$, the pair (x, s) of feasible primal and dual solutions is the central point $(x(\mu), s(\mu))$ associated with μ if and only if

$$xs = \mu e,$$

where e stands for the vector of all ones, with dimension given by the context.

The central path is the curve in $\mathbb{R}^{2n}$ parametrized by the positive real μ, i.e.,

$$\mu \mapsto (x(\mu), s(\mu)).$$

Thus (x, s) is a central point if and only if

$$\begin{array}{rcl} xs & = & \mu e, \\ Ax & = & b, \\ Bs & = & Bc, \\ x, s & \geq & 0, \end{array} \tag{1}$$

where the columns of B^T span the null space of A.

The first-order or Karush–Kuhn–Tucker (KKT) conditions for problem (LP) (or (LD)) are

$$\begin{array}{rcl} xs & = & 0, \\ Ax & = & b, \\ A^T y + s & = & c, \\ x, s & \geq & 0. \end{array}$$

The perturbed KKT conditions for perturbation parameter $\mu > 0$ are

$$\begin{aligned} xs &= \mu e, \\ Ax &= b, \\ A^T y + s &= c, \\ x, s &\geq 0. \end{aligned} \tag{2}$$

Observe that the perturbed KKT conditions are merely the defining relations for the central path and (2) can equivalently be written as (1). Essentially all primal-dual interior-point methods for problem (LP) consist of some variant of the damped Newton method applied to the perturbed KKT conditions (1) or (2).

2. Newton steps. When dealing with an iterative procedure we will use the superscript 0 to denote the previous iterate, no superscript to denote the current iterate, and a subscript of + to denote the subsequent iterate. In two-step algorithms like the Mizuno–Todd–Ye algorithm described in section 4, this notation will apply to the current iterate, the intermediate iterate, and the final iterate.

Given a strictly feasible pair (x, s), we shall define three parameters:

$$\begin{aligned} \mu(x, s) &= s^T x / n, \\ w(x, s) &= sx / \mu(x, s), \\ \phi(x, s) &= 1 / \sqrt{w(x, s)}. \end{aligned}$$

The first two parameters will be extensively studied below. The parameter ϕ has no special meaning and is introduced because it will simplify many formulas in the text. When no confusion can arise, we drop the reference to the variables and continue to use other symbols in a consistent manner. For example, $\bar{w} = w(\bar{x}, \bar{s})$ or $\phi^0 = \phi(x^0, s^0)$.

Given a strictly feasible pair (x, s), we are interested in finding $(x^+, s^+) = (x, s) + (u, v)$ that solves (1) or (2) with $\mu = \gamma \mu(x, s)$, where $\gamma \in [0, 1]$. The Newton equation for (1) at (x, s) with μ replaced by $\gamma \mu$ can be written

$$\begin{array}{rcl} xv + su & = & -xs + \gamma \mu(x, s) e, \\ u & \in & \mathcal{N}(A), \\ v & \in & \mathcal{R}(A^T), \end{array} \tag{3}$$

where as usual $\mathcal{N}$ denotes null space and $\mathcal{R}$ denotes range space. The solution of (3) is obtained by scaling the equations. Define the scaling matrix by $d = \sqrt{x/s}$, $D = \operatorname{diag}(d_1, \dots, d_n)$, and the scaling

$$(p, q) \to (\bar{p}, \bar{q}) = (d^{-1}p, dq)$$

for general $(p, q) \in (\mathbb{R}^n \times \mathbb{R}^n)$.

The relationship between d and the vector ϕ defined above is

$$d \;=\; \sqrt{\frac{x}{s}} \;=\; \frac{x\phi}{\sqrt{\mu}} \;=\; \frac{\sqrt{\mu}}{s\phi}. \tag{4}$$

When applied to the original pair (x, s), the resulting scaled pair will be

$$(\bar{x}, \bar{s}) \;=\; (\sqrt{xs}, \sqrt{xs}). \tag{5}$$

After scaling, system (3) becomes

$$\begin{array}{rcl} \bar{x}\bar{v} + \bar{s}\bar{u} & = & -\bar{x}\bar{s} + \gamma\mu e, \\ \bar{u} & \in & \mathcal{N}(AD), \\ \bar{v} & \in & \mathcal{R}(DA^T). \end{array} \tag{6}$$

Since $\bar{x} > 0$, the first equation can be multiplied by $\bar{x}^{-1}$, leading to

$$\bar{v} + \bar{u} = -\bar{s} + \gamma\mu\bar{x}^{-1},$$

and the solution is simply the orthogonal decomposition of the vector $-\bar{s} + \gamma\mu\bar{x}^{-1}$ along $\mathcal{N}(AD)$ and its orthogonal complement. Let P_{AD} be the projection matrix into $\mathcal{N}(AD)$, and $\tilde{P}_{AD} = I - P_{AD}$:

$$\begin{array}{rcl} \bar{u} & = & P_{AD}(-\bar{s} + \gamma\mu\bar{x}^{-1}), \\ \bar{v} & = & \tilde{P}_{AD}(-\bar{s} + \gamma\mu\bar{x}^{-1}). \end{array} \tag{7}$$

The Newton step in original coordinates is given by $u = d\bar{u}$ and $v = d^{-1}\bar{v}$.

A convenient formulation is obtained by substituting $d = \frac{1}{\sqrt{\mu}}x\phi$ and $d^{-1} = \frac{1}{\sqrt{\mu}}s\phi$.

$$\begin{array}{rcl} u & = & x\phi P_{AX\Phi}\phi \left(-\dfrac{xs}{\mu} + \gamma e \right) \\ v & = & s\phi \tilde{P}_{AX\Phi}\phi \left(-\dfrac{xs}{\mu} + \gamma e \right). \end{array} \tag{8}$$

We now describe two alternative ways of writing the expression for u (the expressions for v are similar).

Using the definition of w,

$$u = -x\phi P_{AX\Phi}\phi(w - \gamma e). \tag{9}$$

Observing the symmetrical formulation of (LD), we see that for any two feasible dual slacks s^1, s^2, $P_{AD}ds^1 = P_{AD}ds^2 = P_{AD}dc$. In particular, we can choose a fixed dual slack and use it in (7). We shall choose s^*, the analytic center of the dual optimal face, and write

$$u = -dP_{AD}d(s^* - \gamma\mu x^{-1}).$$

By the same process as above,

$$u = -x\phi P_{AX\Phi}\phi\left(\frac{xs^*}{\mu} - \gamma e\right). \tag{10}$$

In section 5 when we study the Mizuno–Todd–Ye predictor-corrector algorithm, we will have need for the following proposition.

PROPOSITION 2.1. *Let* $(\hat{x}, \hat{s})$ *and* (x, s) *be feasible pairs*; $\theta \in [0, 1]$. *Consider* $x^+ = x + u$ *and* $s^+ = s + v$, *where* (u, v) *satisfies*

$$\begin{aligned} \hat{x}v + \hat{s}u &= -\theta xs + \hat{\mu}e, \\ u &\in \mathcal{N}(A), \\ v &\in \mathcal{R}(A^T) \ . \end{aligned}$$

Then

$$\mu(x^+, s^+) = (1 - \theta)\mu(x, s) + \hat{\mu} \ . \tag{11}$$

Proof. Left multiplying by e^T, we obtain

$$\hat{x}^T v + \hat{s}^T u = -\theta x^T s + n\hat{\mu} \ .$$

From the definition,

$$x^{+T} s^+ = x^T s + x^T v + s^T u \ ,$$

since $u^T v = 0$. But $\hat{x}^T v = x^T v$, because $\hat{x} - x \in \mathcal{N}(A)$ and $v \in \mathcal{R}(A^T)$, and similarly $\hat{s}^T u = s^T u$. Substituting in the expressions above we immediately obtain (11). □

Two special cases of system (3) have been studied extensively in the literature. They are as follows:

(i) $\gamma = 0$: the resulting directions (h_x^1, h_s^1) are called the primal-dual affine scaling directions (or pure Newton directions);

(ii) $\gamma = 1$: the resulting directions (h_x^2, h_s^2) are called the constant gap-centering directions.

The first equation of the Newton system (3) can be rewritten as

$$xv + su = -(1 - \gamma)xs + \gamma(-xs + \mu e). \tag{12}$$

This is a combination of the solutions of two systems with

$$\begin{aligned} xv^1 + su^1 &= -xs, \\ xv^2 + su^2 &= -xs + \mu e, \end{aligned} \tag{13}$$

where $\mu = \mu(x, s)$. The complete solution is given by

$$(u, v) = (1 - \gamma)(u^1, v^1) + \gamma(u^2, v^2). \tag{14}$$

It is quite common to use these two directions separately, possibly as a way to simplify the analysis. This is done by the predictor-corrector algorithms that we study in this paper.

3. Mathematical tools. In this section we state some lemmas on projections and scalings that will be useful in the analysis below.

3.1. Properties of scaled projections. In this section we slightly extend results published by Megiddo and Shub [12].

Consider the primal feasible set for (LP):

$$\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$$

and the map h defined for $(d, \rho) \in \mathcal{J} = (\mathbb{R}^n_+ \backslash \{0\}) \times \mathbb{R}^n$ by

$$(d, \rho) \mapsto h(d, \rho) = P_{AD}\rho, \tag{15}$$

where P_{AD} represents the projection matrix into the null space of AD.

We study the behavior of this map when $d > 0, (d, \rho) \to (\bar{d}, \bar{\rho}) \in \mathcal{J}$.

Given $\bar{d}$, we define the index sets $B = \{i = 1, \dots, n \mid \bar{d}_i > 0\}$ and $N = \{i = 1, \dots, n \mid \bar{d}_i = 0\}$. The variables with indices in B are called the large variables, and the others are called small variables. It is difficult to describe the behavior of the small variables $h_N(d, \rho)$ of the scaled projection defined above; the theory of Megiddo and Shub concerns the large variables $h_B(d, \rho)$. We shall describe these results conveniently extended to fit our needs.

By definition of projection, $h(d, \rho)$ solves the problem

$$\begin{array}{rl} \text{minimize} & \|h_N - \rho_N\|^2 + \|h_B - \rho_B\|^2 \\ \text{subject to} & A_B D_B h_B = -A_N D_N h_N. \end{array} \tag{16}$$

Assume now that $h_N(d, \rho)$ is given. Then $h_B(d, \rho)$ solves

$$\begin{array}{rl} \text{minimize} & \|h_B - \rho_B\| \\ \text{subject to} & A_B D_B h_B = -A_N D_N h_N(d, \rho). \end{array} \tag{17}$$

Thus, since $h_N(\bar{d}, \bar{\rho})$ is finite and $\bar{D}_N = 0$, $h_B(\bar{d}, \bar{\rho}) = P_{A_B \bar{D}_B} \bar{\rho}_B$. We shall study the point-to-set mapping θ defined for $d \in \mathbb{R}^n_+$ and $\rho \in \mathbb{R}^n$ by

$$(d, \rho) \mapsto \theta(d, \rho) = \{h_B \in \mathbb{R}^{|B|} \mid A_B D_B h_B = -A_N D_N h_N(d, \rho)\}, \tag{18}$$

near a pair $(\bar{d}, \bar{\rho}) \in \mathcal{J}$. Note that at this point, $\theta(\bar{d}, \bar{\rho}) = \mathcal{N}(A_B \bar{D}_B)$.

LEMMA 3.1. *The point-to-set map defined by* (18) *is continuous at* $(\bar{d}, \bar{\rho}) \in \mathcal{J}$.

Proof.

(i) Upper semicontinuity: consider a sequence $(d^k, \rho^k) \to (\bar{d}, \bar{\rho})$ and h^k_B such that $A_B D^k_B h^k_B = -A_N D^k_N h_N(d^k, \rho^k)$ and h^k_B converges to some point $\bar{h}_B$. We must prove that $A_B \bar{D}_B \bar{h}_B = 0$.

The sequence $h_N(d^k, \rho^k)$ is bounded, because $\|h_N(d^k, \rho^k)\| \leq \|\rho^k\|$, since $h(d^k, \rho^k)$ is a projection. Hence $A_B D^k_B h^k_B \to 0$ and, consequently, $A_B \bar{D}_B \bar{h}_B = 0$, completing this part of the proof.

(ii) Lower semicontinuity: consider an arbitrary point $\bar{h}_B \in \mathcal{N}(A_B \bar{D}_B)$. Given an arbitrary sequence $(d^k, \rho^k) \in \mathbb{R}^n_+ \times \mathbb{R}^n$ and such that $(d^k, \rho_k) \to (\bar{d}, \bar{\rho})$, we must construct h^k_B such that $A_B D^k_B h^k_B = -A_N D^k_N h_N(d^k, \rho^k)$ and $h^k_B \to \bar{h}_B$.

Consider $(d^k, \rho^k) \in \mathbb{R}^n_+ \times \mathbb{R}^n$ and $(d^k, \rho^k) \to (\bar{d}, \bar{\rho})$. Since $d^k_B \to \bar{d}_B > 0$, we lose no generality by assuming that $d^k_B > 0$ for all k. Define $h^k_N = h_N(d^k, \rho^k)$. For each k let $\tilde{h}^k_B$ be a minimum-norm solution of $A_B D^k_B h_B = -A_N D^k_N h^k_N$, where the norm is the weighted Euclidean norm $\|D^k_B \cdot \|$. If A^+_B denotes the pseudoinverse of A_B, then we can write $\tilde{h}^k_B = -D^{k-1}_B A^+_B D^k_N h^k_N$. It follows that $\tilde{h}^k_B \to 0$, since $d^k_B \to \bar{d}_B > 0$ and $D^k_N h^k_N \to 0$. Construct

$$h^k_B = (D^k_B)^{-1} \bar{D}_B \bar{h}_B + \tilde{h}^k_B. \tag{19}$$

Then

$$A_B D_B^k h_B^k = A_B \bar{D}_B \bar{h}_B + A_B D_B^k \tilde{h}_B^k = -A_N D_N^k h_N^k,$$

since $\bar{h}_B \in \mathcal{N}(A_B \bar{D}_B)$. Thus $h_B^k \in \theta(d^k, \rho^k)$. Since $D_B^k \to \bar{D}_B > 0$ and $\tilde{h}_B^k \to 0$, it follows that $h_B^k \to \bar{h}_B$, completing the proof. □

LEMMA 3.2. *Let $h(d,\rho)$ be given by* (15). *Consider $(\bar{d},\bar{\rho}) \in \mathcal{J}$ and $(d^k,\rho^k) \in \mathbb{R}_+^n \times \mathbb{R}^n$ such that $(d^k,\rho^k) \to (\bar{d},\bar{\rho})$. Then*

(i) $h_B(d^k,\rho^k) \to h_B(\bar{d},\bar{\rho}) = P_{A_B\bar{D}_B}\bar{\rho}_B$;

(ii) *if $\bar{\rho}_N = 0$, then $h_N(d^k,\rho^k) \to 0$.*

Proof. (i): the map $(d,\rho) \to \operatorname{argmin}\{\|h_B - \rho_B\| : h_B \in \theta(d,\rho)\}$ is well defined by the uniqueness of the minimizer. It is continuous at $(\bar{d},\bar{\rho})$ as a consequence of the continuity of the point-to-set map θ and the continuity of projections (see, for example, Hogan [4]). From the comment immediately preceding (17) we see that

$$h_B(d^k,\rho^k) = \operatorname{argmin}\{\|h_B - \rho_B^k\| \mid h_B \in \theta(d^k,\rho^k)\}\ .$$

Hence, from continuity, $h_B(d^k,\rho^k) \to h_B(\bar{d},\bar{\rho})$. From the comment immediately following (17) we see that $h_B(\bar{d},\bar{\rho}) = P_{A_B\bar{D}_B}\bar{\rho}_B$. This establishes part (i).

(ii): here we follow a similar proof in Megiddo and Shub [12]. Assume that $\bar{\rho}_N = 0$ and by contradiction that for some sequence $d^k \to \bar{d}$, $\rho^k \to \bar{\rho}$ we have $h_N(d^k,\rho^k) \to \bar{h}_N \neq 0$. Define $\epsilon = \|\bar{h}_N\|^2 > 0$. We have the following:

$$\|h(d^k,\rho^k) - \rho^k\|^2 = \|h_B(d^k,\rho^k) - \rho_B^k\|^2 + \|h_N(d^k,\rho^k) - \rho_N^k\|^2.$$

By (i), $h_B(d^k,\rho^k) \to \bar{h}_B$, where $\bar{h}_B = P_{A_B\bar{D}_B}\bar{\rho}_B$. For sufficiently large k,

$$\|h_B(d^k,\rho^k) - \rho_B^k\|^2 > \|\bar{h}_B - \bar{\rho}_B\|^2 - \epsilon/2. \tag{20}$$

Now construct the following sequence:

$$\tilde{h}_B^k = (D_B^k)^{-1}\bar{D}_B\bar{h}_B, \quad \tilde{h}_N^k = 0.$$

It follows that $\tilde{h}_B^k \to \bar{h}_B$ and $\tilde{h}^k \in \mathcal{N}(AD^k)$, since $AD^k\tilde{h}^k = A_B\bar{D}_B\bar{h}_B = 0$.

Comparing this with (20), we have for k sufficiently large $\|\tilde{h}^k - \rho^k\| < \|h(d^k,\rho^k) - \rho^k\|$ and $\tilde{h}^k \in \mathcal{N}(AD^k)$, contradicting the definition of $h(d^k,\rho^k) = P_{AD^k}\rho^k$ and completing the proof. □

3.2. Shifted scalings. This section contains some useful consequences of scalings on projections and norms. The first lemma concerns projections and slightly shifted scalings.

LEMMA 3.3. *Let $q \in \mathbb{R}^n$ be such that $\|q - e\|_\infty \leq \alpha$, where $\alpha \in (0, 0.25)$, and consider the projections $\hat{h} = P_A\rho$, $h = qP_{AQ}q\rho$. Then $\|h - \hat{h}\| \leq 3\alpha\|\hat{h}\|$.*

Proof. Note that since $\rho = \hat{h} + A^T w$ for some $w \in \mathbb{R}^m$,

$$q\rho = q\hat{h} + (AQ)^T w,$$

and thus

$$P_{AQ}q\rho = P_{AQ}q\hat{h}.$$

It follows that

$$q^{-1}h = P_{AQ}q\hat{h}.$$

On the other hand, by definition of projection,

$$q\hat{h} = P_{AQ}q\hat{h} + z,$$

where $z \in \mathcal{R}(QA^T)$. Merging the last expressions, we get

$$q\hat{h} = q^{-1}h + z,$$

where $q^{-1}h \in \mathcal{N}(AQ)$ and $z \in \mathcal{R}(QA^T)$. Subtracting $q^{-1}\hat{h} \in \mathcal{N}(AQ)$ from both sides,

$$(q^{-1} - q)\hat{h} = q^{-1}(h - \hat{h}) + z,$$

and from the orthogonality of the right-hand side terms,

$$\|(q^{-1} - q)\hat{h}\| \geq \|q^{-1}(h - \hat{h})\|.$$

Now use the following facts: $\|(h - \hat{h})\| \leq \|q\|_\infty \|q^{-1}(h - \hat{h})\|$ and $\|(q^{-1} - q)\hat{h}\| \leq \|(q^{-1} - q)\|_\infty \|\hat{h}\|$. Combining these three expressions leads to

$$\|h - \hat{h}\| \leq \|q\|_\infty \|q^{-1} - q\|_\infty \|\hat{h}\|.$$

But $\|q\|_\infty \|q^{-1} - q\|_\infty \leq (1+\alpha)\,(1/(1-\alpha) - (1-\alpha)) \leq 3\alpha$ which is easily verified for $\alpha \in (0, 0.25)$, completing the proof. □

Our second lemma concerns scaled norms. Given a vector $x \in \mathbb{R}^n_{++}$, the following map defines a norm:

$$h \in \mathbb{R}^n \mapsto \|h\|_x = \|x^{-1}h\|.$$

This is the Euclidean norm of the vector corresponding to h after a scaling $\bar{h} = x^{-1}h$. This norm is very usual in interior-point methods, because it characterizes the proximity from a point to a central point in the following sense: let $x(\mu)$ be the primal central point associated with the parameter $\mu > 0$. If $\|x - x(\mu)\|_x \leq \delta < 1$, then a Newton centering iteration from x produces an efficient centering step (which is usually imprecisely stated as being in the region of quadratic convergence of Newton's method).

In the same fashion that we defined the scaled Euclidean norm $\|h\|_x$, we define the scaled norm $\|h\|_x^\infty$. The following lemma relates the scaled norms for different reference points.

LEMMA 3.4. *Consider $x, y \in \mathbb{R}^n_{++}$, $h \in \mathbb{R}^n$, $\alpha \in (0,1)$. If either $\|x - y\|_x^\infty \leq \alpha$ or $\|x - y\|_y^\infty \leq \alpha$, then*

$$\|h\|_x \leq \frac{1}{1-\alpha}\|h\|_y.$$

Proof. To begin with,

$$\|h\|_x = \left\|\frac{h}{x}\right\| = \left\|\frac{y}{x}\frac{h}{y}\right\| \leq \left\|\frac{y}{x}\right\|_\infty \|h\|_y.$$

If $\|x - y\|_x^\infty \leq \alpha$, then $|(x_i - y_i)/x_i| \leq \alpha$ or $1 - y_i/x_i \geq -\alpha$, which implies $y_i/x_i \leq 1+\alpha \leq 1/(1-\alpha)$. In the other case, $|(x_i - y_i)/y_i| \leq \alpha$ or $x_i/y_i \geq 1-\alpha$, which implies $y_i/x_i \leq 1/(1-\alpha)$, completing the proof. □

4. Trajectories, centrality, and proximity. The primal-dual central path defined above is contained in the set of interior points and ends at a point (x^*, s^*) in the relative interior of the optimal face. This point is the analytic center of the face. See problem (24) for an equivalent characterization. For more detail, see McLinden [9] and Sonnevend [16].

In this section we study (primal-dual) proximity criteria that describe how far a pair (x, s) is from the primal-dual central path and then study (primal) proximity criteria to evaluate how far a point in the optimal face is from its analytic center.

4.1. Primal-dual proximity. Given an interior pair (x, s) and a parameter $\mu > 0$ (not necessarily equal to $\mu(x, s)$), the proximity of (x, s) in relation to $(x(\mu), s(\mu))$ is measured by

$$\delta(x, s, \mu) = \left\| \frac{xs}{\mu} - e \right\|. \tag{21}$$

When $\mu = \mu(x, s)$, this is the proximity with relation to the central path

$$\delta(x, s) = \left\| \frac{xs}{\mu(x, s)} - e \right\| = \| w(x, s) - e \|. \tag{22}$$

Let us compute the proximity at the pair (x^+, s^+) resulting from the Newton step described in (3), with $\mu = \mu(x, s)$. We have

$$\begin{aligned} x^+ s^+ &= (x + u)(s + v) \\ &= xs + xv + su + uv \\ &= \gamma\mu e + uv. \end{aligned}$$

Premultiplying the expression above by e^T and noting that $u^T v = 0$, we arrive to $\mu(x^+, s^+) = \gamma\mu$, and thus

$$\frac{x^+ s^+}{\mu(x^+, s^+)} - e = \frac{uv}{\mu(x^+, s^+)}$$

or

$$\delta(x^+, s^+) = \left\| \frac{uv}{\gamma\mu} \right\| = \left\| \frac{uv}{\mu(x^+, s^+)} \right\|. \tag{23}$$

A fundamental result on the effect of the Newton step on proximity is given in the following lemma. This result, due to Mizuno, Todd, and Ye, can be found in [14].

LEMMA 4.1. *Consider an interior pair* (x, s) *and a parameter* $\mu^+ > 0$. *If* $\delta(x, s, \mu^+) = \delta \leq 0.5$, *then* $\delta(x^+, s^+) \leq \delta^2/\sqrt{2}$.

The primal-dual affine-scaling directions are the solution of (3) with $\gamma = 0$. These directions associated with each interior feasible pair (x, s) generate a continuous vector field, which extends continuously to the boundary.

This vector field was thoroughly studied by Adler and Monteiro [1], who describe the trajectories generated by it and the derivatives of these trajectories. The trajectories are parametrized by μ, and there is one trajectory passing through each interior pair (x, s).

For each interior pair (x, s), we defined the vector $w(x, s) = xs/\mu(x, s)$. Each trajectory is associated with this vector in the following two ways:

(i) the trajectory associated with $w > 0$ is composed of the pairs (x, s) such that

$$\frac{xs}{\mu(x,s)} = w.$$

In particular, the central path is the trajectory associated with $w = e$;

(ii) the trajectory associated with $w > 0$ is composed of the minimizer pairs of the parametrized primal-dual penalized function

$$x^T s - \mu \sum_{i=1}^{n} w_i \ln x_i - \mu \sum_{i=1}^{n} w_i \ln s_i.$$

Each trajectory is composed of interior points and ends in the relative interior of the optimal face.

In what follows, we assume that the vectors $w(x, s)$ are always in a compact set defined by

$$\|w(x,s) - e\| \leq \alpha,$$

where $\alpha \in (0, 1)$.

When the weight vectors w are in a compact set bounded away from the boundary of the positive orthant, the trajectories end in the relative interior of the optimal face. Specifically at the limit of the minimizers of the parametrized barrier function, we have

$$\begin{aligned} x^*(w) &= \operatorname{argmin} \left\{ -\sum_{i \in B} w_i \ln x_i \mid x \in F_P \right\}, \\ s^*(w) &= \operatorname{argmin} \left\{ -\sum_{i \in N} w_i \ln s_i \mid x \in F_D \right\}. \end{aligned}$$

In particular, the central path ends at the analytic center of the optimal face $(x^*, s^*) = (x^*(e), s^*(e))$.

The sets of end points of all trajectories for such weights w are sets of minimizers of parametrized continuously differentiable functions and are compact. It is easy to see that the nonzero variables are all bounded away from zero, because the compact sets are in the relative interior of the optimal faces. This is also clear from the fact that the barrier functions become arbitrarily large as the boundaries of the faces are approached.

Similarly, all the trajectories in the bundle associated with this compact set of parameter vectors are in the relative interior of the feasible set and bounded away from the nonoptimal faces.

4.2. Primal proximity. We shall summarize some facts about the analytic center of a polytope and derive properties of descent methods for finding the center.

Consider the primal centering problem

$$\begin{array}{rrcl} \text{minimize} & p(x) & = & -\sum_{i=1}^{n} \ln(x_i) \\ \text{subject to} & Ax & = & b, \\ & x & > & 0, \end{array} \tag{24}$$

where $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m\times n}$, such that its feasible region, S^0, is nonempty, with compact closure S. The analytic center of S is the unique optimal solution of (24);

$$\chi = \operatorname*{argmin}_{x\in S^0} p(x).$$

The analytic center was defined by Sonnevend [16]; see also McLinden [9]. Its properties and the description of the Newton primal centering algorithm (SSD algorithm) are described in Gonzaga [3]. The following facts come from the latter reference.

Given a point $x \in S^0$, the Newton centering direction from x is given by $h(x) = x\bar{h}(x)$, where

$$\bar{h}(x) = -P_{AX}e$$

is the centering direction after scaling the problem so that the point x is taken to e.

The (primal) proximity of x in relation to χ, defined above, is given by

$$\delta(x) = \|\bar{h}(x)\| = \|h(x)\|_x, \tag{25}$$

where $\|\cdot\|_x$ is the norm relative to x.

The following important results are described, for example, in [3]. Let $x \in S^0$ be such that $\delta(x) = \delta < 1$, then

$$\begin{array}{rcl} \|x-\chi\|_x & \leq & \dfrac{\delta}{1-\delta}, \\ \delta(x+h(x)) & \leq & \delta^2. \end{array} \tag{26}$$

The first result above gives an upper bound for $\|x-\chi\|_x$. We shall also need a lower bound for this distance, and this will be provided by the next lemma.

Lemma 4.2. *If $\delta(x) = \delta < 0.5$, then*

$$\|x-\chi\|_x \geq \frac{1-2\delta}{1-\delta}\delta.$$

In particular, if $\delta \leq 0.09$, then $\|x-\chi\|_x \in [0.9\delta, 1.1\delta]$.

Proof. Let $x^+ = x + h(x)$. We know that $\|h(x)\|_x = \delta$ and that $\delta(x^+) \leq \delta^2$. It follows from (26) that

$$\|x^+ - \chi\|_{x^+} \leq \frac{\delta^2}{1-\delta^2},$$

and hence

$$\|x^+ - \chi\|_x \leq \left\|\frac{x^+}{x}\right\|_\infty \frac{\delta^2}{1-\delta^2}.$$

But $x^+/x = e + h(x)/x$, and thus

$$\left\|\frac{x^+}{x}\right\|_\infty \leq 1 + \left\|\frac{h(x)}{x}\right\| \leq 1+\delta.$$

It follows that

$$\|x^+ - \chi\|_x \leq (1+\delta)\frac{\delta^2}{1-\delta^2} = \frac{\delta^2}{1-\delta}.$$

Finally,

$$\begin{aligned}\|x-\chi\|_x &= \|x-x^+ + x^+ - \chi\|_x\\ &\geq \|x-x^+\|_x - \|x^+-\chi\|_x\\ &\geq \delta - \frac{\delta^2}{1-\delta}\\ &= \frac{1-2\delta}{1-\delta}\delta.\end{aligned}$$

The numeric values are obtained by substitution, completing the proof. □

This lemma shows that when the proximity measure is small, it is indeed a good approximation to the actual scaled distance to the center. The values $\delta \leq 0.09$ will be quite reasonable for our analysis below.

One final technical result also will be useful below. It reproduces the bounds above using the norm relative to χ.

LEMMA 4.3. *If $\delta(x) = \delta \leq 0.1$, then for $x^+ = x + h(x)$,*

$$\begin{aligned}\|x^+-\chi\|_\chi &\leq 1.05\delta^2,\\ \|x-\chi\|_\chi &\geq 0.75\delta.\end{aligned}$$

Proof. Using (26), $\|x^+-\chi\|_{x^+} \leq \delta^2/(1-\delta^2)$, since $\delta(x^+) \leq \delta^2$. Using Lemma 3.4 with $\alpha = \delta^2/(1-\delta^2)$, we obtain $\|x^+-\chi\|_\chi \leq \delta^2/(1-2\delta^2)$. The first result in the lemma follows from this with $\delta = 0.1$.

Using Lemma 4.2, $\|x-\chi\|_x \geq \delta(1-2\delta)/(1-\delta)$. From (26), $\|x-\chi\|_x \leq \delta/(1-\delta)$. Using Lemma 3.4 with $\alpha = \delta/(1-\delta)$, we get $\|x-\chi\|_\chi \geq (1-\alpha)\|x-\chi\|_x$. Manipulating these expressions, we arrive at

$$\|x-\chi\|_\chi \geq \left(\frac{1-2\delta}{1-\delta}\right)^2 \delta.$$

Substituting $\delta = 0.1$, we obtain the second result, therefore completing the proof. □

The primal centering direction $h(x)$ is the Newton direction for $p(\cdot)$ from x, and it coincides with the steepest descent direction for $x = e$; i.e., $\bar{h}(x)$ is the Cauchy direction from e. To see this, notice that $h(x) = -xP_{AX}x\nabla p(x) = xP_{AX}xx^{-1}$.

Other scalings give rise to descent directions that are in general not as efficient as this one. We shall apply Lemma 3.3 to study the effect of slightly shifted scalings on the descent directions.

5. The Mizuno–Todd–Ye algorithm.

The MTY algorithm is a path-following predictor-corrector algorithm. All activity is restricted to a region near the central path; i.e., all points (x, s) generated by the algorithm satisfy

$$\delta(x,s) = \|w(x,s) - e\| = \left\|\frac{xs}{\mu(x,s)} - e\right\| \leq \alpha,$$

where $\alpha \in (0, 0.5)$.

ALGORITHM 5.1. *Given $\alpha \leq 0.3$, (x^{0^1}, s^{0^1}) such that $\delta(x^{0^1}, s^{0^1}) \leq \alpha^2/\sqrt{2}$, $k = 1$.*

```
REPEAT
    $x^0 := x^{0^k}$, $s^0 := s^{0^k}$.
  Predictor: Given $(x^0, s^0)$ compute the (affine-scaling) step $(u^0, v^0)$, and let
      $x = x^0 + u^0$, $s = s^0 + v^0$ where $(u^0, v^0)$ is defined by
```

```
        $x^0v^0 + s^0u^0 = -\theta x^0s^0, \quad u^0 \in \mathcal{N}(A),\ v^0 \in \mathcal{R}(A^T)$,
        with $\theta \in (0,1]$ such that $(x,s)$ is feasible and $\delta(x,s) \le \alpha$. (The specific
        value of $\theta$ will be discussed below.)
    Corrector: Given $(x,s)$ compute the (centering) step $(u,v)$ and let $x^+ = x+u$,
        $s^+ = s+v$, where $(u,v)$ is defined by
        $xv + su = -xs + \mu e, \quad u \in \mathcal{N}(A),\ v \in \mathcal{R}(A^T)$,
        with $\mu = \mu(x,s)$.
    Subsequent iterate:
        ${x^0}^{k+1} = x^+$, ${s^0}^{k+1} = s^+$,
        $k = k+1$,
    UNTIL convergence.
```

Observe that our θ in the predictor step is effectively a steplength parameter. To see this, let us denote the predictor step by $(u^0(\theta), v^0(\theta))$. Then

$$\theta(u^0(0), v^0(0)) = (u^0(\theta), v^0(\theta))$$

and

$$(x,s) = (x^0, s^0) + \theta(u^0(0), v^0(0)),$$

which is the usual way of writing the MTY predictor step. Our choice for θ will be the usual one: $\theta = \theta^k$, the largest $\theta \in (0,1]$ such that $\delta(x(\theta), s(\theta)) \le \alpha$ for all $0 \le \theta \le \theta^k$. For further detail see, for example, section 2 of Ye et al. [20].

From Proposition 2.1, with $(\hat{x}, \hat{s}) = (x^0, s^0)$, $\hat{\gamma} = \gamma$, and $\hat{\mu} = 0$, we see that from the predictor step we get $\mu(x,s) = \gamma\mu(x^0, s^0)$. Also, from the same proposition with $(\hat{x}, \hat{s}) = (x,s)$, $\hat{\gamma} = 0$, and $\hat{\mu} = \mu(x,s)$, we see that from the corrector step we get $\mu(x^+, s^+) = \mu(x,s)$. Hence we have $\mu(x^+, s^+) = \mu(x,s) = \gamma\mu(x^0, s^0)$.

We now list some properties of this algorithm. Some proofs are presented here for the sake of completeness. The proofs that are not given here can be found in Mizuno, Todd, and Ye [14]. Mizuno, Todd, and Ye proved that the algorithm is well defined in the sense that the centering step produces (x^+, s^+) such that $\delta(x^+, s^+) \le \alpha^2/\sqrt{2}$.

Bounds on the quantities appearing in the algorithm are given in the lemmas below. Let $\{B, N\}$ be the optimal partition for the linear programming problem, i.e., the index partition associated with the optimal face. As we described in section 4.1, the central path ends at the analytic center of the optimal face, and the pairs (x,s) such that $\|w(x,s) - e\| \le \alpha$ constitute a neighborhood of the central path bounded away from the nonoptimal faces of the feasible polyhedron and correspond to a bundle of w-weighted affine-scaling trajectories. For α small, the bundle of trajectories ends in a compact neighborhood of the analytic center of the optimal face, and so all the sequences generated by the algorithm are in compact sets.

Hence, the algorithm behaves as follows. As the optimal face is approached (and this happens in polynomial time), $x_N^k \to 0$, $s_B^k \to 0$, and x_B^k, s_N^k stay in small neighborhoods of x_B^*, s_N^*, the analytic centers of the primal and dual optimal faces.

LEMMA 5.1. *Let (x^0, s^0) be such that $\delta(x^0, s^0) \le 0.1$ and consider the quantities generated by a step of the MTY algorithm originated in (x^0, s^0). Then*

(i) $x_N = O(\mu)$, $s_B = O(\mu)$, $x_N^0 = O(\mu^0)$, $s_B^0 = O(\mu^0)$,

(ii) $u^0 = O(\mu^0)$, $v^0 = O(\mu^0)$,

(iii) $u_N = O(\mu)$, $v_B = O(\mu)$.

Proof. All of these bounds are implicit in the technical results given in section 3 of Ye et al. [20]. Specifically, (ii) follows from Lemma 3.2 and Theorem 3.1 in [20].

The tools used there can also be used to establish (i) and (iii). Hence we will not include a proof and direct the reader to that paper for proofs. ∎

The lemma above shows that all the variations in (x, s) due to an MTY step are bounded by $O(\mu^0)$, with exception of u_B and v_N. These are the variations in the large variables due to the corrector step.

6. Convergence of the MTY algorithm. In this section we establish the main result of the paper: the points generated by the MTY algorithm always converge to the analytic center of the optimal face. We shall assume that the optimal face is not a single point. Our convergence proofs will be carried out for primal solutions. The symmetric results for dual slacks can always be proved by the same methods using the complete symmetry of conditions (1).

We begin by studying the map that results from the algorithm. Towards this end we describe the relationship between primal-dual pairs (x^0, s^0) and the result (x^+, s^+) of an MTY step originating at (x^0, s^0). It is essential to keep in mind that at this point we are not studying sequences generated by the algorithm. We derive a lemma (a main result of the paper) on the boundary behavior of the algorithmic map for sequences with strong convergence properties; a second lemma extends the result to nonconvergent sequences and provides the main convergence property of the algorithmic map.[1] We then consider a sequence generated by the algorithm and prove in Theorem 6.3 that it converges to the analytic center of the optimal face.

Consider a sequence of interior primal-dual pairs (x^{0^k}, s^{0^k}) and all the quantities that would be generated by applying one MTY step from each of these points, namely (u^{0^k}, v^{0^k}), (x^k, s^k), (u^k, v^k), (x^{+^k}, s^{+^k}), μ^{0^k}, $\mu^k = \gamma^k \mu^{0^k}$, w^{0^k}, w^k, ϕ^{0^k}, ϕ^k. Again, we stress the fact that presently $(x^0, s^0)^{k+1}$ is not necessarily related to $(x^+, s^+)^k$. Recall that we are denoting the analytic center by (x^*, s^*). Also, the $\{B, N\}$ partition of the indices $\{1, \ldots, n\}$ is the partition associated with the optimal face of the linear program in question. Our main interest is in measuring how the large variables approach x^*_B. A good metric for measuring this is given by the norm $\| \cdot \|_{x^*_B}$, defined on $\mathbb{R}^{|B|}$. To simplify notation, we write

$$\| \cdot \|_* \equiv \| \cdot \|_{x^*_B}.$$

Lemma 6.1. *Let (x^{0^k}, s^{0^k}) be such that $\delta(x^{0^k}, s^{0^k}) \leq 0.1$ and assume that $\mu^{0^k} \to 0$, $(x^{0^k}, s^{0^k}) \to (\bar{x}, \bar{s})$, and $w^{0^k} \to \bar{w}^0$. We then have the following:*

(i) *if $\bar{x} = x^*$, then $u^k \to 0$ and $x^{+^k} \to x^*$;*

(ii) *if $\bar{x} \neq x^*$, then for sufficiently large k,*

$$\|x_B^{+^k} - x_B^*\|_* \leq 0.8 \|x_B^{0^k} - x_B^*\|_*.$$

Proof. The proof consists of two technical parts and a conclusion. In the first part we analyze the boundary behavior of the MTY steps; in the second part we describe the centering direction from $\bar{x}$ in the optimal face. Finally, the conclusion is reached from the comparison of the results of the first two parts.

We begin by considering MTY steps. From Lemma 5.1, $(u^{0^k}, v^{0^k}) \to 0$ and, consequently, $(x^k, s^k) \to (\bar{x}, \bar{s})$. From the same lemma, $u_N^k \to 0$. We must describe

[1] The reader might consider Lemma 6.2 before going through the technical proof of Lemma 6.1.

the behavior of u_B^k. From (10),

$$u^k = -x^k \phi^k P_{AX^k\Phi^k} \phi^k \left(\frac{x^k s^*}{\mu^k} - e \right).$$

We are now in a position to use Lemma 3.2 with $d = x\phi$ and $\rho = -\phi \left(\frac{xs^*}{\mu} - e \right)$.

Our first task is to show that these two sequences converge. By hypothesis, $\|\omega(x^{0k}, s^{0k}) - e\| \leq 0.1$. Hence $\|\omega(\bar{x}, \bar{s}) - e\| \leq 0.1$. It follows that $\omega(\bar{x}, \bar{s}) > 0$. We observed that (x^k, s^k) also converges to $(\bar{x}, \bar{s})$. This means that $\phi(x^k, s^k)$ converges to $\bar{\phi} = \omega(\bar{x}, \bar{s})^{-\frac{1}{2}} > 0$. We have demonstrated that d^k converges to $\bar{d} = \bar{x}\bar{\phi}$. Now, s^k converges to $\bar{s}$ and $\omega^k = \frac{x^k s^k}{\mu^k}$ converges to $\bar{\omega}$ implies that $\frac{x_N^k}{\mu^k}$ converges to $\bar{s}_N^{-1}\bar{\omega}_N$, and hence ρ_N^k converges. Since $s_B^* = 0$, we see that $\rho_B^k = \phi_B^k$. This shows that both d^k and ρ^k converge. We can now apply Lemma 3.2 to obtain

$$u_B^k \to \bar{u}_B = \bar{x}_B \bar{\phi}_B P_{A_B \bar{X}_B \bar{\Phi}_B} \bar{\phi}_B. \tag{27}$$

Since $x^{+^k} = x^{0^k} + u^{0^k} + u^k$ and $u^{0^k} \to 0$, $u_N^k \to 0$,

$$x^{+^k} \to \bar{x}^+ = \bar{x} + \bar{u},$$

where $\bar{u}_N = 0$.

Our attention now goes to centering in the optimal face. Consider the following primal centering direction associated with each (x^{0^k}, s^{0^k}):

$$h^k = -x^{0^k} P_{AX^{0^k}} \left(\frac{x^{0^k} s}{\mu^{0^k}} - e \right), \tag{28}$$

where s is an arbitrary dual slack (remember that $dP_{AD}ds = dP_{AD}ds'$ for any dual slacks s, s' and any scaling $d > 0$).

With $s = s^{0^k}$, we see that $h^k = -x^{0^k} P_{AX^{0^k}}(w^{0^k} - e)$. It follows that $\bar{h}_N = 0$ and

$$\|h^k\|_{x^{0^k}} \leq \|w^{0^k} - e\| = \delta(x^{0^k}, s^{0^k}) \leq 0.1.$$

We now consider (28) with $s = s^*$. Lemma 3.2 with $d = x^0$ and $\rho = -\frac{x^0 s^*}{\mu^0} + e$ can be used to determine the behavior of h^k once we demonstrate that d^k and ρ^k converge. In this case d^k converges by hypothesis. Moreover, an argument similar to the one used above will show that ρ^k converges. Hence Lemma 3.2 applies, so $h^k \to \bar{h}$. From these latter two arguments we have that

$$\bar{h}_N = 0, \quad \bar{h}_B = \bar{x}_B P_{A_B \bar{X}_B} e_B, \ \text{ and } \ \|\bar{h}_B\|_{\bar{x}_B} \leq 0.1.$$

We conclude that $\bar{h}$ is the Newton centering direction in the optimal face and that the proximity measure of $\bar{x}$ is

$$\delta(\bar{x}_B) = \|\bar{h}_B\|_{\bar{x}_B} \leq 0.1.$$

Let $z = \bar{x} + \bar{h}$ be the result of a primal centering step. Then by Lemma 4.3,

$$\begin{array}{rcl} \|\bar{x}_B - x_B^*\|_* & \geq & 0.75\delta(\bar{x}_B), \\ \|z_B - x_B^*\|_* & \leq & 1.05\delta^2(\bar{x}_B). \end{array} \tag{29}$$

Our attention now turns to shifted scaling. We study the effect of the direction $\bar{u}_B$ defined in (27), when it is used for primal centering instead of $\bar{h}$. The quantity

$$\bar{u}_B = \bar{x}_B \bar{\phi}_B P_{A\bar{X}_B\bar{\Phi}_B} \bar{\phi}_B$$

corresponds to $\bar{h}_B$ by way of a shifted scaling. Here $\bar{\phi} = 1/\sqrt{\bar{w}}$, as usual. Since $\|\bar{w} - e\| \leq 0.1$, it follows that for $i = 1, \ldots, n$, $\bar{w}_i \in [0.9, 1.1]$, and it is trivial to check that $\bar{\phi}_i \in [0.9, 1.1]$. Hence $\left\|\bar{\phi} - e\right\|_\infty \leq 0.1$ and, by Lemma 3.3,

$$\|\bar{h}_B - \bar{u}_B\|_{\bar{x}_B} \leq 0.3\|\bar{h}_B\|_{\bar{x}_B} = 0.3\delta(\bar{x}_B). \tag{30}$$

If $\bar{x} = x^*$, then $\delta(\bar{x}_B) = 0$ and it follows that $\bar{h}_B = \bar{u}_B = 0$. This proves part (i) of the lemma. Assume from here on that $\|\bar{x}_B - x_B^*\| \neq 0$.

We need (30) in the norm $\|\cdot\|_*$. Using (26), define

$$\alpha = \|\bar{x}_B - x_B^*\|_{\bar{x}_B} \leq \frac{\delta(\bar{x}_B)}{1 - \delta(\bar{x}_B)} \leq \frac{0.1}{0.9}.$$

Using Lemma 3.4,

$$\|\bar{h}_B - \bar{u}_B\|_* \leq \frac{1}{1 - \alpha}\|\bar{h}_B - \bar{u}_B\|_{\bar{x}_B}.$$

Merging this and (30) with $1/(1 - \alpha) \leq 1.2$ we obtain

$$\|\bar{h}_B - \bar{u}_B\|_* \leq 0.4\delta(\bar{x}_B). \tag{31}$$

And now we compare the points $z_B = \bar{x}_B + \bar{h}_B$ and $\bar{x}_B^+ = \bar{x}_B + \bar{u}_B$, using (29). Specifically,

$$\begin{aligned}
\|\bar{x}_B^+ - x_B^*\|_* &\leq \|z_B - x_B^*\|_* + \|\bar{x}_B^+ - z_B\|_* \\
&= \|z_B - x_B^*\|_* + \|\bar{u}_B - \bar{h}_B\|_* \\
&\leq 1.05\delta^2(\bar{x}_B) + 0.4\delta(\bar{x}_B) \\
&\leq 0.51\delta(\bar{x}_B).
\end{aligned}$$

Using (29), we conclude that

$$\frac{\|\bar{x}_B^+ - x_B^*\|_*}{\|\bar{x}_B - x_B^*\|_*} \leq \frac{0.51}{0.75} \leq 0.7.$$

Finally, we conclude from this expression that since $x^{0^k} \to \bar{x}$ and $x^{+^k} \to \bar{x}^+$ for sufficiently large k,

$$\|x_B^{+^k} - x_B^*\|_* \leq 0.8\|x_B^{0^k} - x_B^*\|_*,$$

completing the proof. ◻

The lemma above studies convergent sequences (x^{0^k}, s^{0^k}). The next lemma shows that the reduction in distance from x^* can be extended uniformly for nonconvergent sequences.

LEMMA 6.2. *Let (x^{0^k}, s^{0^k}) be such that $\delta(x^{0^k}, s^{0^k}) \leq 0.1$ and $\mu^{0^k} \to 0$. Then there exists a sequence of positive reals ϵ^k such that $\epsilon^k \to 0$ and for sufficiently large k,*

$$\left\|x_B^{+^k} - x_B^*\right\|_* \leq \max\left\{\epsilon^k, 0.8\left\|x_B^{0^k} - x_B^*\right\|_*\right\}.$$

Proof. Assume by contradiction that there exist $\epsilon > 0$ and an infinite subsequence of (x^{0^k}, s^{0^k}) with indices $\mathcal{K}^0 \subset \mathbb{N}$ such that for $k \in \mathcal{K}^0$,

$$\left\|x_B^{+^k} - x_B^*\right\|_* > \epsilon, \quad \left\|x_B^{+^k} - x_B^*\right\|_* > 0.8\left\|x_B^{0^k} - x_B^*\right\|_*. \tag{32}$$

The sequences (x^{0^k}, s^{0^k}), (w^{0^k}), (w^k) are all in compact sets by construction, and thus there must exist an infinite subsequence with indices $\mathcal{K} \subset \mathcal{K}^0$ such that these three sequences are convergent in $\mathcal{K}$.

In particular, $(x_B^{+^k})_{\mathcal{K}}$ does not converge to x_B^*, due to (32). Applying Lemma 6.1(i), we see that $(x^{0^k})_{\mathcal{K}}$ does not converge to x^*, and thus (ii) must hold for this subsequence. This contradicts (32), completing the proof. □

Finally, we are ready to establish our convergence result.

THEOREM 6.3. *Consider sequences* (x^{0^k}, s^{0^k}), (x^k, s^k) *generated by the MTY algorithm. Then* $(x^{0^k}, s^{0^k}) \to (x^*, s^*)$ *and* $(x^k, s^k) \to (x^*, s^*)$, *where* (x^*, s^*) *is the analytic center of the solution set.*

Proof. We prove the result for the primal variables. The proof for the dual slacks is similar. Also, it is enough to prove that $x^{0^k} \to x^*$, since $u^{0^k} = O(\mu^{0^k}) \to 0$.

Assume by contradiction that the sequence $\{x^{0^k}\}$ has an accumulation point $\bar{x} \neq x^*$. Since $\bar{x}_N = x_N^* = 0$, we have

$$\sigma \equiv \|\bar{x}_B - x_B^*\|_* > 0.$$

Let $\{\epsilon^k\}$ be the sequence guaranteed by Lemma 6.2, and let $\bar{k}$ be such that the conclusions of that lemma are valid for $k \geq \bar{k}$. Choose an index $j \geq \bar{k}$ such that $\|x_B^{0^j} - x_B^*\|_* < 1.1\sigma$ and such that for $k \geq j$, $\epsilon^k < 0.5\sigma$. This index exists because $\epsilon^k \to 0$ and $\bar{x}_B$ is an accumulation point of $\{x_B^{0^k}\}$.

We prove by induction that for any $k > j$, $\|x_B^{0^k} - x_B^*\|_* < 0.9\sigma$.

(a) $\|x_B^{0^{j+1}} - x_B^*\|_* < 0.8 \times 1.1\sigma < 0.9\sigma$ by Lemma 6.2.

(b) Assume that for an index $k > j$, $\|x_B^{0^k} - x_B^*\|_* < 0.9\sigma$. Then, by Lemma 6.2, $\|x_B^{0^{k+1}} - x_B^*\|_* \leq \max\{\epsilon^k, 0.8\|x_B^{0^k} - x_B^*\|_*\} < 0.9\sigma$.

Statements (a) and (b) prove that for all $k > j$, $\|x_B^{0^k} - x_B^*\|_* < 0.9\sigma$, contradicting the fact that σ is an accumulation point of the sequence $(\|x_B^{0^k} - x_B^*\|_*)$, and completing the proof. □

Acknowledgments. The authors gratefully acknowledge Mike Todd for an extremely detailed reading of the manuscript and for numerous suggestions that led to improvements in the presentation. They also thank a second anonymous referee and Hector Martinez for comments on the manuscript. Finally, they thank Florian Potra and Yinyu Ye for active discussions following an initial presentation of this material at the 4th SIAM Meeting on Optimization in Chicago, Illinois, May 1992.

REFERENCES

[1] I. ADLER AND R. D. C. MONTEIRO, *Limiting behavior of the affine scaling continuous trajectories for linear programming problems*, Math. Programming, 50 (1991), pp. 29–51.

[2] A. CHARNES, W. W. COOPER, AND R. M. THRALL, *A structure for classifying and characterizing efficiency and inefficiency in data envelopment analysis*, J. Productivity Anal., 2 (1991), pp. 197–237.

[3] C. Gonzaga, *Path-following methods for linear programming*, SIAM Rev., 34 (1992), pp. 167–224.

[4] W. W. Hogan, *Point-to-set maps in mathematical programming*, SIAM Rev., 15 (1973), pp. 591–603.

[5] P. Huard, *Point-to-set maps in mathematical programming*, Math. Programming Study 10, North–Holland Publishing, Amsterdam, 1979.

[6] M. Kojima, S. Mizuno, and A. Yoshise, *A primal-dual interior-point method for linear programming*, in Progress in Mathematical Programming, Interior-Point and Related Methods, Nimrod Megiddo, ed., Springer-Verlag, Berlin, New York, 1989, pp. 29–47.

[7] M. Kojima, S. Mizuno, and A. Yoshise, *An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems*, Math. Programming, 90 (1991), pp. 331–342.

[8] I. J. Lustig, R. E. 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.

[9] L. McLinden, *An analogue of Moreau's proximation theorem, with application to the nonlinear complementarity problem*, Pacific J. Math., 88 (1980), pp. 101–161.

[10] K. McShane, *Superlinearly convergent $O(\sqrt{n}L)$-iteration interior-point algorithm for linear programming and the monotone linear complementarity problem*, SIAM J. Optim., 4 (1994), pp. 247–261.

[11] N. Megiddo, *Pathways to the optimal set in linear programming*, in Progress in Mathematical Programming, Interior-Point and Related Methods, Nimrod Megiddo, ed., Springer-Verlag, Berlin, New York, 1989, pp. 131–158.

[12] N. Megiddo and M. Shub, *Boundary behavior of interior point algorithms in linear programming*, Math. Oper. Res., 14 (1989), pp. 97–146.

[13] S. Mehrotra, *Quadratic convergence in a primal-dual method*, Math. Oper. Res., 18 (1993), pp. 741–751.

[14] 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.

[15] R. C. Monteiro and I. Adler, *Interior path-following primal-dual algorithm. Part I: Linear programming*, Math. Programming, 44 (1989), pp. 27–41.

[16] G. Sonnevend, *An analytical centre for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming*, in Lecture Notes in Control and Information Sciences 84, Springer-Verlag, Berlin, New York, 1985, pp. 866–876.

[17] R. A. Tapia, Y. Zhang, and Y. Ye, *On the convergence of the iteration sequence in primal-dual interior point methods*, Math. Programming, 68 (1995), pp. 141–154.

[18] M. J. Todd and Y. Ye, *A centered projective algorithm for linear programming*, Math. Oper. Res., 15 (1990), pp. 508–529.

[19] Y. Ye, *On the Q-order of convergence of interior-point algorithms for linear programming*, in Proc. 1992 Symposium on Applied Mathematics, Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing, China, 1992, pp. 534–543.

[20] Y. Ye, O. Güler, R. A. Tapia, and Y. Zhang, *A quadratically convergent $O(\sqrt{n}L)$-iteration algorithm for linear programming*, Math. Programming, 59 (1993), pp. 151–162.

[21] Y. Ye, R. A. Tapia, and Y. Zhang, *A Superlinearly Convergent $O(\sqrt{n}L)$-Iteration Algorithm for Linear Programming*, Technical report TR91-22, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1991.

[22] Y. Zhang and A. El-Bakry, *A Modified Predictor-Corrector Algorithm for Locating Weighted Centers in Linear Programming*, Technical report TR92-19, Dept. of Computational and Applied Mathematics, Rice University, Houston, TX, 1992.

[23] Y. Zhang and R. A. Tapia, *A superlinearly convergent polynomial primal-dual interior-point algorithm for linear programming*, SIAM J. Optim., 3 (1993), pp. 118–133.

[24] Y. Zhang and R. A. Tapia, *Superlinear and quadratic convergence of primal-dual interior-point methods for linear programming revisited*, J. Optim. Theory Appl., 73 (1992), pp. 229–242.

[25] Y. Zhang and R. A. Tapia, *On the Convergence of Interior-Point Methods to the Center of the Solution Set in Linear Programming*, Technical report TR91-30, Dept. of Computational and Applied Mathematics, Rice University, Houston, TX, 1991.

[26] Y. Zhang, R. A. Tapia, and J. E. Dennis, *On the superlinear and quadratic convergence of primal-dual interior-point linear programming algorithms*, SIAM J. Optim., 2 (1992), pp. 304–324.

[27] Y. Zhang, R. Tapia, and F. Potra, *On the superlinear convergence of interior-point algorithms for a general class of problems*, SIAM J. Optim., 3 (1993), pp. 413–422.

SIAM J. OPTIM.
Vol. 7, No. 1, pp. 66–85, February 1997

ON THE QUADRATIC CONVERGENCE OF THE SIMPLIFIED MIZUNO–TODD–YE ALGORITHM FOR LINEAR PROGRAMMING[*]

CLOVIS C. GONZAGA[†] AND RICHARD A. TAPIA[‡]

Abstract. It is known that the Mizuno–Todd–Ye predictor-corrector primal-dual Newton interior-point method generates a duality-gap sequence which converges quadratically to zero, and this is accomplished with an iteration complexity of $O(\sqrt{n}L)$. Very recently, the present authors demonstrated that the iteration sequence generated by this method converges, and this convergence is to the analytic center of the solution set. In the current work we show that within a finite number of iterations, the Newton corrector step can be replaced with a simplified Newton corrector step, and the resulting algorithm maintains $O(\sqrt{n}L)$ iteration complexity, quadratic convergence of the duality-gap sequence to zero, and convergence of the iteration sequence (however, not necessarily to the analytic center). The simplified predictor-corrector algorithm requires only one linear solve per iteration in contrast to the two linear solves per iteration required by the original predictor-corrector algorithm.

Key words. linear programming, primal-dual interior-point algorithm, predictor-corrector algorithm, quadratic convergence

AMS subject classifications. 49M, 65K, 90C

PII. S1052623493243569

1. Introduction and preliminaries.

The basic primal-dual interior-point method for linear programming was originally proposed by Kojima, Mizuno, and Yoshise [4] based on earlier work of Megiddo [8]. This method can be viewed as a perturbed and damped Newton's method applied to the first-order conditions for a particular standard form linear program. They established linear convergence of the duality-gap sequence to zero and an iteration complexity of $O(nL)$ for their basic algorithm. Immediately Kojima, Mizuno, and Yoshise in a second paper [5] and Monteiro and Adler [12] proposed algorithms that fit in the original Kojima–Mizuno–Yoshise framework and established linear convergence of the duality gap sequence to zero and a superior iteration complexity of $O(\sqrt{n}L)$ for their versions of the algorithm. Soon after Mizuno, Todd, and Ye [11] considered a predictor-corrector variant of the Kojima–Mizuno–Yoshise basic algorithm. In their algorithm, the predictor step is a damped Newton step and the corrector step is a perturbed (centered) Newton step. Hence one iteration of the predictor-corrector algorithm requires the solution of two linear systems, essentially two Newton steps. Hence when comparing convergence rate results, they should technically be considered two-step results. Mizuno, Todd, and Ye established linear convergence for their predictor-corrector algorithm and a superior iteration complexity bound of $O(\sqrt{n}L)$.

[*] Received by the editors January 27, 1993; accepted for publication (in revised form) August 3, 1995. This research was initiated in February 1992 while the first author was visiting the Center for Research on Parallel Computation, Rice University. A preliminary version of this paper was presented at the 4th SIAM Meeting on Optimization, Chicago, IL, May 1992.

http://www.siam.org/journals/siopt/7-1/24356.html

[†] Department of Systems Engineering and Computer Science, COPPE-Federal University of Rio de Janeiro, Cx. Postal 68511, 21945-970 Rio de Janiero, RJ Brazil (clovis@mtm.ufsc.br). Part of this work was done while the author was visiting the Cornell Center for Applied Mathematics, with support from NSF, AFOSR, and ONR through NSF grant DMS-8920550 and from the Center for Applied Mathematics.

[‡] Department of Computational and Applied Mathematics and Center for Research on Parallel Computation, Rice University, Houston, TX 77251-1892 (tapia@masc8.rice.edu). The research of this author was supported in part by NSF cooperative agreements CCR-8809615, AFOSR 89-0363, DOE DEFG05-86ER25017, and ARO 9DAAL03-90-G-0093.

We now briefly give a chronological account of the development of fast (superlinear) convergence for these primal-dual interior-point methods. We refer to the Kojima–Mizuno–Yoshise method as the basic method and to the Mizuno–Todd–Ye method as the predictor-corrector method. When we discuss convergence or convergence attributes of one of these methods, we are describing the convergence of the duality-gap to zero. This interpretation has become standard in this area, even though convergence of the duality-gap sequence does not imply convergence of the iteration sequence. The convergence of the iteration sequence is certainly an important issue in its own right and to some extent has been neglected. For an interesting result concerning the convergence of the iteration sequence generated by the basic method, see Tapia, Zhang, and Ye [13]. For a definitive result concerning the convergence of the iteration sequence for the predictor-corrector method, see Gonzaga and Tapia [3].

Zhang, Tapia, and Dennis [20] demonstrated that under certain assumptions the algorithmic parameters in the basic method could be chosen so that superlinear convergence was obtained for degenerate problems and quadratic convergence was obtained for nondegenerate problems. However, they did not demonstrate that polynomial complexity would be retained. Zhang and Tapia [19] demonstrated that the algorithmic parameters in the basic algorithm could be chosen so that the polynomial complexity bound was maintained and superlinear convergence was obtained for degenerate problems, while quadratic convergence was obtained for nondegenerate problems. Ye, Tapia, and Zhang [17] demonstrated that the predictor-corrector algorithm was superlinearly convergent for degenerate problems and quadratically convergent for nondegenerate problems while maintaining its $O(\sqrt{n}L)$ iteration complexity. McShane [7] independently obtained a similar result. Up to this point all superlinear convergence results assumed that the iteration sequence converged. Ye et al. [16] and independently Mehrotra [10], based on Ye, Tapia, and Zhang [17], demonstrated the surprising result that neither the nondegeneracy assumption nor the assumption of iteration sequence convergence was needed for the quadratic convergence of the predictor-corrector algorithm.

In this paper we add to the literature on the predictor-corrector algorithm by demonstrating that its quadratic convergence and $O(\sqrt{n}L)$ complexity are retained if one replaces the Newton corrector step with a simplified Newton step; i.e., the Jacobian from the Newton predictor step is used also in the computation of the corrector step. Hence the corrector step only requires a back-solve, and the complete iteration only requires the solution of one linear system. Actually, the Newton corrector step cannot be replaced with a simplified Newton corrector step at the beginning of the iterative process, but only after a particular criterion is satisfied. We demonstrate that this criterion will be satisfied within a finite number of iterations. We also show that the simplified algorithm generates an iteration sequence which is convergent, but not necessarily to the analytic center.

Recently Ye [15] was able to show that a variant of the Mizuno–Todd–Ye predictor-corrector algorithm could be given that eventually did not require the corrector step. He demonstrated that this variant algorithm gave subquadratic convergence (the Q-rate is two, but the Q_2-factor may be unbounded). Hence Ye attains a convergence rate of two with an algorithm which (eventually) only requires one linear solve per iteration. Our simplified Mizuno–Todd–Ye algorithm gives Q-quadratic convergence but requires the solution of one linear system and an additional back-solve per iteration. It should be clear that any convergence rate analysis based on total number of arithmetic operations per iteration will favor the Ye variant. It should also be clear that numerical efficiency of an algorithm is determined by the effective number of iterations needed for numerical convergence and not convergence rate alone.

The paper is organized as follows. In the remainder of this section we introduce our notation and several fundamental background notions. In section 2 we discuss the primal-dual Newton step and the primal-dual simplified Newton step and derive several properties concerning these two steps. Some results on scaled projections from Gonzaga and Tapia will be collected in section 3. These results will be used in section 5. The Mizuno–Todd–Ye predictor-corrector algorithm is presented in section 4. Section 5 begins with the presentation of the simplified predictor-corrector algorithm and then turns to establishing our convergence theory for the simplified predictor-corrector algorithm. In section 6 we make some observations that imply that quadratic convergence is optimal for both the predictor-corrector method and its simplified variant. We indicate that cubic convergence might be obtained by appropriately modifying the corrector step.

Given a vector x, d, ϕ, the corresponding upper case symbol denotes (as usual) the diagonal matrix X, D, Φ defined by the vector.

We denote component-wise operations on vectors by the usual notations for real numbers. Thus, given two vectors u, v of the same dimension, uv, u/v, etc. denote the vectors with components $u_i v_i$, u_i/v_i, etc. This notation is consistent as long as component-wise operations are given precedence over matrix operations. Note that $uv \equiv Uv$, and if A is a matrix then $Auv \equiv AUv$, but in general $Auv \neq (Au)v$.

We frequently use the $O(\cdot)$ and $\Omega(\cdot)$ notations to express a relationship between functions. Our most common usage will be associated with a sequence $\{x^k\}$ of vectors and a sequence $\{\mu^k\}$ of positive real numbers. In this case $x = O(\mu)$ or $x^k = O(\mu^k)$ means that there is a constant K (dependent on problem data) such that for every $k \in \mathbb{N}$, $\|x^k\| \leq K\mu^k$. Similarly, $x = \Omega(\mu)$ or $x^k = \Omega(\mu^k)$ means that there is $\epsilon > 0$ such that for every $k \in \mathbb{N}$, $\|x^k\| \geq \epsilon\mu^k$. Given a matrix A, $\mathcal{N}(A)$ and $\mathcal{R}(A)$ denote, respectively, its null space and range space. P_A denotes the projection matrix into $\mathcal{N}(A)$, and $\tilde{P}_A = I - P_A$.

The primal and dual linear programming problems are as follows:

$$(LP) \qquad \begin{array}{rrcl} \text{minimize} & c^T x & & \\ \text{subject to} & Ax & = & b, \\ & x & \geq & 0 \end{array}$$

and

$$(LD) \qquad \begin{array}{rrcl} \text{maximize} & b^T y & & \\ \text{subject to} & A^T y + s & = & c, \\ & s & \geq & 0, \end{array}$$

where $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and $A \in \mathbb{R}^{m\times n}$. We assume that both problems have optimal solutions and that the sets of optimal solutions are bounded. This is equivalent to the requirement that both feasible sets contain points satisfying all inequality constraints strictly.

Given any feasible primal-dual pair $(\tilde{x}, \tilde{s})$, the problems can be rewritten as

$$(LP) \qquad \begin{array}{rrcl} \text{minimize} & \tilde{s}^T x & & \\ \text{subject to} & Ax & = & b, \\ & x & \geq & 0 \end{array}$$

and

$$(LD) \qquad \begin{array}{rrcl} \text{minimize} & \tilde{x}^T s & & \\ \text{subject to} & Bs & = & Bc, \\ & s & \geq & 0, \end{array}$$

where B^T is a matrix whose columns span the null space of A. Popular choices for B^T are an orthonormal basis for the null space of A and $B = P_A$, the projection matrix into the null space of A.

The feasible sets for (LP) and (LD) will be denoted, respectively, by $\mathcal{P}$ and $\mathcal{D}$. Their relative interiors will be, respectively, $\mathcal{P}^0$ and $\mathcal{D}^0$.

The set of optimal solutions for the primal-dual pair of problems constitutes a face $F = F_P \times F_D$ of the polyhedron of feasible solutions, where F_P and F_D are, respectively, the primal and dual optimal faces. By hypothesis, this face is a compact set. It is well known that this face is characterized by a partition $\{B, N\}$ of the set of indices $\{1, \ldots, n\}$ such that $F_P = \{x \in \mathcal{P} \mid x_N = 0\}$ and $F_D = \{s \in \mathcal{D} \mid s_B = 0\}$. In the relative interior of the face F, $x_B > 0$ and $s_N > 0$.

We study algorithms that converge to the optimal face. Our main concern is with the behavior of the iterates as they approach the optimal face. We want this to happen in such a manner that all limit points are in the relative interior of the optimal face. We shall see later on how this condition can be enforced by requiring some adherence to the central path. For detail on the central path, see Gonzaga [2].

Given $\mu > 0$, $\mu \in \mathbb{R}$, the pair (x, s) of feasible primal and dual solutions is the central point $(x(\mu), s(\mu))$ associated with μ if

$$xs = \mu e,$$

where e stands for the vector of all ones, with dimension given by the context.

The central path is the curve in $\mathbb{R}^{2n}$ parametrized by the positive real μ, i.e.,

$$\mu \mapsto (x(\mu), s(\mu)).$$

Thus (x, s) is a central point if and only if

$$\begin{array}{rcl} xs & = & \mu e, \\ Ax & = & b, \\ Bs & = & Bc, \\ x, s & \geq & 0, \end{array} \tag{1}$$

where the columns of B^T span the null space of A.

The first-order or Karush–Kuhn–Tucker (KKT) conditions for problem (LP) (or (LD)) are

$$\begin{array}{rcl} xs & = & 0, \\ Ax & = & b, \\ A^T y + s & = & c, \\ x, s & \geq & 0. \end{array}$$

The perturbed KKT conditions for perturbation parameter $\mu > 0$ are

$$\begin{array}{rcl} xs & = & \mu e, \\ Ax & = & b, \\ A^T y + s & = & c, \\ x, s & \geq & 0. \end{array} \tag{2}$$

Observe that the perturbed KKT conditions are merely the defining relations for the central path and (2) can equivalently be written as (1). Essentially all primal-dual interior-point methods for problem (LP) consist of some variant of the damped Newton method applied to the perturbed KKT conditions (1) or (2).

2. The Newton and simplified Newton steps. When dealing with an iterative procedure we will use the superscript 0 to denote the previous iterate, and no superscript to denote the current iterate, and a superscript of + to denote the subsequent iterate. In two-step algorithms like the Mizuno–Todd–Ye (MTY) algorithm described in section 4, this notation will apply to the current iterate, the intermediate iterate, and the final iterate.

Suppose that (x^0, s^0) and (x, s) have been obtained from a form of Newton's method and are both feasible pairs. The Newton step (or correction) for (1) at (x, s) is given by (u, v), the solution of

$$\begin{array}{rcl} xv + su &=& -xs + \mu e, \\ Au &=& 0, \\ Bv &=& 0, \end{array} \tag{3}$$

and the simplified Newton step for (1) at (x, s) is given by (u, v), the solution of

$$\begin{array}{rcl} x^0 v + s^0 u &=& -xs + \mu e, \\ Au &=& 0, \\ Bv &=& 0. \end{array} \tag{4}$$

It should be clear that the difference between (3) and (4) is that (3) uses the Jacobian of (1) at (x, s) and (4) uses the Jacobian of (1) at (x^0, s^0).

We introduce some additional notation that will be used throughout the paper. Given a pair (x, s), we define

$$\begin{array}{rcl} \mu(x,s) &=& x^T s/n, \\ w(x,s) &=& xs/\mu(x,s), \\ \delta(x,s) &=& \|w(x,s) - e\|, \\ \phi(x,s) &=& (\sqrt{w(x,s)})^{-1}. \end{array} \tag{5}$$

When no confusion can arise, we drop the reference to the variables and continue to use other symbols in a consistent manner. For instance, given a pair $(\bar{x}, \bar{s})$, the parameters above will be denoted simply $\bar{\mu}, \bar{w}$, and $\bar{\phi}$.

Given a pair (x, s), $\mu(x, s)$ is the penalty parameter associated with (x, s) in the following sense: if (x, s) is a central point, then $xs = \mu(x, s)e$; otherwise, $\mu(x, s)$ is the penalty parameter associated with the central point that is nearest the pair (x, s) in terms of a certain proximity measure. The vector w consists of logarithmic barrier weights associated with (x, s). It characterizes the weighted primal-dual affine scaling trajectory through (x, s), as studied by Monteiro and Adler [12]. The scalar δ is a measure of proximity from (x, s) to the central point $(x(\mu), s(\mu))$. The definition of ϕ was made merely for convenience; it will simplify expressions below.

At this point we are interested in obtaining usable closed form solutions for the simplified Newton step and the Newton step. We also derive an interesting property of the simplified Newton step. In what follows it is important not to confuse μ in (3) and (4) with $\mu(x, s)$ given in (5), because they are not necessarily the same. Hence μ denotes the μ in (3) and (4) and $\mu(x, s)$ means the $\mu(x, s)$ given in (5). Since no confusion will arise in the case of μ^0, we use μ^0 to denote $\mu(x^0, s^0)$.

Proposition 2.1. *The simplified Newton step* (u, v) *given by* (4) *can be written*

$$\begin{array}{rcl} u &=& x^0\phi^0 P_{AX^0\Phi^0}\phi^0 \left(-\dfrac{xs}{\mu^0} + \dfrac{\mu}{\mu^0}e\right), \\[2ex] v &=& s^0\phi^0 \tilde{P}_{AX^0\Phi^0}\phi^0 \left(-\dfrac{xs}{\mu^0} + \dfrac{\mu}{\mu^0}e\right), \end{array} \tag{6}$$

where $\tilde{P} = I - P$.

Proof. Assume that instead of (4), the simplified Newton equations are written as

$$x^0 v + s^0 u = -xs + \mu e, \quad u \in \mathcal{N}(A), \quad v \in \mathcal{R}(A^T). \tag{7}$$

The solution is obtained by associating a scaling vector

$$d(x,s) = \sqrt{\frac{x}{s}}$$

to each pair (x,s).

Using the definitions in (5) and dropping argument references when no confusion will arise,

$$d = \sqrt{\frac{x}{s}} \;=\; \frac{x\phi}{\sqrt{\mu(x,s)}} \;=\; \frac{\sqrt{\mu(x,s)}}{\phi s}\;. \tag{8}$$

The solution of (7) is obtained by scaling the problems by $\bar{x} = (d^0)^{-1}x$, $\bar{s} = d^0 s$:

$$\begin{array}{rcl} \bar{x}^0\bar{v} + \bar{s}^0\bar{u} &=& -\bar{x}\bar{s} + \mu e, \\ \bar{u} &\in& \mathcal{N}(AD^0), \\ \bar{v} &\in& \mathcal{R}(D^0A^T). \end{array}$$

The choice of this scaling becomes clear when we notice that by direct substitution,

$$\bar{x}^0 = \bar{s}^0 = \sqrt{x^0 s^0}. \tag{9}$$

Dividing the equation by $\bar{s}^0$ and using the definitions of scaled variables,

$$\bar{u} + \bar{v} = -\frac{\bar{x}}{\bar{x}^0}\bar{s} + \mu(\bar{x}^0)^{-1} = \frac{d^0}{x^0}(-xs + \mu e).$$

Hence $\bar{u}$ and $\bar{v}$ are the components of the right-hand side in the complementary subspaces, the null space, and row space of AD^0; they are given by

$$\begin{array}{rcl} \bar{u} &=& P_{AD^0}\dfrac{d^0}{x^0}(-xs + \mu e), \\ \bar{v} &=& \tilde{P}_{AD^0}\dfrac{d^0}{x^0}(-xs + \mu e), \end{array} \tag{10}$$

where $\tilde{P}_{AD^0} = I - P_{AD^0}$. Finally, $u = d^0\bar{u}$ and $v = (d^0)^{-1}\bar{v}$.

A convenient formulation is obtained by substituting $d^0 = \frac{1}{\sqrt{\mu^0}}x^0\phi^0$ and $(d^0)^{-1} = \frac{1}{\sqrt{\mu^0}}s^0\phi^0$, and this leads to (6). □

The simplified Newton step and the Newton step satisfy an interesting property. This property will turn out to be fundamental to the analysis presented in section 5. Hence we derive this property in a form which covers both the simplified Newton step and the Newton step.

Proposition 2.2. *Let* $(\hat{x}, \hat{s})$ *and* (x, s) *be feasible pairs*; $\pi \in [0, 1]$. *Consider* $x^+ = x + u$ *and* $s^+ = s + v$, *where* (u, v) *satisfies*

$$\begin{aligned} \hat{x}v + \hat{s}u &= -(1-\pi)xs + \hat{\mu}e, \\ u &\in \mathcal{N}(A), \\ v &\in \mathcal{R}(A^T). \end{aligned}$$

Then

$$\mu(x^+, v^+) = \pi\mu(x, s) + \hat{\mu}. \tag{11}$$

Proof. Left multiplying by e^T, we obtain

$$\hat{x}^T v + \hat{s}^T u = -(1-\pi)x^T s + n\hat{\mu}$$

from the definition

$$x^{+^T} s^+ = x^T s + x^T v + s^T u,$$

since $u^T v = 0$. But $\hat{x}^T v = x^T v$, because $\hat{x} - x \in \mathcal{N}(A)$ and $v \in \mathcal{R}(A^T)$ and, similarly, $\hat{s}^T u = s^T u$. Substituting in the expressions above we immediately obtain (11). □

3. Scaled projections. In this section we collect some results on scaled projections from Gonzaga and Tapia [3]. These results are extensions of results published by Megiddo and Shub [9]. We use $\mathbb{R}_+$ to denote the nonnegative reals and $\mathbb{R}_{++}$ to denote the positive reals.

Consider the primal feasible set for (LP)

$$\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$$

and the map h defined for $(d, \rho) \in \mathcal{J} = (\mathbb{R}^n{}_+ \backslash \{0\}) \times \mathbb{R}^n$,

$$(d, \rho) \in \mathcal{J} \mapsto h(d, \rho) = P_{AD}\rho, \tag{12}$$

where P_{AD} represents the projection matrix into the null space of AD.

We study the behavior of this map when $d > 0, (d, \rho) \to (\bar{d}, \bar{\rho}) \in \mathcal{J}$.

Given $\bar{d}$, we define the index sets $B = \{i = 1, \ldots, n \mid \bar{d}_i > 0\}$ and $N = \{i = 1, \ldots, n \mid \bar{d}_i = 0\}$. The variables with indices in B are called the large variables, and the others small variables. It is difficult to describe the behavior of the small variables $h_N(d, \rho)$ of the scaled projection defined above; the theory of Megiddo and Shub concerns the large variables $h_B(d, \rho)$. We shall describe these results conveniently extended to fit our needs. The following proposition is Lemma 3.2 of Gonzaga and Tapia [3]. We refer the reader to that paper for the proof.

Proposition 3.1. *Let* $h(d, \rho)$ *be given by* (12). *Consider* $(\bar{d}, \bar{\rho}) \in \mathcal{J}$ *and* $(d^k, \rho^k) \in \mathbb{R}^n{}_+ \times \mathbb{R}^n$ *such that* $(d^k, \rho^k) \to (\bar{d}, \bar{\rho})$. *Then*

(i) $h_B(d^k, \rho^k) \to h_B(\bar{d}, \bar{\rho}) = P_{A_B \bar{D}_B} \bar{\rho}_B$;

(ii) *if* $\bar{\rho}_N = 0$, *then* $h_N(d^k, \rho^k) \to 0$.

Consider compact sets $\Gamma \subset \mathbb{R}^n$ and $\Delta \subset \mathbb{R}^n{}_+$ such that for any $d \in \Delta$, $d_B > 0$ and $d_N = 0$, where $\{B, N\}$ is a partition of $\{1, \ldots, n\}$. We now extend the proposition above for the case of sequences $\{d^k\}$ in $\mathbb{R}^n{}_{++}$ and $\{\rho^k\} \in \mathbb{R}^n$ such that $d^k \to \Delta$ and $\rho^k \to \Gamma$. [1]

[1] A sequence $\{z^k\}$ converges to a set Z if $d(z^k, Z) \to 0$, where $d(z^k, Z) = \inf_{z \in Z} \|z^k - z\|$.

PROPOSITION 3.2. *For the situation described above we have the following:*

(i) *if* $d^k \to \Delta$ *and* $\rho^k \to \Gamma$, *then*

$$h_B(d^k, \rho^k) - P_{A_B D_B^k} \rho_B^k \to 0;$$

(ii) *if* $d^k \to \bar{d} \in \Delta$ *and* $\rho^k \to \bar{\rho} \in \Gamma$, *then*

$$h_B(d^k, \rho^k) - P_{A_B \bar{D}_B} \bar{\rho}_B \to 0.$$

Proof. Implication (ii) follows from (i), since for convergent sequences $P_{A_B D_B^k} \rho_B^k \to P_{A_B \bar{D}_B} \bar{\rho}_B$.

To prove (i), assume by contradiction that there exist $\epsilon > 0$ and sequences $\{d^k\}$ in $\mathbb{R}^n{}_{++}$ and $\{\rho^k\}$ in $\mathbb{R}^n$ such that for $k = 1, 2, \ldots$

$$\|h_B(d^k, \rho^k) - P_{A_B D_B^k} \rho_B^k\| > \epsilon. \tag{13}$$

Since the sequences $\{d^k\}$ and $\{\rho^k\}$ converge to compact sets they must be bounded. Hence they have accumulation points $\bar{d}, \bar{\rho}$, such that for some $\mathcal{K} \subset \mathbb{N}$, $d^k \xrightarrow{\mathcal{K}} \bar{d}$ and $\rho^k \xrightarrow{\mathcal{K}} \bar{\rho}$. From the facts that d^k converges to Δ and ρ^k converges to Γ and the compactness of Δ and Γ, $\bar{d} \in \Delta$ and $\bar{\rho} \in \Gamma$. From Proposition 3.1,

$$h_B(d^k, \rho^k) \xrightarrow{\mathcal{K}} P_{A_B \bar{D}_B} \bar{\rho}_B,$$

and since $\bar{D}_B > 0$,

$$P_{A_B D_B^k} \rho_B^k \xrightarrow{\mathcal{K}} P_{A_B \bar{D}_B} \bar{\rho}_B.$$

By subtracting these last expressions, we see that

$$h_B(d^k, \rho^k) - P_{A_B D_B^k} \rho^k \xrightarrow{\mathcal{K}} 0,$$

contradicting (13) and completing the proof. □

Now we present two facts related to projections and slightly shifted scalings.

PROPOSITION 3.3. *Let* $q \in \mathbb{R}^N$ *be such that* $\|q - e\|_\infty \leq \alpha$, $\alpha \in (0, 0.25)$, *and consider the projections* $\hat{h} = P_A \rho$, $h = q P_{AQ} q \rho$. *Then* $\|h - \hat{h}\| \leq 3\alpha \|\hat{h}\|$.

Proof. See [3]. □

Given a vector $x \in \mathbb{R}^n{}_{++}$, the following map defines a norm

$$h \in \mathbb{R}^n \mapsto \|h\|_x = \|x^{-1} h\|.$$

This is the Euclidean norm of the vector corresponding to h after a scaling $\bar{h} = x^{-1} h$. This norm is very usual in interior-point methods.

The following result shows that all scaled norms for x in a compact set in the interior of the positive orthant are uniformly equivalent.

PROPOSITION 3.4. *Let* $\Delta \subset \mathbb{R}^n{}_{++}$ *be a compact set. Then there is a number* $\Gamma > 0$ *such that for any* $h \in \mathbb{R}^n$, $x \in \Delta$,

$$\frac{1}{\Gamma} \|h\| \leq \|h\|_x \leq \Gamma \|h\|.$$

Proof. By definition, given $x \in \Delta$, $\|h\|_x = \|x^{-1} h\|$. We immediately obtain

$$\min_{i=1,\ldots,n} x_i^{-1} \|h\| \leq \|h\|_x \leq \max_{i=1,\ldots,n} x_i^{-1} \|h\|.$$

Since x_i , $i = 1, \ldots, n$, are bounded and bounded away from zero for $x \in \Delta$, the scalar Γ must exist, completing the proof. □

4. The Mizuno–Todd–Ye algorithm. The MTY algorithm is a path-following predictor-corrector algorithm. All activity is restricted to a region near the central path; i.e., all points (x, s) generated by the algorithm satisfy

$$\delta(x,s) = \|w(x,s) - e\| = \|\frac{xs}{\mu(x,s)} - e\| \leq \alpha,$$

where $\alpha \in (0, 0.5)$.

We shall describe a typical iteration of the algorithm and list its properties. Complete proofs can be found in Mizuno, Todd, and Ye [11].

Given $\alpha = 0.1$,[1] a typical iteration begins with feasible (x^0, s^0) such that $\delta(x^0, s^0) = \|w^0 - e\| \leq \alpha^2/\sqrt{2}$.

Predictor step. Given (x^0, s^0), compute the (affine-scaling) step (u^0, v^0) and let $x = x^0 + u^0,\ s = s^0 + v^0$, where (u^0, v^0) is defined by

$$x^0v^0 + s^0u^0 = -(1-\gamma)x^0s^0, \ \ u^0 \in \mathcal{N}(A), \ \ v^0 \in \mathcal{R}(A^T),$$

with $\gamma \in [0, 1)$ such that $\delta(x, s) = \|w(x, s) - e\| \leq \alpha$. (The specific choice of γ will be discussed below.)

Corrector step. Given (x, s), compute the (centering) step (u, v) and let $x^+ = x + u$, $s^+ = s + v$, where (u, v) is defined by

$$xv + su = -xs + \mu e, \quad u \in \mathcal{N}(A), v \in \mathcal{R}(A^T),$$

with $\mu = \mu(x, s)$.

Observe that our γ in the predictor step is effectively a steplength parameter. To see this let us denote the predictor step by $(u^0(\gamma), v^0(\gamma))$ and let $\theta = 1 - \gamma$. Then

$$\theta(u^0(0), v^0(0)) = (u^0(\gamma), v^0(\gamma))$$

and

$$(x, s) = (x^0, s^0) + \theta(u^0(0), v^0(0)),$$

which is the usual way of writing the MTY predictor step. The usual choice for θ is θ^k, the largest $\theta \in (0, 1]$ such that $\delta(x(\theta), s(\theta)) \leq \alpha$ for all $0 \leq \theta \leq \theta^k$. For further details, see, for example, section 2 of Ye et al. [16]. Hence our choice for γ in the predictor step is $\gamma = 1 - \theta^k$ and can be viewed as the smallest $\gamma \in [0, 1)$ in the sense just described.

Mizuno, Todd, and Ye [11] prove that the algorithm is well defined in the sense that the centering step produces (x^+, s^+) such that $\delta(x^+, s^+) \leq \alpha^2/\sqrt{2}$. Ye et al. [16] (and independently Mehrotra [10]) prove that the duality-gap (or, equivalently, the parameter μ) converges to zero Q-quadratically; i.e.,

$$\mu^+ = \mu(x^+, s^+) = O({\mu^0}^2).$$

Using Proposition 2.2 with $(\hat{x}, \hat{s}) = (x^0, s^0)$, $\pi = \gamma$, and $\hat{\mu} = 0$, we see that for the predictor step,

$$\mu(x, s) = \gamma\mu(x^0, s^0).$$

[1] The original paper uses $\alpha = 0.5$. We shall use a convenient value of 0.1, since this simplifies some formulas ahead.

Using Proposition 2.2 with $(\hat{x}, \hat{s}) = (x, s)$, $\pi = 0$, and $\hat{\mu} = \gamma\mu(x^0, s^0)$, we see that for the corrector step,

$$\mu(x^+, s^+) = \gamma\mu(x^0, s^0).$$

So, on one hand we have $\mu^+ = O(\mu^{0^2})$ and on the other hand we have $\mu^+ = \gamma\mu^0$. It follows that

$$\gamma = O(\mu^0).$$

Bounds on the quantities appearing in the algorithm are given in the propositions below. Let $\{B, N\}$ be the optimal partition for the linear programming problem, i.e., the index partition associated to the optimal face. It is well known (see Adler and Monteiro [1]) that the central path ends at the analytic center of the optimal face and that the pairs (x, s) such that $\|w(x, s) - e\| \leq \alpha$ constitute a neighborhood of the central path corresponding to the bundle of w-weighted affine-scaling trajectories for w such that $\|w - e\| \leq \alpha$. For α small, the bundle of trajectories ends in a compact neighborhood of the analytic center of the optimal face, contained in the relative interior of the face. Namely, the end points in the primal optimal face are the w-weighted centers given by

$$x^*(w) = \operatorname{argmin} \left\{ -\sum_{i \in B} w_i \log x_i \mid x \in F_P \right\}.$$

Hence, the algorithm behaves as follows. As the optimal face is approached (and this happens in polynomial time), $x_N^k \to 0$, $s_B^k \to 0$ and x_B^k, s_N^k remain in small neighborhoods of x_B^* and s_N^*, the analytic centers of the primal and dual optimal faces.

Actually, it is always true that $x^k \to x^*$, $s^k \to s^*$, due to the results proved in Gonzaga and Tapia [3], which we describe.

As was stressed in the beginning of section 6 of Gonzaga and Tapia [3], it is important to realize that our estimates do not require $(x^{0^{k+1}}, s^{0^{k+1}})$ to be related to (x^{+^k}, s^{+^k}); i.e., (x^{0^k}, s^{0^k}) does not have to be generated by the MTY algorithm. All that is required is that (x^{0^k}, s^{0^k}) satisfy the condition $w\|(x^{0^k}, s^{0^k}) - e\| \leq \alpha$ for the appropriate choice of α. Hence, in what follows in this section and in section 5, we employ this broad interpretation when discussing quantities generated by the MTY algorithm or the simplified MTY algorithm for only one iteration.

PROPOSITION 4.1. *Consider quantities generated by the MTY algorithm. Then*

(i) $x_N = O(\mu)$, $s_B = O(\mu)$, $x_N^0 = O(\mu^0)$, $s_B^0 = O(\mu^0)$,

(ii) $u^0 = O(\mu^0)$, $v^0 = O(\mu^0)$,

(iii) $u_N = O(\mu)$, $v_B = O(\mu)$.

Proof. See Lemma 5.1 of [3]. □

The proposition above shows that the variations in (x, s) due to either an MTY predictor or corrector step are bounded by $O(\mu^0)$, with the exception of u_B and v_N. These are the variations in the large variables due to the corrector step.

The following proposition is the main result in Gonzaga and Tapia [3]. It is related to the map that associates to a pair (x^0, s^0) the pair (x^+, s^+) resulting from an MTY iteration. The proposition says that near the optimal face, an MTY iteration causes the large variables to approach the large variables of the analytic center (x^*, s^*) of the optimal face. The proposition describes only the behavior of the primal variables;

the dual variables behave in a similar fashion, due to the symmetry of the optimality conditions (1).

The approach to the center is measured in the norm relative to x^*_B, defined for $h\in \mathbb{R}^n$ by $\|h_B\|_* = \|(x^*_B)^{-1}h_B\|$.

PROPOSITION 4.2. *Consider a sequence (x^{0^k}, s^{0^k}) of primal-dual pairs (not necessarily generated by the algorithm) such that $\delta(x^{0^k}, s^{0^k}) \leq 0.1$ and $\mu^{0^k} \to 0$. Then there exists a sequence of positive reals $\{\epsilon^k\}$ such that $\epsilon^k \to 0$ and for sufficiently large k,*

$$\|x^{+^k}_B - x^*_B\|_* \leq \max\{\epsilon^k, 0.8\|x^{0^k}_B - x^*_B\|_*\}.$$

Proof. See Lemma 6.2 of [3]. □

This result implies that the iterates approach (x^*, s^*) and thus the sequence generated by the algorithm converges to the central optimum.

We are now concerned with bounding the sum of the variations (corrections) made to either the x-variable or the s-variable in either the predictor step or the corrector step in all iterations. The variation in x due to a predictor step is u^0. By the total variation in x due to predictor steps we mean $\sum_k \|u^{0^k}\|$. If we do not mention predictor steps or corrector steps we mean both steps. Analogous terminology is used for corresponding situations.

PROPOSITION 4.3. *Consider quantities x^{0^k}, s^{0^k}, x^k, etc. generated by the MTY algorithm starting at (x^{0^1}, s^{0^1}). Then*

(i) $\sum_{k=1}^{\infty} \mu^{0^k} = O(\mu^{0^1})$;

(ii) *the total variation in x_N and in s_B is bounded by $O(\mu^{0^1})$;*

(iii) *the total variation in x_B and in s_N due to predictor steps is bounded by $O(\mu^{0^1})$.*

Proof. To prove (i), it is enough to show that for some constant $\beta \in (0, 1)$, $\mu^{k+1} \leq \beta\mu^k$. This was shown by Mizuno, Todd, and Ye [11] when proving the polynomiality of the algorithm. Now (ii) and (iii) are direct consequences of Proposition 4.1, completing the proof. □

5. The simplified Mizuno–Todd–Ye algorithm. The simplified MTY algorithm is the MTY algorithm with the Newton corrector step replaced by a simplified Newton step. This means that the computation of the projections in (6) for the corrector step are reduced to a back substitution, instead of a complete solution of the system.

We now state the complete algorithm.

ALGORITHM 5.1. *Given $\alpha = 0.1$ and feasible (x^{0^1}, s^{0^1}) such that $\delta(x^{0^1}, s^{0^1}) \leq \alpha^2/\sqrt{2}$, set $k = 1$.*

```
    REPEAT
        $x^0 := x^{0^k}$, $s^0 := s^{0^k}$, $\mu^0 := \mu(x^0, s^0)$.
    Predictor: Given $(x^0, s^0)$ compute $(u^0, v^0)$, and let $x := x^0 + u^0$, $s := s^0 + v^0$
        where $(u^0, v^0)$ satisfies
        $x^0v^0 + s^0u^0 = -(1 - \gamma)x^0s^0$,    $u^0 \in \mathcal{N}(A)$, $v^0 \in \mathcal{R}(A^T)$,
        and $\gamma$ is as in the MTY predictor step.
    Simplified Corrector: Given $(x, s)$ set $\mu := \mu(x, s)$. Compute $(\hat{u}, \hat{v})$ satisfying
        $x^0\hat{v} + s^0\hat{u} = -xs + \mu e$,    $\hat{u} \in \mathcal{N}(A)$, $\hat{v} \in \mathcal{R}(A^T)$,
        and set $x^+ := x + \hat{u}$, $s^+ := s + \hat{v}$.
```

```
Safeguard: If $\delta(x^+, s^+) > \frac{\alpha}{2}$, then discard $(x^+, s^+)$ and compute the Newton
    corrector step
    $xv + su = -xs + \mu e, \quad u \in \mathcal{N}(A),\ v \in \mathcal{R}(A^T)$,
    and set $x^+ := x + u$, $s^+ := s + v$.
Subsequent iterate:
    ${x^0}^{k+1} := x^+$, ${s^0}^{k+1} := s^+$.
    $k := k + 1$.
UNTIL convergence.
```

Before we formally state the convergence properties that we have derived for the simplified predictor-corrector algorithm, there is value in collecting some fundamental observations. In what follows all quantities should be indexed by k; however, we will not always write the index k as we have been doing above.

PROPOSITION 5.2. *Let $\{(x^0, s^0)^k, (x, s)^k, (x^+, s^+)^k\}$ be generated by the simplified MTY predictor-corrector algorithm. Then*

(i) ${x^+}^T s^+ = x^T s$,

(ii) $x^T s = \gamma {x^0}^T s^0$,

(iii) $\gamma = O({x^0}^T s^0)$,

(iv) $x^T s \le (1 - \frac{\delta}{\sqrt{n}}) {x^0}^T s^0$ *for some* $\delta > 0$ *that does not depend on* k.

Proof. The proof of (i) follows from Proposition 2.2 with $(\hat{x}, \hat{s}) = (x, s)$, $\pi = 0$, and $\hat{\mu} = \mu(x, s)$. The proof of (ii) follows from Proposition 2.2 with $(\hat{x}, \hat{s}) = (x^0, s^0)$, $(x, s) = (x^0, s^0)$, $\pi = \gamma$, and $\hat{\mu} = 0$. Both (iii) and (iv) follow from Theorem 4.1 of Ye et al. [16], once we observe that their β is related to our α by the relationship $\beta = \frac{\alpha}{2}$ and their steplength θ is related to our γ by the relationship $\theta = 1 - \gamma$. □

The algorithm uses a simplified Newton iteration in the corrector step. If the simplified corrector produces the reduction in the proximity δ that ensures the quadratic convergence of the algorithm, i.e., if $\delta(x^+, s^+) \le \frac{\alpha}{2}$, then the step is accepted. Otherwise, the simplified step is discarded and the algorithm performs a Newton corrector step.

Two things must be proved: first, that the iterates are still convergent, not necessarily to the analytic center of the optimal face, and second, that the safeguard cannot be activated more than a finite number of times.

The predictor step is the same as that for the MTY algorithm. Our analysis will be based on a comparison of the simplified and exact corrector steps. The conclusions will be the following: for points near the optimal face,

(i) the simplified corrector step does not center the large variables. The variation in x_B and s_N due to simplified steps will be bounded by $O(\mu^0)$;

(ii) the behavior of the small variables x_N and s_B tends to be identical in both methods.

These two facts will be proved and then used to contradict the hypothesis that the safeguard is activated an infinite number of times.

We begin by studying the behavior of the large variables.

PROPOSITION 5.3. *Consider the corrector directions (u^k, v^k) and $(\hat{u}^k, \hat{v}^k)$ generated at iteration k of Algorithm* 5.1 *(independently of which one is actually accepted by the algorithm). Then there exist a number $K > 0$ and sequences $\{\theta_x^k\}$, $\{\theta_s^k\}$ in $\mathbb{R}_+$ such that $\theta_x^k \to 0$, $\theta_s^k \to 0$, and*

$$\|\hat{u}_B^k\| \le \gamma^k K(\|u_B^k\| + \theta_x^k),\ \|\hat{v}_N^k\| \le \gamma^k K(\|v_N^k\| + \theta_s^k).$$

Hence $\hat{u}_B = O(\mu^0)$ and $\hat{v}_N = O(\mu^0)$.

Proof. We shall prove the result for $\hat{u}_B^k$. The proof of the other result is similar.

Dropping the index k for notational simplicity, the primal directions are computed from (6):

$$\begin{aligned} \hat{u} &= x^0\phi^0 P_{AX^0\Phi^0}\phi^0 \left(-\frac{xs}{\mu^0}+\frac{\mu}{\mu^0}e\right), \\ u &= x\phi P_{AX\Phi}\phi \left(-\frac{xs}{\mu}+e\right). \end{aligned}$$

Substituting $\mu = \gamma\mu^0$, we obtain

$$\begin{aligned} \frac{\hat{u}}{\gamma} &= x^0\phi^0 P_{AX^0\Phi^0}\phi^0\rho, \\ u &= x\phi P_{AX\Phi}\phi\rho \end{aligned}$$

for $\rho = \left(-\frac{xs}{\mu}+e\right)$. The points x^k and x^{0^k} approach the relative interior of the optimal face, converging to a small compact neighborhood of the central optimum x^*. The vectors ϕ and ϕ^0 have the following bounds.

By construction, $w_i^0 \in [0.95, 1.05]$, $w_i \in [0.9, 1.1]$. Since $\phi_i = \frac{1}{\sqrt{w_i}}$ by definition, the following bounds easily can be checked:

$$\phi_i^0 \in [0.97, 1.03],\ \phi_i \in [0.95, 1.06],\ \frac{\phi_i^0}{\phi_i} \in [0.92, 1.08]. \tag{14}$$

Thus $x^0\phi^0$ and $x\phi$ also converge to compact sets. Since $\|\rho\| = \delta(x,s) \le 0.1$, the vectors $\phi\rho$ and $\phi^0\rho$ are also in compact sets, and we can use Proposition 3.2 to obtain

$$\begin{array}{lcl} \frac{\hat{u}_B}{\gamma} - x_B^0\phi_B^0 P_{A_B X_B^0 \Phi_B^0}\phi_B^0\rho_B & \to & 0, \\ u_B - x_B\phi_B P_{A_B X_B \Phi_B}\phi_B\rho_B & \to & 0. \end{array} \tag{15}$$

The scaled projections above are almost in the format required by Proposition 3.3, on slightly shifted scalings. To put them in the desired format, let us write

$$\rho_B = x_B^0 (x_B^0)^{-1}\rho_B.$$

Due to Proposition 4.1, since $x_B^0 = \Omega(1)$, we have

$$x_B = x_B^0 + u_B^0 = x_B^0(e + O(\mu^0)) \ . \tag{16}$$

It follows that $(x_B^0)^{-1} = x_B^{-1}(e + O(\mu^0))$. Thus,

$$\rho_B = x_B^0 x_B^{-1}\rho_B(e + O(\mu^0)) = x_B^0 x_B^{-1}\rho_B + O(\mu^0).$$

Since $O(\mu^0) \to 0$, (15) can be written as

$$\frac{\hat{u}_B}{\gamma} - x_B^0\phi_B^0 P_{A_B X_B^0 \Phi_B^0} x_B^0\phi_B^0 x_B^{-1}\rho_B \to 0, \tag{17}$$

$$u_B - x_B\phi_B P_{A_B X_B \Phi_B} x_B\phi_B x_B^{-1}\rho_B \to 0. \tag{18}$$

Defining $q = \dfrac{x_B^0\phi_B^0}{x_B\phi_B}$, we see from (14) and (16) that for μ sufficiently small, $q_i \in [0.9, 1.1]$, and thus $\|q-e\|_\infty \le 0.1$. Now (17) can be written as

$$\frac{\hat{u}_B}{\gamma} - x_B\phi_B q P_{A_B X_B \Phi_B Q} x_B\phi_B q\ x_B^{-1}\rho_B \to 0. \tag{19}$$

Defining $\hat{h}_B = qP_{A_B X_B \Phi_B Q} x_B \phi_B q\ x_B^{-1}\rho_B$, $h_B = P_{A_B X_B \Phi_B} x_B \phi_B\ x_B^{-1}\rho_B$, we see from Proposition 3.3 that

$$\|h_B - \hat{h}_B\| \leq 0.3\|h_B\|.$$

Dividing (18) by $x_B\phi_B$ and using scaled norms, it follows that

$$\|u_B\|_{x_B\phi_B} - \|h_B\| \to 0. \tag{20}$$

Subtracting (18) from (19) establishes that

$$\frac{\frac{\hat{u}_B}{\gamma} - u_B}{x_B\phi_B} + \hat{h}_B - h_B \to 0 \tag{21}$$

or (making the iteration indices explicit)

$$\begin{aligned} \|\tfrac{\hat{u}_B^k}{\gamma^k} - u_B^k\|_{x_B^k\phi_B^k} &\leq \|h_B^k - \hat{h}_B^k\| + \sigma_1^k, \quad \sigma_1^k \to 0 \\ &\leq 0.3\|h_B^k\| + \sigma_1^k \\ &\leq 0.3\|u_B^k\|_{x_B^k\phi_B^k} + \sigma_2^k, \end{aligned}$$

where the last inequality comes from (20), with $\sigma_2^k \to 0$.

Using Proposition 3.4 twice to relate $\|\cdot\|_{x_B^k\phi_B^k}$ and $\|\cdot\|$, we see that there exists a constant $K_1 > 0$ such that

$$\|\tfrac{\hat{u}_B^k}{\gamma^k} - u_B^k\| \leq K_1\|u_B^k\| + \theta_x^k,$$

where $\theta_x^k \to 0$. Finally,

$$\|\tfrac{\hat{u}_B^k}{\gamma^k}\| \leq \|u_B^k\| + K_1\|u_B^k\| + \theta_x^k.$$

The final statement follows from the fact that $\{u_B^k\}$ and $\{v_B^k\}$ are bounded, and $\gamma = O(\mu^0)$ from (iii) of Proposition 5.2. □

PROPOSITION 5.4. *Consider the quantities x^{0^k}, s^{0^k}, x^k, etc. generated by Algorithm* 5.1*, starting at* (x^{0^1}, s^{0^1}). *Then*

(i) *at all iterations $\hat{u} = O(\mu^0)$, $\hat{v} = O(\mu^0)$, and the total variation in (x, s) due to simplified Newton steps is bounded by $O(\mu^{0^1})$;*

(ii) *the sequences $\{(x^{0^k}, s^{0^k})\}$ and $\{(x^k, s^k)\}$ converge to a pair $(\bar{x}, \bar{s})$ in the optimal face.*

If the safeguard is activated an infinite number of times, [1] *then $(\bar{x}, \bar{s}) = (x^*, s^*)$, the central optimal pair. Otherwise, $(\bar{x}, \bar{s})$ is not necessarily equal to (x^*, s^*).*

Proof. (i): recall that $\mu(x, s) = \gamma\mu(x^0, s^0)$ and apply Proposition 2.1 with $\mu = \gamma\mu^0$ to obtain

$$\hat{u} = \gamma x^0\phi^0 P_{AX^0\Phi^0}\phi^0\rho$$

and

$$\hat{v} = \gamma s^0\phi^0 \tilde{P}_{AX^0\Phi^0}\phi^0\rho,$$

[1] We shall prove below that this hypothesis is vacuous, but it will be needed to establish a contradiction.

where $\rho = \left(-\frac{xs}{\mu} + e\right)$. Since $\delta(x,s) \leq 0.1$, we see that $\|\rho\| = \delta(x,s) \leq 0.1$. Moreover, since $\delta(x^0, s^0) = \|w(x^0, s^0) - e\| \leq 0.1$, we see that the components of $w(x^0, s^0)$ are contained in $[0.9, 1.1]$; hence the components of ϕ^0 are contained in $[0.95, 1.06]$. Also, the sequence $\{(x^{0^k}, s^{0^k})\}$ is bounded, and projection operators are bounded. It follows from the above expressions and the fact that all quantities are bounded that $\hat{u} = O(\gamma) = O(\mu^0)$ and $\hat{v} = O(\gamma) = O(\mu^0)$. Now apply Proposition 4.3 (i) to obtain the result on total variation.

(ii): if the safeguard is activated a finite number of times, the conclusion follows from (i), because then the sequences generated by the algorithm are Cauchy sequences. Otherwise, the convergence proof is similar to the proof for the MTY algorithm, presented in Gonzaga and Tapia [3].

We shall prove the result for the primal variables. The proof for the dual slacks is similar. Also, it is enough to prove that $x^{0^k} \to x^*$, since $u^{0^k} = O(\mu^{0^k}) \to 0$.

Assume by contradiction that the sequence $\{x^{0^k}\}$ has an accumulation point $\bar{x} \neq x^*$. Since $\bar{x}_N = x_N^* = 0$, we have

$$\sigma \equiv \|\bar{x}_B - x_B^*\|_* > 0.$$

Let $\mathcal{K} \subset \mathbb{N}$ be the set of iterations in which the safeguard is activated (MTY iterations). Our first step is to show that $\bar{x}$ must also be an accumulation point of $(x^{0^k})_{k \in \mathcal{K}}$.

Let $\mathcal{K}_1 \subset \mathbb{N}$ be a subsequence such that $x^{0^k} \xrightarrow{\mathcal{K}_1} \bar{x}$, and let $j(k)$ be the first index in $\mathcal{K}$ greater than or equal to k. Then for any $k \in \mathcal{K}_1$, $\|x^{0^{j(k)}} - x^{0^k}\| = O(\mu^{0^k})$ by (i), and thus $x^{0^{j(k)}} \xrightarrow{\mathcal{K}_1} \bar{x}$. Thus it is enough to consider in our assumption subsequences in $\mathcal{K}$.

Let $\{\epsilon^k\}$ be the sequence given by Proposition 4.2, and let $\bar{k}$ be such that for $k \geq \bar{k}$ the conclusions of that proposition are valid and $\epsilon^k < 0.5\sigma$.

Choose an index $j \geq \bar{k}$ with the following characteristics: $j \in \mathcal{K}$, $\|{x_B^0}^j - x_B^*\|_* < 1.1\sigma$, and the total variation of x due to simplified steps after j satisfies

$$\sum_{\substack{k \notin \mathcal{K} \\ k \geq j}} \|x^{0^{k+1}} - x^{0^k}\|_* < 0.05\sigma. \tag{22}$$

Such an index exists by definition of σ and by (i). We shall prove by induction that for $k \in \mathcal{K}$, $k > j$, $\|x_B^{0^k} - x_B^*\|_* < 0.95\sigma$.

(a) $\|{x_B^0}^{j+1} - x_B^*\|_* < 0.8 \times 1.1\sigma < 0.9\sigma$ by Proposition 4.2. Let $k' = j(j+1)$ be the next index in $\mathcal{K}$. Using (22),

$$\|{x_B^0}^{k'} - x_B^*\|_* \leq \|{x_B^0}^{j+1} - x_B^*\|_* + \|{x_B^0}^{k'} - {x_B^0}^{j+1}\|_* < 0.95\sigma.$$

(b) Assume that for an index $k \in \mathcal{K}$, $k > j$, $\|x_B^{0^k} - x_B^*\|_* < 0.95\sigma$. Then by Proposition 4.2, $\|{x_B^0}^{k+1} - x_B^*\|_* \leq \max\{\epsilon^k, 0.8\|x_B^{0^k} - x_B^*\|_*\} < 0.9\sigma$. As in (a), using (22), let $k' = j(k+1)$ be the next index in $\mathcal{K}$:

$$\|{x_B^0}^{k'} - x^*\|_* \leq \|{x_B^0}^{k+1} - x^*\|_* + \|{x_B^0}^{k'} - {x_B^0}^{k+1}\|_* \leq 0.95\sigma.$$

Statements (a) and (b) prove that for all $k \in \mathcal{K}$, $k > j$, $\|x_B^{0^k} - x_B^*\|_* < 0.95\sigma$, contradicting the fact that σ is an accumulation point of the sequence $(\|x_B^{0^k} - x_B^*\|_*)_{k \in \mathcal{K}}$ and completing the proof. □

Having described the behavior of the large variables, we can now compare the small variables for the exact and simplified Newton corrector steps.

At a typical iteration, the simplified step $(\hat{u}, \hat{v})$ and the exact step (u, v) satisfy the equations below:

$$\begin{array}{rcl} x_B^0 \hat{v}_B + s_B^0 \hat{u}_B & = & -x_B s_B + \mu e_B, \\ x_N^0 \hat{v}_N + s_N^0 \hat{u}_N & = & -x_N s_N + \mu e_N, \end{array} \tag{23}$$

$$\begin{array}{rcl} x_B v_B + s_B u_B & = & -x_B s_B + \mu e_B, \\ x_N v_N + s_N u_N & = & -x_N s_N + \mu e_N, \end{array} \tag{24}$$

where $\mu = \gamma \mu^0, \;\; \gamma = O(\mu^0)$.

Before we state the main result, we establish some relationships within a typical iteration:

(i) Large variables: since $u^0 = O(\mu^0), \;\; v^0 = O(\mu^0)$, and all components of x_B and s_N are bounded away from zero,

$$x_B^0 = x_B(e + O(\mu^0)) \;, \; s_N^0 = s_N(e + O(\mu^0)). \tag{25}$$

(ii) Small variables: by construction,

$$x^0 s^0 = \mu^0 w^0,$$
$$x \; s = \mu \; w,$$

where $w_i^0 \in [0.95, 1.05], \;\; w_i \in [0.9, 1.1], \;\; i = 1, \ldots, n$. Dividing these expressions,

$$\frac{x_N^0}{x_N} = \frac{1}{\gamma} \frac{s_N}{s_N^0} \frac{w_N^0}{w_N}, \quad \frac{s_B^0}{s_B} = \frac{1}{\gamma} \frac{x_B}{x_B^0} \frac{w_B^0}{w_B}.$$

From (25), it is immediate that $s_N / s_N^0 = (e + O(\mu^0))$, and $x_B / x_B^0 = (e + O(\mu^0))$. By a simple calculation, $w_i^0 / w_i \in [0.85, 1.17], i = 1, \ldots, n$.

Defining

$$\sigma_N = \frac{s_N}{s_N^0} \; \frac{w_N^0}{w_N}, \;\; \sigma_B = \frac{x_B}{x_B^0} \; \frac{w_B^0}{w_B},$$

it follows that for sufficiently small μ^0,

$$\sigma_i \in [0.8, 1.2],$$

and we can write

$$x_N^0 = \frac{1}{\gamma} \sigma_N x_N \;, \; s_B^0 = \frac{1}{\gamma} \sigma_B s_B. \tag{26}$$

PROPOSITION 5.5. *Consider an application of Algorithm* 5.1. *Then the safeguard cannot be activated an infinite number of times.*

Proof. Assume by contradiction that the safeguard is activated at the iterations with indices in an infinite set $\mathcal{K}$.

From Proposition 5.4, the sequences $(x^0, s^0)^k$ and $(x, s)^k$ converge to the analytic center (x^*, s^*) of the optimal face. Using Lemma 6.2 of [3], we conclude that the sequence $(x_B^k + u_B^k)$ also converges to x_B^*, which implies that

$$u^k \to 0, \quad v^k \to 0, \tag{27}$$

where the second part follows from the symmetry of the steps. Also, Proposition 5.4 (i) shows that

$$\hat{u}^k \to 0, \quad \hat{v}^k \to 0. \tag{28}$$

Let us substitute relations (25) and (26) into the Newton equations (23). We shall analyze the first equation (indices in B); the analysis for the other one is similar. Our approach is to compare the behavior of the small variables in the simplified and exact corrector steps. To begin with,

$$(e + O(\mu^0))x_B\hat{v}_B + \frac{1}{\gamma}\sigma_B s_B \hat{u}_B \; = \; -x_B s_B + \mu e_B. \tag{29}$$

By subtracting (24) from (29) and restoring the iteration indices, we get

$$((e + O(\mu^{0^k}))\hat{v}_B^k - v_B^k)x_B^k \; = \; -\left(\frac{1}{\gamma^k}\sigma_B^k \hat{u}_B^k - u_B^k\right) s_B^k.$$

Taking norms,

$$\|(e + O(\mu^{0^k}))\hat{v}_B^k - v_B^k)x_B^k\| \; \leq \; \|s_B^k\|_\infty \quad \left(\|\sigma_B^k\|_\infty \frac{1}{\gamma^k}\|\hat{u}_B^k\| \; + \; \|u_B^k\|\right).$$

From Proposition 5.3, $\|\hat{u}_B^k\|/\gamma^k \leq K(\|u_B^k\| + \theta_x^k)$, where $\theta_x^k \to 0$. Since $\|\sigma_B^k\|_\infty \leq 1.2$ for sufficiently large k and $\|s_B^k\|_\infty = O(\mu^k)$ by Proposition 4.1, the inequality becomes

$$\begin{aligned} \|((e + O(\mu^{0^k}))\hat{v}_B^k - v_B^k)x_B^k\| &\leq O(\mu^k)(1.2K(\|u_B^k\| + \theta_x^k) + \|u_B^k\|) \\ &\leq K_1\mu^k(\|u_B^k\| + \theta_x^k), \end{aligned}$$

where K_1 is a constant that depends on the problem data. Since $u_B^k \to 0$ by (27) and since $\theta_x^k \to 0$, we conclude that

$$\frac{1}{\mu^k}((e + O(\mu^{0^k}))\hat{v}_B^k - v_B^k)x_B^k \to 0.$$

Since x_B^k has all components bounded away from zero,

$$(e + O(\mu^{0^k}))\frac{\hat{v}_B^k - v_B^k}{\mu^k} + \frac{O(\mu^{0^k})}{\mu^k}v_B^k \to 0,$$

and since $v_B^k = O(\mu^k)$ by Proposition 4.1, we conclude that

$$\frac{v_B^k - \hat{v}_B^k}{\mu^k} \longrightarrow 0, \quad \frac{u_N^k - \hat{u}_N^k}{\mu^k} \longrightarrow 0. \tag{30}$$

The second expression is obtained by a similar process, using the second equation in (23).

Now we shall establish a contradiction. At a typical iteration, let

$$w^+ = \frac{(x+u)(s+v)}{\mu}, \quad \hat{w} = \frac{(x+\hat{u})(s+\hat{v})}{\mu}.$$

From the analysis of the MTY algorithm presented in section 4, we see that

$$\|w^+ - e\| \le \tfrac{\alpha^2}{\sqrt{2}} < 0.01 \ .$$

At any iteration $k \in \mathcal{K}$,

$$\|\hat{w} - e\| > \tfrac{\alpha}{2} \ge 0.05.$$

At such an iteration, either $\|\hat{w}_N - e_N\| > 0.02$ or $\|\hat{w}_B - e_B\| > 0.02$. Assume that at an infinite number of iterations $\mathcal{K}_1 \subset \mathcal{K}$, $\|\hat{w}_N - e_N\| > 0.02$ (the analysis for the other case is completely similar).

Then for $k \in \mathcal{K}_1$,

$$\|\hat{w}_N - e_N\| > 0.02, \quad \|w_N^+ - e_N\| < 0.01.$$

This implies that in these iterations,

$$\|\hat{w}_N - w_N^+\| \quad = \quad \|(\hat{w}_N - e_N) - (w_N^+ - e_N)\| \ge 0.01. \tag{31}$$

On the other hand, we have (by definition)

$$\begin{aligned} \mu w_N^+ &= (x_N + u_N)\,(s_N + v_N), \\ \mu \hat{w}_N &= (x_N + \hat{u}_N)\,(s_N + \hat{v}_N). \end{aligned}$$

By subtracting, we get

$$\mu(\hat{w}_N - w_N^+) = (x_N + \hat{u}_N)\,(s_N + \hat{v}_N) - (x_N + u_N)\,(s_N + v_N).$$

Reordering terms in this expression, we obtain

$$\hat{w}_N - w_N^+ = \frac{\hat{u}_N - u_N}{\mu}(s_N + \hat{v}_N) + \frac{x_N + u_N}{\mu}(\hat{v}_N - v_N).$$

Let us analyze the terms in the right-hand side (restoring the index k):

(i) by (30), $\dfrac{\hat{u}_N^k - u_N^k}{\mu^k}(s_N^k + v_N^k) \to 0$;

(ii) by Proposition 4.1, $x_N^k = O(\mu^k)$ and $u_N^k = O(\mu^k)$. From (27) and (28), $v^k \to 0$ and $\hat{v}^k \to 0$. Hence

$$\left\| \frac{x_N^k + u_N^k}{\mu^k}(\hat{v}_N^k - v_N^k) \le K_2 \right\| \hat{v}_N^k - v_N^k,$$

where K_2 depends on problem data, so this term converges to zero.

We conclude that $(w_N^+)^k - \hat{w}_N^k \longrightarrow 0$, contradicting (31), and complete the proof. □

We are now ready to formally state our convergence results.

THEOREM 5.1. *Let $\{(x^0, s^0)^k\}$ and $\{(x, s)^k\}$ denote the sequences generated by the simplified MTY predictor-corrector algorithm. Then*

(i) *the safeguard in the corrector step is activated only a finite number of times*;

(ii) *the algorithm has iteration complexity* $O(\sqrt{n}L)$;

(iii) *the duality-gap sequence* $\{{x^0}^T s^0\}$ *converges quadratically to zero*;

(iv) *both sequences* $\{(x^0, s^0)\}$ *and* $\{(x, s)\}$ *converge to a point* $(\bar{x}, \bar{s})$ *in the optimal face.*

Proof. Property (i) follows from Proposition 5.5. Also, (ii) follows from (iv) of Proposition 5.2 in a standard manner. See Mizuno, Todd, and Ye [11] for details. Property (iii) is a combination of (i), (ii), and (iii) of Proposition 5.2. Finally, (iv) is (ii) of Proposition 5.4. □

6. Concluding remarks. The fact that so much of Theorem 5.1 follows from Proposition 5.2 and Proposition 5.2 depends so little on the corrector step leads us to take a closer look at the role of the corrector step in our convergence theory.

Consider a typical simplified MTY predictor-corrector iteration represented by $\{(x^0, s^0), (x, s), (x^+, s^+)\}$. The predictor step takes (x^0, s^0) to (x, s) and the corrector step takes (x, s) to (x^+, s^+). A close look at the derivation of our theory shows that for the establishment of $O(\sqrt{n}L)$ complexity and quadratic convergence, we only used the fact that the corrector step satisfies

$$(32) \qquad \begin{array}{rrl} \text{(i)} & {x^+}^T s^+ \leq & x^T s, \text{ and} \\ \text{(ii)} & \delta(x^+, s^+) \leq & \frac{\alpha}{2}. \end{array}$$

Hence any corrector step satisfying (32) will lead to $O(\sqrt{n}L)$ complexity and quadratic convergence but not necessarily iteration sequence convergence. It follows that quadratic convergence is the best that should be expected from either the MTY algorithm or the simplified MTY predictor-corrector algorithm. This is because for both of these algorithms, the corrector step does not improve the duality-gap. For example, ${x^+}^T s^+ = x^T s$, and therefore the quadratic decrease is obtained entirely from the damped Newton predictor step, and quadratic decrease (in general) is optimal for a (damped) Newton method. Clearly the same is true for any corrector step that does not decrease the duality-gap.

We are accustomed to expect cubic decrease from the pair consisting of a Newton step and a simplified Newton step and quartic decrease from the pair consisting of two Newton steps. In order to attain these objectives along with $O(\sqrt{n}L)$ complexity, the predictor-corrector approach will have to be modified so that the corrector step still satisfies (32) but also gives the appropriate decrease in the duality-gap. For example, if we replace μ with $\gamma\mu$ in the simplified corrector step of Algorithm 5.1 and the safeguard is activated only a finite number of times, then we would obtain cubic convergence from the simplified MTY algorithm. We did not pursue this issue in the present work.

The contribution of this paper is the demonstration that in the MTY predictor-corrector algorithm the Newton corrector step can be replaced with a safeguarded simplified Newton corrector step and all the algorithmic properties are maintained, except that the convergence of the iteration sequence is no longer to the analytic center. Whether this loss is important or not clearly depends on the application.

Acknowledgment. This paper owes much to Mike Todd, who made numerous comments and suggestions that led to improvements in the quality of the proofs and in the presentation. The authors are grateful for his unusual level of detail in reading and understanding the paper while acting as a referee. They also acknowledge the other (anonymous) referee for making many useful observations.

REFERENCES

[1] I. Adler and R. D. C. Monteiro, *Limiting behavior of the affine scaling continuous trajectories for linear programming problems*, Math. Programming, 50 (1991), pp. 29–51.

[2] C. Gonzaga, *Path following algorithms for linear programming*, SIAM Rev., 34 (1992), pp. 167–224.

[3] C. Gonzaga and R. A. Tapia, *On the Convergence of the Mizuno–Todd–Ye Algorithm to the Analytic Center of the Solution Set*, Technical report TR92-41, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1992.

[4] M. Kojima, S. Mizuno, and A. Yoshise, *A primal-dual interior-point method for linear programming*, in Progress in Mathematical Programming, Interior-point and Related Methods, Nimrod Megiddo, ed., Springer-Verlag, Berlin, New York, 1989, pp. 29–47.

[5] M. Kojima, S. Mizuno, and A. Yoshise, *An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems*, Math. Programming, 90 (1991), pp. 331–342.

[6] I. J. Lustig, R. E. 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] K. McShane, *Superlinearly convergent $O(\sqrt{n}L)$-iteration interior-point algorithm for linear programming and the monotone linear complementarity problem*, SIAM J. Optim., 4 (1994), pp. 247–261.

[8] N. Megiddo, *Pathways to the optimal set in linear programming*, in Progress in Mathematical Programming, Interior-point and Related Methods, Nimrod Megiddo, ed., Springer-Verlag, Berlin, New York, 1989, pp. 131–158.

[9] N. Megiddo and M. Shub, *Boundary behaviour of interior point algorithms in linear programming*, Math. Oper. Res., 14 (1989), pp. 97–146.

[10] S. Mehrotra, *Quadratic convergence in a primal-dual method*, Math. Oper. Res., 18 (1993), pp. 741–751.

[11] 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.

[12] R. C. Monteiro and I. Adler, *Interior path-following primal-dual algorithm. Part I: Linear programming*, Math. Programming, 44 (1989), pp. 27–41.

[13] R. A. Tapia, Y. Zhang, and Y. Ye, *On the convergence of the iteration sequence in primal-dual interior point methods*, Math. Programming, 68 (1995), pp. 141–154.

[14] M. J. Todd and Y. Ye, *A centered projective algorithm for linear programming*, Math. Oper. Res., 15 (1990), pp. 508–529.

[15] Y. Ye, *On the Q-order of convergence of interior-point algorithms for linear programming*, in Proc. 1992 Symposium on Applied Mathematics, Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing, China, 1992, pp. 534–543.

[16] Y. Ye, O. Güler, R. A. Tapia, and Y. Zhang, *A quadratically convergent $O(\sqrt{n}L)$-iteration algorithm for linear programming*, Math. Programming, 59 (1993), pp. 151–162.

[17] Y. Ye, R. A. Tapia, and Y. Zhang, *A Superlinearly Convergent $O(\sqrt{n}L)$-Iteration Algorithm for Linear Programming*, Technical report TR91-22, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1991.

[18] Y. Zhang and R. A. Tapia, *A superlinearly convergent polynomial primal-dual interior-point algorithm for linear programming*, SIAM J. Optim., 3 (1993), pp. 118–133.

[19] Y. Zhang and R. A. Tapia, *Superlinear and quadratic convergence of primal-dual interior-point methods for linear programming revisited*, J. Optim. Theory Appl., 73 (1992), pp. 229–242.

[20] Y. Zhang, R. A. Tapia, and J. E. Dennis, *On the superlinear and quadratic convergence of primal-dual interior-point linear programming algorithms*, SIAM J. Optim., 2 (1992), pp. 304–324.

SIAM J. Optim. © 1997 Society for Industrial and Applied Mathematics
Vol. 7, No. 1, pp. 86–125, February 1997 006

INTERIOR-POINT METHODS FOR THE MONOTONE SEMIDEFINITE LINEAR COMPLEMENTARITY PROBLEM IN SYMMETRIC MATRICES[*]

MASAKAZU KOJIMA[†], SUSUMU SHINDOH[‡], AND SHINJI HARA[§]

Abstract. The SDLCP (semidefinite linear complementarity problem) in symmetric matrices introduced in this paper provides a unified mathematical model for various problems arising from systems and control theory and combinatorial optimization. It is defined as the problem of finding a pair (X, Y) of $n \times n$ symmetric positive semidefinite matrices which lies in a given $n(n+1)/2$ dimensional affine subspace $\mathcal{F}$ of $\mathcal{S}^2$ and satisfies the complementarity condition $X \bullet Y = 0$, where $\mathcal{S}$ denotes the $n(n+1)/2$-dimensional linear space of symmetric matrices and $X \bullet Y$ the inner product of X and Y. The problem enjoys a close analogy with the LCP in the Euclidean space. In particular, the central trajectory leading to a solution of the problem exists under the nonemptiness of the interior of the feasible region and a monotonicity assumption on the affine subspace $\mathcal{F}$. The aim of this paper is to establish a theoretical basis of interior-point methods with the use of Newton directions toward the central trajectory for the monotone SDLCP.

Key words. interior-point method, linear complementarity problem, linear matrix inequality, semidefinite program, linear program

AMS subject classifications. 90C33, 90C05, 90C25

PII. S1052623494269035

1. Introduction.

Let $\hat{\mathcal{S}}$ denote the set of all $n \times n$ real matrices and $\mathcal{S}$ the set of all $n \times n$ symmetric real matrices. We identify $\hat{\mathcal{S}}$ with the n^2-dimensional Euclidean space $R^{n \times n}$ and $\mathcal{S}$ with an $n(n+1)/2$-dimensional linear subspace of $\hat{\mathcal{S}} = R^{n \times n}$. The *inner product* $\boldsymbol{X} \bullet \boldsymbol{Y}$ of $\boldsymbol{X}$ and $\boldsymbol{Y}$ in the linear space $\hat{\mathcal{S}}$ is Tr $\boldsymbol{X}^T\boldsymbol{Y}$, i.e., the trace of $\boldsymbol{X}^T\boldsymbol{Y}$. We write $\boldsymbol{X} \succ \boldsymbol{O}$ if $\boldsymbol{X} \in \hat{\mathcal{S}}$ is positive definite, i.e., $\boldsymbol{u}^T\boldsymbol{X}\boldsymbol{u} > 0$ for every nonzero $\boldsymbol{u} \in R^n$, and $\boldsymbol{X} \succeq \boldsymbol{O}$ if $\boldsymbol{X}$ is positive semidefinite, i.e., $\boldsymbol{u}^T\boldsymbol{X}\boldsymbol{u} \geq 0$ for every $\boldsymbol{u} \in R^n$. Here $\boldsymbol{O}$ stands for the $n \times n$ zero matrix. We use the symbol $\mathcal{S}_+$ for the set of symmetric positive semidefinite matrices and $\mathcal{S}_{++}$ for the set of symmetric positive definite matrices,

$$\mathcal{S}_+ = \{\boldsymbol{X} \in \mathcal{S} : \boldsymbol{X} \succeq \boldsymbol{O}\},\ \mathcal{S}_{++} = \{\boldsymbol{X} \in \mathcal{S} : \boldsymbol{X} \succ \boldsymbol{O}\}.$$

This paper introduces the SDLCP (the semidefinite linear complementarity problem) in symmetric matrices: find an $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}^2$ such that

$$(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F},\ \boldsymbol{X} \succeq \boldsymbol{O},\ \boldsymbol{Y} \succeq \boldsymbol{O} \text{ and } \boldsymbol{X} \bullet \boldsymbol{Y} = 0. \tag{1}$$

Here $\mathcal{F}$ is an $n(n+1)/2$-dimensional affine subspace of $\mathcal{S}^2$. We call $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}$ with $\boldsymbol{X} \succeq \boldsymbol{O}$ and $\boldsymbol{Y} \succeq \boldsymbol{O}$ a *feasible solution* of the SDLCP (1) and $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}$ with

[*] Received by the editors June 3, 1994; accepted for publication (in revised form) September 1, 1995. This work was partially supported by the Okawa Institute of Information and Telecommunication.

http://www.siam.org/journals/siopt/7-1/26903.html

[†] Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro-ku, Tokyo 152, Japan (kojima@is.titech.ac.jp).

[‡] Department of Mathematics and Physics, The National Defense Academy, Hashirimizu 1-10-20, Yokosuka, Kanagawa, 239, Japan (shindoh@cc.nda.ac.jp).

[§] Department of Computational Intelligence and Systems Science, Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, Nagatsuta-cho, Midori-ku, Yokohama 227, Japan (hara@ssie.titech.ac.jp).

$\boldsymbol{X} \succ \boldsymbol{O}$ and $\boldsymbol{Y} \succ \boldsymbol{O}$ an *interior feasible solution* of the SDLCP (1). We impose a certain monotonicity condition (Condition 1.2 below) of the affine subspace $\mathcal{F}$. *The purpose of this paper is to establish a general theoretical framework of interior-point methods for the monotone SDLCP* (1).

The SDLCP is a generalization of SDPs (semidefinite programs) which have various applications in systems and control theory and combinatorial optimization. See [1, 2, 3, 5, 12, 15, 35, 44, 45], etc. Given $\boldsymbol{C} \in \mathcal{S}$, $\boldsymbol{A}_i \in \mathcal{S}$ $(i = 1, 2, \ldots, m)$, and $b_i \in R$ $(i = 1, 2, \ldots, m)$, a primal-dual pair of SDPs is defined as

$$(2) \qquad \left\{ \begin{array}{lll} \mathcal{P}: & \text{minimize} & \boldsymbol{C} \bullet \boldsymbol{X} \\ & \quad \text{subject to} & \boldsymbol{A}_i \bullet \boldsymbol{X} = b_i \ (i = 1, 2, \ldots, m), \\ & & \boldsymbol{X} \succeq \boldsymbol{O} \ (\text{ or } \boldsymbol{X} \in \mathcal{S}_+), \\ \mathcal{D}: & \text{maximize} & \sum_{i=1}^m b_i z_i \\ & \quad \text{subject to} & \sum_{i=1}^m \boldsymbol{A}_i z_i + \boldsymbol{Y} = \boldsymbol{C}, \\ & & \boldsymbol{Y} \succeq \boldsymbol{O} \ (\text{ or } \boldsymbol{Y} \in \mathcal{S}_+). \end{array} \right.$$

We call $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{z}) \in \mathcal{S}^2 \times R^m$ a *feasible solution* of the primal-dual pair (2) of SDPs if it satisfies all the constraints in (2) and an *interior feasible solution* of (2) if it satisfies $\boldsymbol{X} \succ \boldsymbol{O}$ and $\boldsymbol{Y} \succ \boldsymbol{O}$ in addition to the constraints. Let

$$(3) \qquad \mathcal{F} = \left\{ (\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}^2 \ : \ \begin{array}{l} \boldsymbol{A}_i \bullet \boldsymbol{X} = b_i \ (i = 1, 2, \ldots, m), \\ \sum_{i=1}^m \boldsymbol{A}_i z_i + \boldsymbol{Y} = \boldsymbol{C} \ \text{ for some } \boldsymbol{z} \in R^m \end{array} \right\}.$$

Then $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{z}) \in \mathcal{S}^2 \times R^m$ is a feasible solution (or an interior feasible solution) of the primal-dual pair (2) of SDPs if and only if $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}^2$ is a feasible solution (or an interior feasible solution) of the SDLCP (1). Furthermore, if we assume that there is an interior feasible solution of (2), we can state a common necessary and sufficient optimality condition for $\boldsymbol{X}$ to be a minimum solution of the primal problem $\mathcal{P}$ and $(\boldsymbol{Y}, \boldsymbol{z})$ to be a maximum solution of the dual problem $\mathcal{D}$ in terms of the SDLCP (1). In this case, the $n(n+1)/2$-dimensional affine subspace $\mathcal{F}$ enjoys the self-orthogonality, i.e.,

$$(\boldsymbol{X}' - \boldsymbol{X}) \bullet (\boldsymbol{Y}' - \boldsymbol{Y}) = 0 \ \text{ for every } (\boldsymbol{X}', \boldsymbol{Y}'), \ (\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F},$$

which is a special case of the monotonicity (see Condition 1.2 below). Therefore the monotone SDLCP (1) is at least as general as the primal-dual pair (2) of SDPs, and we can specialize and/or modify both theoretical results and interior-point methods presented in this paper to adapt them to the primal-dual pair (2) of SDPs. However, our primary concern is not an extension of the primal-dual pair of SDPs but rather a basic idea for designing a wide class of interior-point methods for mathematical programs in the space of symmetric matrices.

A distinctive and important feature of our theoretical framework of interior-point methods for the monotone SDLCP (1) is the use of “a Newton direction for approximating a point on the central trajectory” at each iteration. This feature enables us to transfer many useful technologies developed in the class of primal-dual interior-point methods for LPs (linear programs) ([9, 20, 25, 26, 27, 30, 33, 34, 40, 41], etc.) and their extensions to LCPs (linear complementarity problems) ([19, 21, 22], etc.) and horizontal LCPs ([47, 48, 49], etc.). Indeed the Generic Interior-Point Method presented in section 5 opens up the possibilities of extensions of a great variety of primal-dual interior-point methods developed so far — central trajectory following methods, potential-reduction methods, predictor-corrector methods, infeasible interior-point methods, etc.—to the monotone SDLCP (1).

In recent years, many studies ([1, 5, 45, 15, 35, 44], etc.) have been done on extensions of interior-point methods developed for LPs to SDPs. Our primal-dual interior-point methods for the SDLCP (1) are built on the same materials, the logarithmic barrier function, the central trajectory, the potential function, etc., as those used in the existing ones. In particular, we follow "a recipe" proposed by Alizadeh [2] to extend known interior-point algorithms for LPs into similar algorithms for SDPs using those materials (see Figures 2.1 and 3.2 of [2]). Alizadeh gave "a direct extension of Ye's projective potential-reduction method" [46] based on his recipe. It should be emphasized, however, that our extension of primal-dual interior-point methods to the SDLCP (1) is not so direct as in the case of Ye's projective potential-reduction method. There is a brief discussion of the difficulty below.

We use the notation $\boldsymbol{a}\bullet\boldsymbol{b}$ to denote the inner product $\sum_{j=1}^n a_j b_j$ of every $\boldsymbol{a}$, $\boldsymbol{b} \in R^n$ and the notation diag $\boldsymbol{a}$ to denote the $n\times n$ diagonal matrix with the diagonal elements $a_1, a_2, \ldots, a_n$ for every $\boldsymbol{a} = (a_1, a_2, \ldots, a_n) \in R^n$. Let $\boldsymbol{c} \in R^n$, $\boldsymbol{a}_i \in R^n$ $(i = 1, 2, \ldots, m)$. Consider the primal-dual pair of LPs:

$$(4) \qquad \left\{ \begin{array}{lll} \mathcal{P}: & \text{minimize} & \boldsymbol{c} \bullet \boldsymbol{x} \\ & \text{subject to} & \boldsymbol{a}_i \bullet \boldsymbol{x} = b_i \ (i = 1, 2, \ldots, m), \\ & & \boldsymbol{x} \geq \boldsymbol{0}, \\ \mathcal{D}: & \text{maximize} & \sum_{i=1}^m b_i z_i \\ & \text{subject to} & \sum_{i=1}^m \boldsymbol{a}_i z_i + \boldsymbol{y} = \boldsymbol{c}, \\ & & \boldsymbol{y} \geq \boldsymbol{0}. \end{array} \right.$$

By taking

$$\begin{array}{l} \boldsymbol{C} = \text{diag } \boldsymbol{c}, \ \boldsymbol{A}_i = \text{diag } \boldsymbol{a}_i \ (i = 1, 2, \ldots, m), \ \boldsymbol{O} = \text{ diag } \boldsymbol{0}, \\ \boldsymbol{X} = \text{diag } \boldsymbol{x}, \text{ and } \boldsymbol{Y} = \text{ diag } \boldsymbol{y}, \end{array}$$

we embed the primal-dual LPs (4) into the primal-dual SDPs (2). This convenience makes it possible for us to simultaneously present one iteration of primal-dual interior-point methods for LPs and SDPs. Suppose that we know an interior feasible solution $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{z}) \in \mathcal{S}^2 \times R^m$ of the primal-dual pair (2) of SDPs. Alizadeh's recipe leads us to the Newton equation

$$(5) \qquad \left\{ \begin{array}{l} d\boldsymbol{X}\boldsymbol{Y} + \boldsymbol{X}d\boldsymbol{Y} = \mu\boldsymbol{I} - \boldsymbol{X}\boldsymbol{Y}, \\ \boldsymbol{A}_i \bullet d\boldsymbol{X} = 0 \ (i = 1, 2, \ldots, m), \\ \sum_{i=1}^m \boldsymbol{A}_i dz_i + d\boldsymbol{Y} = \boldsymbol{O} \end{array} \right.$$

for a search direction $(d\boldsymbol{X}, d\boldsymbol{Y}, d\boldsymbol{z}) \in \mathcal{S}^2 \times R^m$, where $\mu > 0$ denotes a search direction parameter. Then we generate a new point $(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}, \bar{\boldsymbol{z}}) \in \mathcal{S}^2 \times R^m$ with appropriate step lengths $\alpha_p > 0$ and $\alpha_d > 0$ such that

$$\begin{array}{c} \bar{\boldsymbol{X}} = \boldsymbol{X} + \alpha_p \, d\boldsymbol{X}, \\ (\bar{\boldsymbol{Y}}, \bar{\boldsymbol{z}}) = (\boldsymbol{Y}, \boldsymbol{z}) + \alpha_d \, (d\boldsymbol{Y}, d\boldsymbol{z}). \end{array}$$

When we are concerned with the primal-dual pair (4) of LPs, all the matrices $\boldsymbol{A}_i$, $\boldsymbol{X}$, $\boldsymbol{Y}$, $d\boldsymbol{X}$, $d\boldsymbol{Y}$ appearing in the Newton equation (5) are diagonal; hence they are commutative. In this case, we can transform (5) into the system of equations

$$(6) \qquad \left\{ \begin{array}{l} \boldsymbol{D}^{-1}d\boldsymbol{X} + \boldsymbol{D}d\boldsymbol{Y} = \mu(\boldsymbol{X}\boldsymbol{Y})^{-1/2} - (\boldsymbol{X}\boldsymbol{Y})^{1/2}, \\ \boldsymbol{D}\boldsymbol{A}_i \bullet \boldsymbol{D}^{-1}d\boldsymbol{X} = 0 \ (i = 1, 2, \ldots, m), \\ \sum_{i=1}^m \boldsymbol{D}\boldsymbol{A}_i dz_i + \boldsymbol{D}d\boldsymbol{Y} = \boldsymbol{O} \end{array} \right.$$

in a "scaled Newton direction" $(\boldsymbol{D}^{-1}d\boldsymbol{X}, \boldsymbol{D}d\boldsymbol{Y}, d\boldsymbol{z})$, where $\boldsymbol{D} = \boldsymbol{X}^{1/2}\boldsymbol{Y}^{-1/2}$. We can easily verify that the scaled Newton direction $(\boldsymbol{D}^{-1}d\boldsymbol{X}, \boldsymbol{D}d\boldsymbol{Y}, d\boldsymbol{z})$ satisfies

$$(7)\quad (\boldsymbol{D}^{-1}d\boldsymbol{X}) + (\boldsymbol{D}d\boldsymbol{Y}) = \mu(\boldsymbol{X}\boldsymbol{Y})^{-1/2} - (\boldsymbol{X}\boldsymbol{Y})^{1/2} \text{ and } (\boldsymbol{D}^{-1}d\boldsymbol{X}) \bullet (\boldsymbol{D}d\boldsymbol{Y}) = 0.$$

This relation has been playing a crucial role in the development of primal-dual interior-point algorithms for LPs. See [20, Section 2].

In the case of SDPs, we can derive from the Newton equation (5) a similar system of equations,

$$(6)' \qquad \begin{cases} \sqrt{\boldsymbol{X}}^{-1}d\boldsymbol{X}\sqrt{\boldsymbol{Y}} + \sqrt{\boldsymbol{X}}d\boldsymbol{Y}\sqrt{\boldsymbol{Y}}^{-1} = \mu\sqrt{\boldsymbol{X}}^{-1}\sqrt{\boldsymbol{Y}}^{-1} - \sqrt{\boldsymbol{X}}\sqrt{\boldsymbol{Y}}, \\ \sqrt{\boldsymbol{X}}\boldsymbol{A}_i\sqrt{\boldsymbol{Y}}^{-1} \bullet \sqrt{\boldsymbol{X}}^{-1}d\boldsymbol{X}\sqrt{\boldsymbol{Y}} = 0 \ (i = 1, 2, \dots, m), \\ \sum_{i=1}^{m} \sqrt{\boldsymbol{X}}\boldsymbol{A}_i\sqrt{\boldsymbol{Y}}^{-1}dz_i + \sqrt{\boldsymbol{X}}d\boldsymbol{Y}\sqrt{\boldsymbol{Y}}^{-1} = \boldsymbol{O}, \end{cases}$$

in a "scaled Newton direction" $(\sqrt{\boldsymbol{X}}^{-1}d\boldsymbol{X}\sqrt{\boldsymbol{Y}}, \sqrt{\boldsymbol{X}}d\boldsymbol{Y}\sqrt{\boldsymbol{Y}}^{-1}, d\boldsymbol{z})$ and a similar relation,

$$(7)' \qquad \begin{cases} \sqrt{\boldsymbol{X}}^{-1}d\boldsymbol{X}\sqrt{\boldsymbol{Y}} + \sqrt{\boldsymbol{X}}d\boldsymbol{Y}\sqrt{\boldsymbol{Y}}^{-1} = \mu\sqrt{\boldsymbol{X}}^{-1}\sqrt{\boldsymbol{Y}}^{-1} - \sqrt{\boldsymbol{X}}\sqrt{\boldsymbol{Y}}, \\ \sqrt{\boldsymbol{X}}^{-1}d\boldsymbol{X}\sqrt{\boldsymbol{Y}} \bullet \sqrt{\boldsymbol{X}}d\boldsymbol{Y}\sqrt{\boldsymbol{Y}}^{-1} = 0. \end{cases}$$

But (6) and (7) do not hold any more because the matrices $\boldsymbol{X}$, $\boldsymbol{Y}$, $d\boldsymbol{X}$, and $d\boldsymbol{Y}$ appearing in the Newton equation (5) are $n \times n$ general symmetric matrices whose multiplication is not necessarily commutative, and the derivation of the scaled Newton equation (6) essentially relies on the commutativity of these matrices. This noncommutativity of general symmetric matrices certainly causes some difficulty in straightforward extensions of primal-dual interior-point methods to the SDP (2). What is worse and more substantial in the case of SDPs, however, is that the Newton equation (5) does not necessarily have a symmetric solution (i.e., a solution $(d\boldsymbol{X}, d\boldsymbol{Y}, d\boldsymbol{z})$ with symmetric $d\boldsymbol{X}$ and $d\boldsymbol{Y}$). (This fact was also pointed out in the recent paper [3] by Alizadeh, Haeberly, and Overton. They proposed some variants of the Newton equation which are different from (5) to get a symmetric search direction.) Therefore, following Alizadeh's recipe is not enough to generalize primal-dual interior-point methods from LPs to SDPs. In this paper, we will devise "a new system of equations" in a modified Newton direction towards the central trajectory and establish some fundamental results (including a system of equations similar to (6)' and a relation similar to (7)'; see Corollary 4.3) which are necessary to analyze the convergence of primal-dual interior-point methods using the modified Newton direction.

The SDLCP (1) presents an extraordinary similarity to the LCP in the Euclidean space: find an $(\boldsymbol{x}, \boldsymbol{y}) \in R^{2n}$ such that

$$(8) \qquad \boldsymbol{y} = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{q}, \ \boldsymbol{x} \geq \boldsymbol{0}, \ \boldsymbol{y} \geq \boldsymbol{0}, \ \text{ and } \boldsymbol{x} \bullet \boldsymbol{y} = 0,$$

where $\boldsymbol{M} \in \hat{\mathcal{S}}$ is a given constant matrix and $\boldsymbol{q} \in R^n$ a given constant vector. Letting F be the n-dimensional affine subspace $\{(\boldsymbol{x}, \boldsymbol{y}) \in R^{2n} : \boldsymbol{y} = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{q}\}$, we can rewrite (8) as

$$(9) \qquad (\boldsymbol{x}, \boldsymbol{y}) \in F, \ \boldsymbol{x} \geq \boldsymbol{0}, \ \boldsymbol{y} \geq \boldsymbol{0}, \ \text{ and } \boldsymbol{x} \bullet \boldsymbol{y} = 0.$$

If we allow F to be a general affine subspace of R^{2n}, the LCP (9) is equivalent to the so-called horizontal linear complementarity problem (see, for example, [4, 6, 31, 43, 48]).

Thus we have the clear correspondence between the SDLCP (1) and the horizontal LCP (9) in the Euclidean space:

$$\begin{aligned}(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{F} \subset \mathcal{S}^2 &\Longleftrightarrow (\boldsymbol{x},\boldsymbol{y}) \in F \subset R^{2n},\\ \text{“} \succ,\ \succeq \text{”} &\Longleftrightarrow \text{“} >,\ \geq \text{”},\\ \boldsymbol{X} \bullet \boldsymbol{Y} &\Longleftrightarrow \boldsymbol{x} \bullet \boldsymbol{y}.\end{aligned}$$

(See also Figures 2.1 and 3.2 of [2].)

It is interesting to note that the sets $\mathcal{S}_+ = \{\boldsymbol{X} \in \mathcal{S} : \boldsymbol{X} \succeq \boldsymbol{O}\}$ and $\mathcal{S}_{++} = \{\boldsymbol{X} \in \mathcal{S} : \boldsymbol{X} \succ \boldsymbol{O}\}$ play the roles of the nonnegative orthant R^n_+ and the positive orthant R^n_{++}, respectively, in the space $\mathcal{S}$ of symmetric matrices. We have the following properties.

LEMMA 1.1.

1. *$\mathcal{S}_+$ is a closed convex cone in $\mathcal{S}$ and its interior coincides with $\mathcal{S}_{++}$.*
2. *$\{\boldsymbol{Y} \in \mathcal{S} : \boldsymbol{X} \bullet \boldsymbol{Y} \geq 0$ for every $\boldsymbol{X} \in \mathcal{S}_+\} = \mathcal{S}_+$ (self-polarity).*
3. *Let $\boldsymbol{X},\ \boldsymbol{Y} \in \mathcal{S}_+$. Then $\boldsymbol{X} \bullet \boldsymbol{Y} \geq 0$, and $\boldsymbol{X} \bullet \boldsymbol{Y} = 0$ if and only if $\boldsymbol{XY} = \boldsymbol{O}$ (Lemma* 2.3 *of* [1]*).*
4. *Suppose that $\boldsymbol{A} \in \mathcal{S}_{++}$ and $\alpha > 0$. Let λ_{min} be the minimum eigenvalue of $\boldsymbol{A}$. If $\boldsymbol{X} \in \mathcal{S}_+$ and $\boldsymbol{A} \bullet \boldsymbol{X} \leq \alpha$ then the sum of all eigenvalues of $\boldsymbol{X}$ is not greater than α/λ_{min}; hence the set $\{\boldsymbol{X} \in \mathcal{S}_+ : \boldsymbol{A} \bullet \boldsymbol{X} \leq \alpha\}$ is bounded.*

The properties 1, 2, and 4 are easily verified, and their proofs are omitted here. See [13] for further properties of $\mathcal{S}_+$. In view of property 3 of Lemma 1.1, we can rewrite the SDLCP (1) as $(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{F}, (\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{S}^2_+$, and $\boldsymbol{XY} = \boldsymbol{O}$.

Among many kinds of assumptions on the LCP (8) and the horizontal LCP (9), the monotonicity assumption

$$(\boldsymbol{x}' - \boldsymbol{x}) \bullet (\boldsymbol{y}' - \boldsymbol{y}) \geq 0 \ \text{ for every } (\boldsymbol{x}',\boldsymbol{y}') \text{ and } (\boldsymbol{x},\boldsymbol{y}) \in F$$

is the most popular one. Indeed, the monotone LCP and the monotone horizontal LCP have important applications to LPs and convex quadratic programs. We impose a similar assumption on the SDLCP (1) throughout the paper.

Condition 1.2. The $n(n+1)/2$-dimensional affine subspace $\mathcal{F}$ associated with the SDLCP (1) is monotone, i.e., $(\boldsymbol{X}' - \boldsymbol{X}) \bullet (\boldsymbol{Y}' - \boldsymbol{Y}) \geq 0$ for every $(\boldsymbol{X}',\boldsymbol{Y}')$ and $(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{F}$.

Suppose that F is an n-dimensional monotone affine subspace of R^{2n} and that the horizontal LCP (9) has an interior feasible solution, i.e., $(\boldsymbol{x}^0,\boldsymbol{y}^0) \in F$ such that $(\boldsymbol{x}^0,\boldsymbol{y}^0) > \boldsymbol{0}$. It is well known that for every $\mu > 0$ there exists a unique interior feasible solution $(\boldsymbol{x}(\mu),\boldsymbol{y}(\mu))$ satisfying $x_j(\mu)y_j(\mu) = \mu$ $(j = 1,2,\ldots,n)$ and that the set $C = \{(\boldsymbol{x}(\mu),\boldsymbol{y}(\mu)) : \mu > 0\}$ forms a smooth trajectory converging to a solution of the horizontal LCP (9) as $\mu \to 0$. The set C is called the central trajectory or the path of centers.

Remark 1.3. Megiddo [30] presented the result above in connection with interior-point methods for the monotone LCP (8) with $F = \{(\boldsymbol{x},\boldsymbol{y}) \in R^{2n} : \boldsymbol{y} = \boldsymbol{Mx} + \boldsymbol{q}\}$. It was shown recently that $\dim F \leq n$ for any monotone affine subspace F of R^{2n} and that any monotone horizontal LCP with F of dimension n is reducible to a positive semidefinite LCP (8) (see [4, 11, 39, 43]). Hence the result above is equivalent to the one by Megiddo [30] on the monotone LCP (8) with $F = \{(\boldsymbol{x},\boldsymbol{y}) \in R^{2n} : \boldsymbol{y} = \boldsymbol{Mx} + \boldsymbol{q}\}$. See also [19, 28] for the existence of the central trajectory for more general complementarity problems.

The central trajectory has provided us with a theoretical basis for a wide class of interior-point methods which originated from a primal-dual interior-point method

([20, 30, 34, 40], etc.) for LPs and later extended to the monotone LCP (8) ([19, 21, 22], etc.) and the monotone horizontal LCP ([48, 49], etc.). A common feature of methods in this class is to move in "a Newton direction for approximating a point on the central trajectory" at each iteration.

It is well known that convex quadratic programs in the Euclidean space can be transformed into the LCP (8) or the horizontal LCP (9) via the Karush–Kuhn–Tucker optimality condition. As an extension of the primal SDP $\mathcal{P}$ stated above, consider a convex quadratic program of the form

$$\text{QP : minimize } \boldsymbol{C} \bullet \boldsymbol{X} + \boldsymbol{X} \bullet (\boldsymbol{Q}\boldsymbol{X}) \text{ subject to } \boldsymbol{X} \in \mathcal{S}_+ \cap (\mathcal{L} + \boldsymbol{D}).$$

Here $\boldsymbol{Q} \in \mathcal{S}_+$ is a given matrix and $\mathcal{L}$ a given linear subspace of $\mathcal{S}$. Then it is easily verified that $\boldsymbol{X} \in \mathcal{S}$ is a minimum solution of the quadratic program if the conditions

$$\begin{aligned} &\boldsymbol{X} \in \mathcal{S}_+ \cap (\mathcal{L} + \boldsymbol{D}),\ \boldsymbol{Y} - (\boldsymbol{Q}\boldsymbol{X} + \boldsymbol{X}\boldsymbol{Q}) \in \left(\mathcal{L}^{\perp} + \boldsymbol{C}\right), \\ &\boldsymbol{Y} \in \mathcal{S}_+, \quad \text{and } \boldsymbol{X} \bullet \boldsymbol{Y} = 0 \end{aligned}$$

hold for some $\boldsymbol{Y} \in \mathcal{S}_+$, where $\mathcal{L}^{\perp}$ is the orthogonal complement of $\mathcal{L}$. We can rewrite these conditions as the SDLCP (1) with the affine subspace

$$\mathcal{F} = \{(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}^2 : \boldsymbol{Y} - (\boldsymbol{Q}\boldsymbol{X} + \boldsymbol{X}\boldsymbol{Q}) \in \left(\mathcal{L}^{\perp} + \boldsymbol{C}\right),\ \boldsymbol{X} \in (\mathcal{L} + \boldsymbol{D})\}.$$

It is easily verified that $\mathcal{F}$ is an $n(n+1)/2$-dimensional monotone affine subspace. Thus we can apply the generic IP method described in section 5 to the convex QP. It should be noted, however, that the convex QP is equivalent to an SDP of the form

$$\begin{array}{lll} \text{SDP:} & \text{minimize} & \boldsymbol{C} \bullet \boldsymbol{X} + \boldsymbol{I} \bullet \boldsymbol{Z} \\ & \text{subject to} & \boldsymbol{X} \in \mathcal{S}_+ \cap (\mathcal{L} + \boldsymbol{D}), \\ & & \begin{pmatrix} \boldsymbol{I} & \boldsymbol{L}^T\boldsymbol{X} \\ \boldsymbol{X}\boldsymbol{L} & \boldsymbol{Z} \end{pmatrix} \succeq \boldsymbol{O}. \end{array}$$

Here $\boldsymbol{L}$ denotes an $n \times n$ matrix such that $\boldsymbol{Q} = \boldsymbol{L}\boldsymbol{L}^T$. This fact itself never denies the significance of the monotone SDLCP because the direct SDLCP formulation is of a smaller size than the SDP formulation but raises questions like how general the monotone SDLCP is and whether it is essentially different from the SDP. In their recent paper [24], Kojima, Shida, and Shindoh showed that the monotone SDLCP (1) is reducible to an SDP involving an additional m-dimensional variable vector and an $(m+1) \times (m+1)$ variable symmetric matrix, where $m = n(n+1)/2$.

In section 2, we list notation and symbols that are used throughout the paper. Sections 3 through 8 are devoted to our main results:

- The existence of the central trajectory (section 3).
- The existence of modified Newton directions towards the central trajectory (section 4).
- A generic interior-point method (section 5).
- Some properties of the solution set of the monotone SDLCP (1) (section 6).
- Basic lemmas necessary to analyze the computational complexity of interior-point methods for the monotone SDLCP (1) (section 7).
- A central trajectory following method which is an extension of the algorithm given by Kojima–Mizuno–Yoshise [21] for the monotone LCP (8) in the Euclidean space to the monotone SDLCP (1) (section 8.1).

- A potential-reduction method based on the algorithm given by Kojima–Mizuno–Yoshise [22] for the monotone LCP (8) to the monotone SDLCP (1) (section 8.2).
- An infeasible interior-point potential-reduction method based on the constrained potential reduction algorithm given by Mizuno–Kojima–Todd [32, Algorithm I] for LPs to the monotone SDLCP (1) (section 8.3).

2. Notation and symbols.

R^m : the m-dimensional Euclidean space.

$\hat{\mathcal{S}} = R^{n\times n}$, the set of all $n \times n$ matrices.

$\mathcal{S}$: the $n(n+1)/2$-dimensional linear subspace of $\hat{\mathcal{S}}$ consisting of all $n \times n$ symmetric matrices.

$\tilde{\mathcal{S}}$: the $n(n-1)/2$-dimensional linear subspace of $\hat{\mathcal{S}}$ consisting of all $n \times n$ skew-symmetric matrices.

$\mathcal{S}_+ = \{\boldsymbol{X} \in \mathcal{S} : \boldsymbol{X} \succeq \boldsymbol{O}\}$.

$\mathcal{S}_{++} = \{\boldsymbol{X} \in \mathcal{S} : \boldsymbol{X} \succ \boldsymbol{O}\}$.

$\hat{\mathcal{S}}_{++} = \{\boldsymbol{X} \in \hat{\mathcal{S}} : \boldsymbol{X} \succ \boldsymbol{O}\}$.

$\boldsymbol{I}$, $\boldsymbol{O}$: the $n \times n$ identity matrix, the $n \times n$ zero matrix, respectively.

Tr $\boldsymbol{X}$: the trace of $\boldsymbol{X} \in \hat{\mathcal{S}}$.

$\boldsymbol{X} \bullet \boldsymbol{Y} = \mathrm{Tr}\ \boldsymbol{X}^T\boldsymbol{Y}$ for $\boldsymbol{X}, \boldsymbol{Y} \in \hat{\mathcal{S}}$ (the inner product of $\boldsymbol{X}$ and $\boldsymbol{Y}$).

$\|\boldsymbol{X}\|_F = (\boldsymbol{X} \bullet \boldsymbol{X})^{1/2}$ (the Frobenius norm of $\boldsymbol{X} \in \hat{\mathcal{S}}$).

$$\mathcal{F}^0 = \left\{ (\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{S}^2 : \begin{array}{l} (\boldsymbol{X},\boldsymbol{Y}) = \sum_{i=1}^{n(n+1)/2} c_i(\boldsymbol{M}^i, \boldsymbol{N}^i) \\ \text{for some } c_i \in R \ (i = 1, 2, \ldots, n(n+1)/2) \end{array} \right\},$$

where $(\boldsymbol{M}^i, \boldsymbol{N}^i) \in \mathcal{S}^2$ $(i = 1, 2, \ldots, n(n+1)/2)$ are linearly independent.

$\mathcal{F} = (\boldsymbol{X}^0, \boldsymbol{Y}^0) + \mathcal{F}^0$ for some $(\boldsymbol{X}^0, \boldsymbol{Y}^0) \in \mathcal{F}$

(an $n(n+1)/2$-dimensional affine subspace associated with the SDLCP (1)).

$\mathcal{F}_+ = \{(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{F} : \boldsymbol{X} \succeq \boldsymbol{O},\ \boldsymbol{Y} \succeq \boldsymbol{O}\}$

(the set of feasible solutions of the SDLCP (1)).

$\mathcal{F}_{++} = \{(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{F} : \boldsymbol{X} \succ \boldsymbol{O},\ \boldsymbol{Y} \succ \boldsymbol{O}\}$

(the set of interior feasible solutions of the SDLCP (1)).

$\mathcal{F}^* = \{(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{F}_+ : \boldsymbol{X} \bullet \boldsymbol{Y} = 0\}$

(the set of solutions of the SDLCP (1)).

$$\tilde{\mathcal{F}}^0 = \left\{ (\boldsymbol{X},\boldsymbol{Y}) \in \hat{\mathcal{S}}^2 : \begin{array}{l} (\boldsymbol{X},\boldsymbol{Y}) = \sum_{j=1}^{n(n-1)/2} \tilde{c}_j(\tilde{\boldsymbol{M}}^j, \tilde{\boldsymbol{N}}^j) \\ \text{for some } \tilde{c}_j \in R \ (j = 1, 2, \ldots, n(n-1)/2) \end{array} \right\},$$

where $(\tilde{\boldsymbol{M}}^j, \tilde{\boldsymbol{N}}^j) \in \tilde{\mathcal{S}}^2$ $(j = 1, 2, \ldots, n(n-1)/2)$ are linearly independent (an $n(n-1)/2$-dimensional linear subspace of $\tilde{\mathcal{S}}^2$).

$\phi(\mu, \boldsymbol{X}, \boldsymbol{Y}) = \boldsymbol{X} \bullet \boldsymbol{Y} - \mu \log\det \boldsymbol{X}\boldsymbol{Y}$ for every $(\mu, \boldsymbol{X}, \boldsymbol{Y}) \in R_{++} \times \mathcal{S}^2_{++}$

(the logarithmic barrier function).

$\lambda_1, \lambda_2, \ldots, \lambda_n$: the eigenvalues of $\boldsymbol{X}\boldsymbol{Y}$, where $(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{S}^2_{++}$.

(Note that all λ_i's are positive. See below.)

$$\begin{aligned}
\boldsymbol{\Lambda} &= \text{ diag } (\lambda_1, \lambda_2, \dots, \lambda_n). \\
\lambda_{min} &= \min\{\lambda_1, \lambda_2, \dots, \lambda_n\}. \\
\lambda_{max} &= \max\{\lambda_1, \lambda_2, \dots, \lambda_n\}. \\
\boldsymbol{H}(\beta) &= \beta\mu\sqrt{\boldsymbol{X}}^{-1}\sqrt{\boldsymbol{Y}}^{-1} - \sqrt{\boldsymbol{X}}\sqrt{\boldsymbol{Y}} \in \hat{\mathcal{S}}, \quad \text{where } \beta \geq 0 \text{ and } \mu > 0. \\
\mathcal{N}(\gamma) &= \left\{ (\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}_{++} : \left(\sum_{j=1}^{n} (\lambda_j - \mu)^2 \right)^{1/2} \leq \gamma\mu, \quad \text{where } \mu = \frac{\boldsymbol{X} \bullet \boldsymbol{Y}}{n} \right\}, \\
&\qquad \text{where } \gamma > 0 \text{ (a horn neighborhood of the central trajectory).} \\
f(\boldsymbol{X}, \boldsymbol{Y}) &= (n + \nu) \log \boldsymbol{X} \bullet \boldsymbol{Y} - \log\det \boldsymbol{X}\boldsymbol{Y} - n \log n \ \text{ for every } (\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}^2_{++}, \\
&\qquad \text{where } \nu \geq 0 \text{ is a parameter (the potential function).}
\end{aligned}$$

Let $\boldsymbol{X} \in \mathcal{S}_+$. Then we can find a symmetric matrix $\boldsymbol{B}$ such that $\boldsymbol{X} = \boldsymbol{B}\boldsymbol{B}$. Note that such a matrix $\boldsymbol{B}$ is uniquely determined and is positive semidefinite. We denote such a matrix $\boldsymbol{B}$ by $\sqrt{\boldsymbol{X}}$ throughout the paper:

$$\sqrt{\boldsymbol{X}} : \text{ the matrix in } \mathcal{S}_+ \text{ uniquely determined by } \boldsymbol{X} = \sqrt{\boldsymbol{X}}\sqrt{\boldsymbol{X}} \text{ for } \boldsymbol{X} \in \mathcal{S}_+.$$

By the definition, we see that

$$\begin{aligned}
&\text{Tr } \boldsymbol{A} = \sum_{j=1}^{n} \alpha_j, \quad \text{where } \alpha_1, \alpha_2, \dots, \alpha_n \text{ denote the eigenvalues of a matrix } \boldsymbol{A} \in \hat{\mathcal{S}}, \\
&\text{Tr } \boldsymbol{A} = \text{Tr } \boldsymbol{B}^{-1}\boldsymbol{A}\boldsymbol{B} \ \text{ for every } \boldsymbol{A} \in \hat{\mathcal{S}} \ \text{ and every nonsingular } \boldsymbol{B} \in \hat{\mathcal{S}}, \\
&\boldsymbol{M} \bullet \boldsymbol{X} = \boldsymbol{M}^T \bullet \boldsymbol{X} \ \text{ if } \boldsymbol{M} \in \mathcal{S} \ \text{ or } \boldsymbol{X} \in \mathcal{S}.
\end{aligned}$$

We will often use these relations throughout the paper.

The following fact is also utilized often: If $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}^2_{++}$, all the eigenvalues of $\boldsymbol{X}\boldsymbol{Y}$ are real and positive. This is because $\boldsymbol{X}\boldsymbol{Y}$ has the same eigenvalues as the symmetric positive definite matrix $\sqrt{\boldsymbol{X}}\boldsymbol{Y}\sqrt{\boldsymbol{X}}$.

3. The central trajectory. Let $\mathcal{F}_+$, $\mathcal{F}_{++}$, and $\mathcal{F}^*$ denote the set of feasible solutions, the set of interior feasible solutions, and the set of solutions, of the SDLCP (1), respectively:

$$\begin{aligned}
\mathcal{F}_+ &= \{(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F} : \boldsymbol{X} \succeq \boldsymbol{O}, \ \boldsymbol{Y} \succeq \boldsymbol{O}\}, \\
\mathcal{F}_{++} &= \{(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F} : \boldsymbol{X} \succ \boldsymbol{O}, \ \boldsymbol{Y} \succ \boldsymbol{O}\}, \\
\mathcal{F}^* &= \{(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}_+ : \boldsymbol{X} \bullet \boldsymbol{Y} = 0\}.
\end{aligned}$$

THEOREM 3.1. *Suppose that the SDLCP* (1) *has an interior feasible solution, i.e.,* $(\boldsymbol{X}^0, \boldsymbol{Y}^0) \in \mathcal{F}_{++}$.

1. *For every* $\mu > 0$, *there exists a unique* $(\boldsymbol{X}(\mu), \boldsymbol{Y}(\mu)) \in \mathcal{F}_{++}$ *such that* $\boldsymbol{X}(\mu)\boldsymbol{Y}(\mu) = \mu\boldsymbol{I}$, *where* $\boldsymbol{I}$ *denotes the* $n \times n$ *identity matrix.*

2. $(\boldsymbol{X}(\mu), \boldsymbol{Y}(\mu))$ *is the unique minimizer of the logarithmic barrier function*

$$\phi(\mu, \boldsymbol{X}, \boldsymbol{Y}) = \boldsymbol{X} \bullet \boldsymbol{Y} - \mu \log\det \boldsymbol{X}\boldsymbol{Y} \ \textit{ over } \mathcal{F}_{++}.$$

3. *The set* $\mathcal{C} = \{(\boldsymbol{X}(\mu), \boldsymbol{Y}(\mu)) : \mu > 0\}$ *forms a smooth trajectory. (We call* $\mathcal{C}$ *the central trajectory.)*

4. $(\boldsymbol{X}(\mu), \boldsymbol{Y}(\mu))$ *converges to a solution of the SDLCP* (1), $(\boldsymbol{X}^*, \boldsymbol{Y}^*) \in \mathcal{F}^*$ *as* $\mu > 0$ *tends to zero.*

The existence of the central trajectory is known if we restrict ourselves to the primal-dual pair (2) of SDPs $\mathcal{P}$ and $\mathcal{D}$, where $\mathcal{F}$ is given as in (3); see [5, 44, 45].

Besides item 2 of Theorem 3.1, there are some other characterizations of the central trajectory $\mathcal{C}$. One is the following: an $(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{F}_{++}$ lies on the central trajectory $\mathcal{C}$ if and only if all eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ of $\boldsymbol{XY}$ have a common value $\mu > 0$.

We give some remarks on relations of the SDLCP (1) in symmetric matrices with the horizontal LCP (9) (or the LCP (8) with $F = \{(\boldsymbol{x},\boldsymbol{y}) \in R^{2n} : \boldsymbol{y} = \boldsymbol{Mx} + \boldsymbol{q}\}$) in the Euclidean space. Each eigenvalue λ_j of the product $\boldsymbol{XY}$ of a pair of matrices $(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{S}^2_{++}$ plays the role of the product $x_j y_j$ of a complementary pair of variables x_j and y_j in $(\boldsymbol{x},\boldsymbol{y}) \in R^{2n}_{++}$. We have seen above that the central trajectory $\mathcal{C}$ can be rewritten as

$$\mathcal{C} = \left\{ (\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{F}_{++} : \begin{array}{c} \lambda_j = \mu > 0 \ (j = 1, 2, \ldots, n) \ \text{ for some } \mu > 0, \\ \text{where } \lambda_1, \lambda_2, \ldots, \lambda_n \ \text{ are the eigenvalues of } \boldsymbol{XY} \end{array} \right\}.$$

We also see that the logarithmic barrier function ϕ and the potential function f,

$$\begin{aligned} \phi(\mu, \boldsymbol{X}, \boldsymbol{Y}) &= \boldsymbol{X} \bullet \boldsymbol{Y} - \mu \log\det \boldsymbol{XY}, \\ f(\boldsymbol{X}, \boldsymbol{Y}) &= (n + \nu) \log \boldsymbol{X} \bullet \boldsymbol{Y} - \log\det \boldsymbol{XY} - n \log n, \end{aligned}$$

which we will utilize in section 8, can be rewritten in terms of the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ of $\boldsymbol{XY}$ as follows:

$$\begin{aligned} \phi(\mu, \boldsymbol{X}, \boldsymbol{Y}) &= \sum_{j=1}^{n} \lambda_j - \mu \sum_{j=1}^{n} \log \lambda_j, \\ f(\boldsymbol{X}, \boldsymbol{Y}) &= (n + \nu) \log \left(\sum_{j=1}^{n} \lambda_j \right) - \sum_{j=1}^{n} \log \lambda_j - n \log n. \end{aligned}$$

If we replace λ_j by $x_j y_j$, we have the central trajectory, the logarithmic barrier function, and the potential function that have been used widely in interior-point methods for the LCP (8) in the Euclidean space. See also [7] and Figures 2.1 and 3.2 of [2].

We give another characterization of the central trajectory $\mathcal{C}$ in terms of the potential function in section 8.2 where we present a potential-reduction method. See (60).

The remainder of this section is devoted to a proof of Theorem 3.1. Let $\|\boldsymbol{X}\|_F$ denote the Frobenius norm of a matrix $\boldsymbol{X} \in \hat{\mathcal{S}}$; $\|\boldsymbol{X}\|_F^2 = \boldsymbol{X} \bullet \boldsymbol{X}$. Let

$$\mathcal{F}^0 = \{(\boldsymbol{X}' - \boldsymbol{X}, \boldsymbol{Y}' - \boldsymbol{Y}) : (\boldsymbol{X},\boldsymbol{Y}), (\boldsymbol{X}',\boldsymbol{Y}') \in \mathcal{F}\}.$$

Then $\mathcal{F}^0$ forms an $n(n+1)/2$-dimensional linear subspace of $\mathcal{S}^2$. Let $p = n(n+1)/2$, $(\boldsymbol{X}^0, \boldsymbol{Y}^0) \in \mathcal{F}$ and let $(\boldsymbol{M}^i, \boldsymbol{N}^i) (i = 1, 2, \ldots, p)$ be a basis of $\mathcal{F}^0$. Then

$$\mathcal{F} = (\boldsymbol{X}^0, \boldsymbol{Y}^0) + \mathcal{F}^0, \tag{10}$$

$$\mathcal{F}^0 = \left\{ (d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{S}^2 : \begin{array}{l} (d\boldsymbol{X}, d\boldsymbol{Y}) = \sum_{i=1}^{p} c_i (\boldsymbol{M}^i, \boldsymbol{N}^i) \\ \text{for some } c_i \in R \ (i = 1, 2, \ldots, p) \end{array} \right\}. \tag{11}$$

We also note that $\mathcal{F}^0$ is monotone, i.e., $d\boldsymbol{X} \bullet d\boldsymbol{Y} \geq 0$ for every $(d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{F}^0$.

We need a series of lemmas.

LEMMA 3.2.

1. *Suppose that* $(\mu, \boldsymbol{X}, \boldsymbol{Y}) \in R_{++} \times \mathcal{F}_{++}$. *Then*

$$\begin{aligned} &\phi(\mu, \boldsymbol{X} + d\boldsymbol{X}, \boldsymbol{Y} + d\boldsymbol{Y}) - \phi(\mu, \boldsymbol{X}, \boldsymbol{Y}) \\ &\quad = \Phi_1(d\boldsymbol{X}, d\boldsymbol{Y}) \ + \ \Phi_2(d\boldsymbol{X}, d\boldsymbol{Y}) \\ &\qquad + o(\|\sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X} \sqrt{\boldsymbol{X}}^{-1}\|_F^2) + \ o(\|\sqrt{\boldsymbol{Y}}^{-1} d\boldsymbol{Y} \sqrt{\boldsymbol{Y}}^{-1}\|_F^2) \\ &\qquad\qquad\qquad\qquad\qquad\qquad \textit{for all } (d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{F}^0, \end{aligned} \tag{12}$$

where

$$\left\{ \begin{array}{lcl} \Phi_1(d\boldsymbol{X}, d\boldsymbol{Y}) & = & (\boldsymbol{Y} - \mu \boldsymbol{X}^{-1}) \bullet d\boldsymbol{X} + (\boldsymbol{X} - \mu \boldsymbol{Y}^{-1}) \bullet d\boldsymbol{Y}, \\ \Phi_2(d\boldsymbol{X}, d\boldsymbol{Y}) & = & d\boldsymbol{X} \bullet d\boldsymbol{Y} + \dfrac{\mu}{2} \|\sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X} \sqrt{\boldsymbol{X}}^{-1}\|_F^2 \\ & & \qquad + \dfrac{\mu}{2} \|\sqrt{\boldsymbol{Y}}^{-1} d\boldsymbol{Y} \sqrt{\boldsymbol{Y}}^{-1}\|_F^2. \end{array} \right. \tag{13}$$

2. $\phi(\mu, \cdot)$ *is strictly convex on* $\mathcal{F}_{++}$.

Proof. 1. Let $(d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{F}^0$, $\xi_1, \xi_2, \ldots, \xi_n$ be the eigenvalues of the symmetric matrix $\sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X} \sqrt{\boldsymbol{X}}^{-1}$, and $\eta_1, \eta_2, \ldots, \eta_n$ be the eigenvalues of the symmetric matrix $\sqrt{\boldsymbol{Y}}^{-1} d\boldsymbol{Y} \sqrt{\boldsymbol{Y}}^{-1}$. It suffices to derive (12) under the assumption that $(d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{F}^0$ is so small that the absolute values of all eigenvalues $\xi_1, \xi_2, \ldots, \xi_n$, $\eta_1, \eta_2, \ldots, \eta_n$ are less than one. The assumption ensures that $(\boldsymbol{X} + d\boldsymbol{X}, \boldsymbol{Y} + d\boldsymbol{Y}) \in \mathcal{F}_{++}$. We then see that

$$\begin{aligned} &\phi(\mu, \boldsymbol{X} + d\boldsymbol{X}, \boldsymbol{Y} + d\boldsymbol{Y}) \\ &\quad = (\boldsymbol{X} + d\boldsymbol{X}) \bullet (\boldsymbol{Y} + d\boldsymbol{Y}) - \mu \log\det(\boldsymbol{X} + d\boldsymbol{X}) - \mu \log\det(\boldsymbol{Y} + d\boldsymbol{Y}) \\ &\quad = \boldsymbol{X} \bullet \boldsymbol{Y} + d\boldsymbol{X} \bullet \boldsymbol{Y} + \boldsymbol{X} \bullet d\boldsymbol{Y} + d\boldsymbol{X} \bullet d\boldsymbol{Y} \\ &\qquad -\mu \left(\log\det \boldsymbol{X} + \log \prod_{j=1}^{n} (1 + \xi_j) \right) - \mu \left(\log\det \boldsymbol{Y} + \log \prod_{j=1}^{n} (1 + \eta_j) \right) \\ &\quad = \phi(\mu, \boldsymbol{X}, \boldsymbol{Y}) + d\boldsymbol{X} \bullet \boldsymbol{Y} + \boldsymbol{X} \bullet d\boldsymbol{Y} + d\boldsymbol{X} \bullet d\boldsymbol{Y} \\ &\qquad -\mu \left(\sum_{j=1}^{n} \xi_j - \frac{1}{2} \sum_{j=1}^{n} \xi_j^2 + o\left(\sum_{j=1}^{n} \xi_j^2 \right) \right) - \mu \left(\sum_{j=1}^{n} \eta_j - \frac{1}{2} \sum_{j=1}^{n} \eta_j^2 + o\left(\sum_{j=1}^{n} \eta_j^2 \right) \right) \\ &\quad = \phi(\mu, \boldsymbol{X}, \boldsymbol{Y}) + \Phi_1(d\boldsymbol{X}, d\boldsymbol{Y}) + \Phi_2(d\boldsymbol{X}, d\boldsymbol{Y}) \\ &\qquad +o\left(\|\sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X} \sqrt{\boldsymbol{X}}^{-1}\|_F^2 \right) + o\left(\|\sqrt{\boldsymbol{Y}}^{-1} d\boldsymbol{Y} \sqrt{\boldsymbol{Y}}^{-1}\|_F^2 \right). \end{aligned}$$

Thus we have shown assertion 1.

2. Note that the quadratic form $\Phi_2(d\boldsymbol{X}, d\boldsymbol{Y})$ is positive for any nonzero $(d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{F}^0$. This ensures that $\phi(\mu, \cdot)$ is strictly convex on $\mathcal{F}_{++}$. □

LEMMA 3.3. *Let* $\mu \in R_{++}$.

1. *If* $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}_{++}$ *satisfies the condition* $\boldsymbol{XY} = \mu \boldsymbol{I}$, *then* $(\boldsymbol{X}, \boldsymbol{Y})$ *is a global minimizer of* $\phi(\mu, \cdot)$ *over* $\mathcal{F}_{++}$.

2. *Assume that there is an* $(\boldsymbol{X}^0, \boldsymbol{Y}^0) \in \mathcal{F}_{++}$. *Then there is a unique global minimizer of* $\phi(\mu, \cdot)$ *over* $\mathcal{F}_{++}$.

Proof. Let λ_j $(j = 1, 2, \cdots, n)$ denote the n positive eigenvalues of the symmetric positive definite matrix $\sqrt{\boldsymbol{X}} \boldsymbol{Y} \sqrt{\boldsymbol{X}}$. Then

$$\phi(\mu, \boldsymbol{X}, \boldsymbol{Y}) = \boldsymbol{X} \bullet \boldsymbol{Y} - \mu \log\det \boldsymbol{XY} = \sum_{j=1}^{n} (\lambda_j - \mu \log \lambda_j).$$

Hence we have shown that

$$\phi(\mu, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{j=1}^{n} (\lambda_j - \mu \log \lambda_j) . \tag{14}$$

1. Note that each term $\lambda_j - \mu \log \lambda_j$ in the parentheses ($\cdot$) attains the minimum under the condition $\lambda_j > 0$ if and only if $\lambda_j = \mu$. On the other hand, we can rewrite the condition $\boldsymbol{XY} = \mu \boldsymbol{I}$ as $\sqrt{\boldsymbol{X}}\boldsymbol{Y}\sqrt{\boldsymbol{X}} = \mu\boldsymbol{I}$ or equivalently $\lambda_j = \mu$ $(j = 1, 2, \ldots, n)$. Thus assertion 1 follows.

2. Take a real number θ such that $\phi(\mu, \boldsymbol{X}^0, \boldsymbol{Y}^0) \leq \theta$. We show that the level set

$$\Gamma = \{(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}_{++} : \phi(\mu, \boldsymbol{X}, \boldsymbol{Y}) \leq \theta\}$$

of the function $\phi(\mu, \cdot)$ is a bounded and closed subset of $\mathcal{S}^2$. Then assertion 2 follows from the continuity and the strict convexity of the function $\phi(\mu, \cdot)$ over the level set Γ. We first show that the level set is contained in the bounded set

$$\Gamma^* = \{\boldsymbol{X} \in \mathcal{S}_+ : \boldsymbol{Y}^0 \bullet \boldsymbol{X} \leq \gamma\} \times \{\boldsymbol{Y} \in \mathcal{S}_+ : \boldsymbol{X}^0 \bullet \boldsymbol{Y} \leq \gamma\},$$

where

$$\gamma = 2n \left(\theta - n(\mu - \mu \log \mu) + \mu \log 2\right) + \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0.$$

Assume that $(\boldsymbol{X}, \boldsymbol{Y}) \in \Gamma$. Let $j \in \{1, 2, \ldots, n\}$ be fixed. We see from $(\boldsymbol{X}, \boldsymbol{Y}) \in \Gamma$ and (14) that

$$\begin{aligned} \theta &\geq \phi(\mu, \boldsymbol{X}, \boldsymbol{Y}) \\ &= \sum_{j=1}^{n} (\lambda_j - \mu \log \lambda_j) \\ &\geq (n-1)(\mu - \mu \log \mu) + \lambda_j - \mu \log \lambda_j \\ &\geq n(\mu - \mu \log \mu) - \mu \log 2 + \lambda_j / 2. \end{aligned}$$

Hence all the positive eigenvalues λ_j $(j = 1, 2, \ldots, n)$ of the symmetric positive definite matrix $\sqrt{\boldsymbol{X}}\boldsymbol{Y}\sqrt{\boldsymbol{X}}$ are bounded from above by the number

$$\gamma' = 2 \left(\theta - n(\mu - \mu \log \mu) + \mu \log 2\right).$$

This implies that $\boldsymbol{X} \bullet \boldsymbol{Y} = \mathrm{Tr}\ \sqrt{\boldsymbol{X}}\boldsymbol{Y}\sqrt{\boldsymbol{X}} = \sum_{j=1}^{n} \lambda_j \leq n\gamma'$. On the other hand, we see by Condition 1.2 that $\boldsymbol{X} \bullet \boldsymbol{Y} + \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0 \geq \boldsymbol{X}^0 \bullet \boldsymbol{Y} + \boldsymbol{Y}^0 \bullet \boldsymbol{X}$. Hence $(\boldsymbol{X}, \boldsymbol{Y})$ satisfies that $\boldsymbol{Y}^0 \bullet \boldsymbol{X} \leq \gamma$ and $\boldsymbol{X}^0 \bullet \boldsymbol{Y} \leq \gamma$. See Lemma 1.1 for $\boldsymbol{Y}^0 \bullet \boldsymbol{X} \geq 0$, $\boldsymbol{X}^0 \bullet \boldsymbol{Y} \geq 0$ and the boundedness of the set Γ^*. Thus we have shown that the level set Γ is contained in the bounded set Γ^*.

Now we will prove that the level set Γ is closed. Let $\{(\boldsymbol{X}^k, \boldsymbol{Y}^k)\} \subset \Gamma$ be a sequence converging to some $(\boldsymbol{X}^*, \boldsymbol{Y}^*) \in \mathcal{S}^2$. It suffices to show that $(\boldsymbol{X}^*, \boldsymbol{Y}^*) \in \mathcal{S}^2_{++}$. Since the sequence is bounded, there is a positive number δ such that

$$\log \det \boldsymbol{X}^k \leq \delta \ \text{ and } \ \log \det \boldsymbol{Y}^k \leq \delta \ \text{ for every } k = 1, 2, \ldots \ .$$

It follows that for every $k = 1, 2, \ldots,$

$$\begin{aligned} \mu \log \det \boldsymbol{X}^k &= -\phi(\mu, \boldsymbol{X}^k, \boldsymbol{Y}^k) + \boldsymbol{X}^k \bullet \boldsymbol{Y}^k - \mu \log \det \boldsymbol{Y}^k \geq -\theta - \mu\delta, \\ \mu \log \det \boldsymbol{Y}^k &= -\phi(\mu, \boldsymbol{X}^k, \boldsymbol{Y}^k) + \boldsymbol{X}^k \bullet \boldsymbol{Y}^k - \mu \log \det \boldsymbol{X}^k \geq -\theta - \mu\delta; \end{aligned}$$

hence

$$\left.\begin{array}{ll} \boldsymbol{X}^k \in \mathcal{S}_{++}, & \det \boldsymbol{X}^k \geq \exp\left((-\theta - \mu\delta)/\mu\right) > 0 \\ \boldsymbol{Y}^k \in \mathcal{S}_{++}, & \det \boldsymbol{Y}^k \geq \exp\left((-\theta - \mu\delta)/\mu\right) > 0 \end{array}\right\} \text{ for every } k = 1, 2, \ldots .$$

This ensures that $(\boldsymbol{X}^*, \boldsymbol{Y}^*) \in \mathcal{S}^2_{++}$. Thus we have shown that the level set Γ is closed. □

LEMMA 3.4. *Let L be an m-dimensional monotone linear subspace of R^{2m}. Then its orthogonal complement $L^\perp = \{(\boldsymbol{a}, \boldsymbol{b}) \in R^{2m} : \boldsymbol{a}^T\boldsymbol{u} + \boldsymbol{b}^T\boldsymbol{v} = 0$ for every $(\boldsymbol{u}, \boldsymbol{v}) \in L\}$ is antitone, i.e., $\boldsymbol{a} \bullet \boldsymbol{b} \leq 0$ for every $(\boldsymbol{a}, \boldsymbol{b}) \in L^\perp$.*

Proof. We represent the m-dimensional monotone linear space L as

$$L = \left\{ (\boldsymbol{u}, \boldsymbol{v}) \in R^{2m} : \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}^T \\ -\boldsymbol{B}^T \end{pmatrix} \boldsymbol{z},\ \boldsymbol{z} \in R^m \right\},$$

where $\boldsymbol{A}$, $\boldsymbol{B}$ are $m \times m$ matrices such that rank $(\boldsymbol{A}, -\boldsymbol{B}) = m$. We can easily verify that $L^\perp = \{(\boldsymbol{a}, \boldsymbol{b}) \in R^{2m} : \boldsymbol{A}\boldsymbol{a} - \boldsymbol{B}\boldsymbol{b} = \boldsymbol{0}\}$. By the monotonicity, $\boldsymbol{B}\boldsymbol{A}^T$ is negative semidefinite. Consider the quadratic form

$$\boldsymbol{u}^T(\boldsymbol{A} - \boldsymbol{B})(\boldsymbol{A} - \boldsymbol{B})^T\boldsymbol{u} = \boldsymbol{u}^T\left((\boldsymbol{A}, -\boldsymbol{B})(\boldsymbol{A}, -\boldsymbol{B})^T - 2\boldsymbol{B}\boldsymbol{A}^T\right)\boldsymbol{u} \text{ for every } \boldsymbol{u} \in R^m.$$

The right-hand side is positive for any nonzero $\boldsymbol{u} \in R^m$ because the $m \times 2m$ matrix $(\boldsymbol{A},\ -\boldsymbol{B})$ is of full row rank, i.e., rank $(\boldsymbol{A}, -\boldsymbol{B}) = m$, and $-\boldsymbol{B}\boldsymbol{A}^T$ is positive semidefinite. Hence so is the left-hand side. This ensures rank $(\boldsymbol{A} - \boldsymbol{B}) = m$. By Theorem 11 of Sznajder–Gowda [39], we obtain that the m-dimensional subspace $\{(\boldsymbol{a}, \boldsymbol{b}) \in R^{2m} : \boldsymbol{A}\boldsymbol{a} + \boldsymbol{B}\boldsymbol{b} = \boldsymbol{0}\}$ is monotone, which implies that the orthogonal complement $L^\perp = \{(\boldsymbol{a}, \boldsymbol{b}) \in R^{2m} : \boldsymbol{A}\boldsymbol{a} - \boldsymbol{B}\boldsymbol{b} = \boldsymbol{0}\}$ of L is antitone. □

LEMMA 3.5. *Let $\mu \in R_{++}$ be fixed. Suppose that $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}_{++}$ is the global minimizer of $\phi(\mu, \cdot)$ over $\mathcal{F}_{++}$. Then it satisfies $\boldsymbol{X}\boldsymbol{Y} = \mu\boldsymbol{I}$.*

Proof. Recall the relations (12) and (13) in Lemma 3.2. At the global minimizer $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}_{++}$, the linear term $\Phi_1(d\boldsymbol{X}, d\boldsymbol{Y})$ with respect to $(d\boldsymbol{X}, d\boldsymbol{Y})$ in (12) must satisfy the equality

$$\Phi_1(d\boldsymbol{X}, d\boldsymbol{Y}) = (\boldsymbol{Y} - \mu\boldsymbol{X}^{-1}) \bullet d\boldsymbol{X} + (\boldsymbol{X} - \mu\boldsymbol{Y}^{-1}) \bullet d\boldsymbol{Y} = 0 \quad \text{for every } (d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{F}^0.$$

Hence $(\boldsymbol{Y} - \mu\boldsymbol{X}^{-1}, \boldsymbol{X} - \mu\boldsymbol{Y}^{-1}) \in \mathcal{S}^2$ lies in the orthogonal complement of the $n(n+1)/2$-dimensional monotone linear subspace $\mathcal{F}^0$. By Lemma 3.4,

$$0 \geq (\boldsymbol{Y} - \mu\boldsymbol{X}^{-1}) \bullet (\boldsymbol{X} - \mu\boldsymbol{Y}^{-1}) = \text{Tr }\left(\boldsymbol{X}\boldsymbol{Y} - 2\mu\boldsymbol{I} + \mu^2\boldsymbol{Y}^{-1}\boldsymbol{X}^{-1}\right).$$

Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ denote the eigenvalues of the symmetric positive definite matrix $\sqrt{\boldsymbol{X}}\boldsymbol{Y}\sqrt{\boldsymbol{X}}$. Then they are all real and positive. It follows from the inequality above that

$$0 \geq \text{Tr }\left(\boldsymbol{X}\boldsymbol{Y} - 2\mu\boldsymbol{I} + \mu^2\boldsymbol{Y}^{-1}\boldsymbol{X}^{-1}\right) = \sum_{j=1}^{n} \frac{(\lambda_j - \mu)^2}{\lambda_j}.$$

Hence all the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ of the symmetric matrix $\sqrt{\boldsymbol{X}}\boldsymbol{Y}\sqrt{\boldsymbol{X}}$ are equal to μ. This implies that $\sqrt{\boldsymbol{X}}\boldsymbol{Y}\sqrt{\boldsymbol{X}} = \mu\boldsymbol{I}$. Therefore $\boldsymbol{X}\boldsymbol{Y} = \sqrt{\boldsymbol{X}}(\sqrt{\boldsymbol{X}}\boldsymbol{Y}\sqrt{\boldsymbol{X}})\sqrt{\boldsymbol{X}}^{-1} = \mu\boldsymbol{I}$. □

Assertions 1 and 2 of Theorem 3.1 follow from Lemmas 3.3 and 3.5. To prove assertion 3 of Theorem 3.1, define a mapping $\boldsymbol{H} : R \times R^p \to \mathcal{S} = R^p$ by

$$\boldsymbol{H}(\mu, \boldsymbol{c}) = \left(\boldsymbol{X}^0 + \sum_{i=1}^{p} c_i \boldsymbol{M}^i\right)\left(\boldsymbol{Y}^0 + \sum_{i=1}^{p} c_i \boldsymbol{N}^i\right) - \mu \boldsymbol{I} \text{ for every } (\mu, \boldsymbol{c}) \in R^{1+p}.$$

Here $(\boldsymbol{X}^0, \boldsymbol{Y}^0) \in \mathcal{F}$ and $\{(\boldsymbol{M}^i, \boldsymbol{N}^i) \in \mathcal{S}^2 \ (i = 1, 2, \ldots, p)\}$ denote a basis of $\mathcal{F}^0$ (see (11)). Then each point $(\boldsymbol{X}, \boldsymbol{Y}) = (\boldsymbol{X}(\mu), \boldsymbol{Y}(\mu)) \in \mathcal{S}^2_{++}$ on the central trajectory is characterized as a unique solution of the system of equations

$$(\boldsymbol{X}, \boldsymbol{Y}) \;=\; \left(\boldsymbol{X}^0 + \sum_{i=1}^{p} c_i \boldsymbol{M}^i, \boldsymbol{Y}^0 + \sum_{i=1}^{p} c_i \boldsymbol{N}^i\right) \text{ and } \boldsymbol{H}(\mu, \boldsymbol{c}) \;=\; \boldsymbol{O}.$$

LEMMA 3.6. *Let $\mu \in R_{++}$ and $(\boldsymbol{X}, \boldsymbol{Y}) = (\boldsymbol{X}^0 + \sum_{i=1}^{p} c_i^0 \boldsymbol{M}^i, \boldsymbol{Y}^0 + \sum_{i=1}^{p} c_i^0 \boldsymbol{N}^i) \in \mathcal{F}_{++}$. Then the Jacobian matrix of the mapping $\boldsymbol{H}(\mu, \cdot)$ with respect to $\boldsymbol{c} \in R^p$ at $\boldsymbol{c}^0$ is nonsingular.*

Proof. It suffices to show that the system of linear equations

$$\left.\frac{d\boldsymbol{H}(\mu, \boldsymbol{c}^0 + td\boldsymbol{c})}{dt}\right|_{t=0} = \boldsymbol{O} \tag{15}$$

has no nonzero solution $d\boldsymbol{c} \in R^p$. A simple calculation shows that

$$\left.\frac{d\boldsymbol{H}(\mu, \boldsymbol{c}^0 + td\boldsymbol{c})}{dt}\right|_{t=0} = \boldsymbol{X}\left(\sum_{i=1}^{p} dc_i \boldsymbol{N}^i\right) + \left(\sum_{i=1}^{p} dc_i \boldsymbol{M}^i\right)\boldsymbol{Y}.$$

Let $d\boldsymbol{c} \in R^p$ be a solution of (15) and $(d\boldsymbol{X}, d\boldsymbol{Y}) = \sum_{i=1}^{p} dc_i (\boldsymbol{M}^i, \boldsymbol{N}^i) \in \mathcal{F}^0$. Then $\boldsymbol{X} d\boldsymbol{Y} + d\boldsymbol{X}\boldsymbol{Y} = \boldsymbol{O}$. By the monotonicity of the linear subspace $\mathcal{F}^0$ and $(d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{F}^0$ we see that $d\boldsymbol{X} \bullet d\boldsymbol{Y} \geq 0$. We also see that $\boldsymbol{O} = \boldsymbol{X} d\boldsymbol{Y} + d\boldsymbol{X}\boldsymbol{Y} = \sqrt{\boldsymbol{X}}\sqrt{\boldsymbol{X}} d\boldsymbol{Y} + d\boldsymbol{X}\boldsymbol{Y}$; hence

$$\begin{aligned}\boldsymbol{O} &= \sqrt{\boldsymbol{X}} d\boldsymbol{Y} \sqrt{\boldsymbol{X}} + \sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X}\boldsymbol{Y}\sqrt{\boldsymbol{X}} \\ &= \sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X} d\boldsymbol{Y}\sqrt{\boldsymbol{X}} + \sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X}\sqrt{\boldsymbol{X}}^{-1}\sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X}\boldsymbol{Y}\sqrt{\boldsymbol{X}}.\end{aligned}$$

Thus

$$\begin{aligned}0 &= \mathrm{Tr}\ \left(\sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X} d\boldsymbol{Y}\sqrt{\boldsymbol{X}} + \sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X}\sqrt{\boldsymbol{X}}^{-1}\sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X}\boldsymbol{Y}\sqrt{\boldsymbol{X}}\right) \\ &= d\boldsymbol{X} \bullet d\boldsymbol{Y} + \mathrm{Tr}\ d\boldsymbol{X}\sqrt{\boldsymbol{X}}^{-1}\sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X}\boldsymbol{Y} \\ &\geq \mathrm{Tr}\ \sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X}\boldsymbol{Y} d\boldsymbol{X}\sqrt{\boldsymbol{X}}^{-1}.\end{aligned}$$

Since the matrix $\boldsymbol{Y} \in \mathcal{S}$ is positive definite, the inequality above implies that every column of the matrix $d\boldsymbol{X}\sqrt{\boldsymbol{X}}^{-1}$ is zero. Hence $(\boldsymbol{O}, \boldsymbol{O}) = (d\boldsymbol{X}, d\boldsymbol{Y}) = \sum_{i=1}^{p} dc_i (\boldsymbol{M}^i, \boldsymbol{N}^i)$. Recall that $\{(\boldsymbol{M}^i, \boldsymbol{N}^i) \in \mathcal{S}^2 \ (i = 1, 2, \ldots, p)\}$ is a basis of $\mathcal{F}^0$. Therefore we obtain $d\boldsymbol{c} = \boldsymbol{0}$. □

Obviously, the mapping $\boldsymbol{H}$ is C^∞ on $R \times R^p$. Applying the implicit function theorem (see, for example, [14]), we obtain assertion 3 of Theorem 3.1.

LEMMA 3.7. *For every $\bar{\mu} > 0$, the subset $\{(\boldsymbol{X}(\mu), \boldsymbol{Y}(\mu)) : 0 < \mu \leq \bar{\mu}\}$ of the central trajectory is bounded.*

Proof. Let $0 < \mu \leq \bar{\mu}$. By Condition 1.2, $(\boldsymbol{X}^0, \boldsymbol{Y}^0) \in \mathcal{F}_{++}$, and $(\boldsymbol{X}(\mu), \boldsymbol{Y}(\mu)) \in \mathcal{F}_{++}$, we see that $(\boldsymbol{X}(\mu) - \boldsymbol{X}^0) \bullet (\boldsymbol{Y}(\mu) - \boldsymbol{Y}^0) \geq 0$. It follows that

$$\begin{aligned} \boldsymbol{Y}^0 \bullet \boldsymbol{X}(\mu) + \boldsymbol{X}^0 \bullet \boldsymbol{Y}(\mu) &\leq \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0 + \boldsymbol{X}(\mu) \bullet \boldsymbol{Y}(\mu) \\ &\leq \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0 + n\mu \\ &\leq \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0 + n\bar{\mu}. \end{aligned}$$

Thus the subset $\{(\boldsymbol{X}(\mu), \boldsymbol{Y}(\mu)) : 0 < \mu \leq \bar{\mu}\}$ is contained in the bounded set

$$\{\boldsymbol{X} \in \mathcal{S}_+ : \boldsymbol{Y}^0 \bullet \boldsymbol{X} \leq \gamma\} \times \{\boldsymbol{Y} \in \mathcal{S}_+ : \boldsymbol{X}^0 \bullet \boldsymbol{Y} \leq \gamma\},$$

where $\gamma = \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0 + n\bar{\mu}$. □

In view of the lemma above, there exists at least one accumulation point of $(\boldsymbol{X}(\mu), \boldsymbol{Y}(\mu))$ as $\mu > 0$ tends to 0. By the continuity, every accumulation point is a solution of the SDLCP (1). The convergence of $(\boldsymbol{X}(\mu), \boldsymbol{Y}(\mu))$ to a single point as $\mu > 0$ tends to 0 follows from the fact that the central trajectory $\mathcal{C}$ is characterized as the algebraic system of equations $\boldsymbol{H}(\mu, \boldsymbol{c}) = \boldsymbol{O}$. The details are omitted here. See Theorem 4.4 of Kojima–Megiddo–Noma–Yoshise [19]. This completes the proof of Theorem 3.1.

4. Newton directions toward the central trajectory. Let $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}^2_{++}$ and $\mu = \boldsymbol{X} \bullet \boldsymbol{Y}/n$. Choose $\beta \geq 0$. It might seem natural to regard the system of linear equations

$$(\boldsymbol{X} + d\boldsymbol{X}, \boldsymbol{Y} + d\boldsymbol{Y}) \in \mathcal{F} \text{ and } d\boldsymbol{X}\boldsymbol{Y} + \boldsymbol{X}d\boldsymbol{Y} = \boldsymbol{Q} \tag{16}$$

in variable matrices $d\boldsymbol{X}$, $d\boldsymbol{Y} \in \mathcal{S}$ as the Newton equation at $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}^2_{++}$ for approximating a point $(\boldsymbol{X}', \boldsymbol{Y}') = (\boldsymbol{X} + d\boldsymbol{X}, \boldsymbol{Y} + d\boldsymbol{Y}) \in \mathcal{S}^2_{++}$ on the central trajectory that satisfies

$$(\boldsymbol{X}', \boldsymbol{Y}') \in \mathcal{F} \text{ and } \boldsymbol{X}'\boldsymbol{Y}' = \beta\mu\boldsymbol{I}. \tag{17}$$

Here $\boldsymbol{Q} = \beta\mu\boldsymbol{I} - \boldsymbol{X}\boldsymbol{Y}$. However the system (16) does not necessarily have a solution. Hence we need a suitable modification in the system (16) to consistently define Newton directions toward the central trajectory. For this purpose, we introduce an $n(n-1)/2$-dimensional linear subspace $\tilde{\mathcal{F}}^0$ of $\tilde{\mathcal{S}}^2$, where $\tilde{\mathcal{S}} \subset \hat{\mathcal{S}}$ is the $n(n-1)/2$-dimensional linear subspace consisting of all $n \times n$ skew-symmetric matrices. It should be noted that $\mathcal{S}$ and $\tilde{\mathcal{S}}$ are orthogonal complements to each other in the linear space $\hat{\mathcal{S}}$. Since $\mathcal{F}^0 \subset \mathcal{S}^2$ and $\tilde{\mathcal{F}}^0 \subset \tilde{\mathcal{S}}^2$,

$$\begin{aligned} d\boldsymbol{X} \bullet d\tilde{\boldsymbol{X}} \ = \ d\boldsymbol{Y} \bullet d\tilde{\boldsymbol{Y}} \ = \ d\boldsymbol{X} \bullet d\tilde{\boldsymbol{Y}} \ = \ d\boldsymbol{Y} \bullet d\tilde{\boldsymbol{X}} \ = \ 0 \\ \text{for every } (d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{F}^0 \text{ and } (d\tilde{\boldsymbol{X}}, d\tilde{\boldsymbol{Y}}) \in \tilde{\mathcal{F}}^0. \end{aligned} \tag{18}$$

We impose $\tilde{\mathcal{F}}^0$ on the condition below.

Condition 4.1. $\tilde{\mathcal{F}}^0$ is monotone, i.e., $d\tilde{\boldsymbol{X}} \bullet d\tilde{\boldsymbol{Y}} \geq 0$ for every $(d\tilde{\boldsymbol{X}}, d\tilde{\boldsymbol{Y}}) \in \tilde{\mathcal{F}}^0$. For example, we can take $\tilde{\mathcal{F}}^0 = \{(t\boldsymbol{W}, (1-t)\boldsymbol{W}) : \boldsymbol{W} \in \tilde{\mathcal{S}}\}$, where $t \in [0,1]$ is an arbitrary constant.

Now we consider a (modified) Newton equation at $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}^2_{++}$ for approximating a point $(\boldsymbol{X}', \boldsymbol{Y}') = (\boldsymbol{X} + d\boldsymbol{X}, \boldsymbol{Y} + d\boldsymbol{Y})$ on the central trajectory which satisfies (17):

$$\left\{ \begin{array}{l} (\boldsymbol{X} + d\boldsymbol{X}, \boldsymbol{Y} + d\boldsymbol{Y}) \in \mathcal{F}, \ (d\tilde{\boldsymbol{X}}, d\tilde{\boldsymbol{Y}}) \in \tilde{\mathcal{F}}^0, \ \text{ and} \\ \boldsymbol{X}(d\boldsymbol{Y} + d\tilde{\boldsymbol{Y}}) + (d\boldsymbol{X} + d\tilde{\boldsymbol{X}})\boldsymbol{Y} = \boldsymbol{Q} \end{array} \right. \tag{19}$$

in variable matrices $d\boldsymbol{X}, d\boldsymbol{Y} \in \mathcal{S}$, and $\tilde{d\boldsymbol{X}}, \tilde{d\boldsymbol{Y}} \in \tilde{\mathcal{S}}$. Here $\boldsymbol{Q} = \beta\mu\boldsymbol{I} - \boldsymbol{X}\boldsymbol{Y}$.

THEOREM 4.2. *Let* $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}^2_{++}$, $\mu = \boldsymbol{X} \bullet \boldsymbol{Y}/n$, *and* $\beta \geq 0$. *Then the Newton equation* (19) *has a unique solution* $(d\boldsymbol{X}, d\boldsymbol{Y}, \tilde{d\boldsymbol{X}}, \tilde{d\boldsymbol{Y}}) \in \mathcal{S}^2 \times \tilde{\mathcal{S}}^2$.

It should be noted that Theorem 4.2 is valid even when the feasible region $\mathcal{F}_+$ of the SDLCP (1) in symmetric matrices is empty or the central trajectory $\mathcal{C}$ does not exist.

Proof of Theorem 4.2. Let $\{(\boldsymbol{M}^i, \boldsymbol{N}^i) \in \mathcal{S}^2 \ (i = 1, 2, \ldots, p)\}$ be a basis of $\mathcal{F}^0$ and $\{(\tilde{\boldsymbol{M}}^j, \tilde{\boldsymbol{N}}^j) \in \tilde{\mathcal{S}}^2 \ (j = 1, 2, \ldots, \tilde{p})\}$ be a basis of $\tilde{\mathcal{F}}^0$, and let $(\boldsymbol{X}^0, \boldsymbol{Y}^0) \in \mathcal{F}$, where $p = n(n+1)/2$ and $\tilde{p} = n(n-1)/2$. Note that $p + \tilde{p} = n^2$. Then the first relation of the Newton equation (19) can be written as $(\boldsymbol{X} + d\boldsymbol{X}, \boldsymbol{Y} + d\boldsymbol{Y}) = (\boldsymbol{X}^0, \boldsymbol{Y}^0) + \sum_{i=1}^p c_i(\boldsymbol{M}^i, \boldsymbol{N}^i)$, hence

$$d\boldsymbol{X} \ = \ \boldsymbol{X}^0 - \boldsymbol{X} + \sum_{i=1}^p c_i \boldsymbol{M}^i \ \text{ and } \ d\boldsymbol{Y} \ = \ \boldsymbol{Y}^0 - \boldsymbol{Y} + \sum_{i=1}^p c_i \boldsymbol{N}^i,$$

where c_i $(i = 1, 2, \ldots, p)$ are real variables. With new variables $\tilde{c}_j$ $(j = 1, 2, \ldots, \tilde{p})$, we also rewrite the second relation of (19) as $(\tilde{d\boldsymbol{X}}, \tilde{d\boldsymbol{Y}}) = \sum_{j=1}^{\tilde{p}} \tilde{c}_j(\tilde{\boldsymbol{M}}^j, \tilde{\boldsymbol{N}}^j)$. Now the last equation in (19) is reduced to

$$\sum_{i=1}^p c_i(\boldsymbol{X}\boldsymbol{N}^i + \boldsymbol{M}^i\boldsymbol{Y}) + \sum_{j=1}^{\tilde{p}} \tilde{c}_j(\boldsymbol{X}\tilde{\boldsymbol{N}}^j + \tilde{\boldsymbol{M}}^j\boldsymbol{Y}) = \boldsymbol{Q} - \boldsymbol{X}(\boldsymbol{Y}^0 - \boldsymbol{Y}) - (\boldsymbol{X}^0 - \boldsymbol{X})\boldsymbol{Y}.$$

Hence we have only to show that the equation above in n^2 variables c_i $(i = 1, 2, \ldots, p)$ and $\tilde{c}_j$ $(j = 1, 2, \ldots, \tilde{p})$ has a unique solution. It suffices to show that the set of n^2 matrices

$$(20) \quad (\boldsymbol{X}\boldsymbol{N}^i + \boldsymbol{M}^i\boldsymbol{Y}) \ \ (i = 1, 2, \ldots, p) \quad \text{and} \quad (\boldsymbol{X}\tilde{\boldsymbol{N}}^j + \tilde{\boldsymbol{M}}^j\boldsymbol{Y}) \ \ (j = 1, 2, \ldots, \tilde{p})$$

forms a basis of the n^2-dimensional linear space $\hat{\mathcal{S}}$. Assume that

$$(21) \qquad \sum_{i=1}^p c'_i(\boldsymbol{X}\boldsymbol{N}^i + \boldsymbol{M}^i\boldsymbol{Y}) + \sum_{j=1}^{\tilde{p}} \tilde{c}'_j(\boldsymbol{X}\tilde{\boldsymbol{N}}^j + \tilde{\boldsymbol{M}}^j\boldsymbol{Y}) = \boldsymbol{O}$$

for some c'_i $(i = 1, 2, \ldots, p)$ and $\tilde{c}'_j$ $(j = 1, 2, \ldots, \tilde{p})$. Let

$$d\boldsymbol{X}' = \sum_{i=1}^p c'_i \boldsymbol{M}^i, \ d\boldsymbol{Y}' = \sum_{i=1}^p c'_i \boldsymbol{N}^i, \ \tilde{d\boldsymbol{X}}' = \sum_{j=1}^{\tilde{p}} \tilde{c}'_j \tilde{\boldsymbol{M}}^j, \text{ and } \tilde{d\boldsymbol{Y}}' = \sum_{j=1}^{\tilde{p}} \tilde{c}'_j \tilde{\boldsymbol{N}}^j.$$

Then $(d\boldsymbol{X}', d\boldsymbol{Y}') \in \mathcal{F}^0$ and $(\tilde{d\boldsymbol{X}}', \tilde{d\boldsymbol{Y}}') \in \tilde{\mathcal{F}}^0$. We also see from (21) that

$$(22) \qquad \boldsymbol{O} = \boldsymbol{X}(d\boldsymbol{Y}' + \tilde{d\boldsymbol{Y}}') + (d\boldsymbol{X}' + \tilde{d\boldsymbol{X}}')\boldsymbol{Y}.$$

Since $\boldsymbol{X}$ and $\boldsymbol{Y}$ are positive definite, it follows from (22) that

$$\boldsymbol{O} = \sqrt{\boldsymbol{X}}(d\boldsymbol{Y}' + \tilde{d\boldsymbol{Y}}')\sqrt{\boldsymbol{Y}}^{-1} + \sqrt{\boldsymbol{X}}^{-1}(d\boldsymbol{X}' + \tilde{d\boldsymbol{X}}')\sqrt{\boldsymbol{Y}}.$$

From the above equality, we obtain that

$$0 = \|\sqrt{\boldsymbol{X}}(d\boldsymbol{Y}' + \tilde{d\boldsymbol{Y}}')\sqrt{\boldsymbol{Y}}^{-1}\|_F^2 \ + \ \|\sqrt{\boldsymbol{X}}^{-1}(d\boldsymbol{X}' + \tilde{d\boldsymbol{X}}')\sqrt{\boldsymbol{Y}}\|_F^2$$

$$
\begin{aligned}
& + \sqrt{\boldsymbol{X}}(d\boldsymbol{Y}' + d\tilde{\boldsymbol{Y}}')\sqrt{\boldsymbol{Y}}^{-1} \bullet \sqrt{\boldsymbol{X}}^{-1}(d\boldsymbol{X}' + d\tilde{\boldsymbol{X}}')\sqrt{\boldsymbol{Y}} \\
& + \sqrt{\boldsymbol{X}}^{-1}(d\boldsymbol{X}' + d\tilde{\boldsymbol{X}}')\sqrt{\boldsymbol{Y}} \bullet \sqrt{\boldsymbol{X}}(d\boldsymbol{Y}' + d\tilde{\boldsymbol{Y}}')\sqrt{\boldsymbol{Y}}^{-1} \\
= {} & \|\sqrt{\boldsymbol{X}}(d\boldsymbol{Y}' + d\tilde{\boldsymbol{Y}}')\sqrt{\boldsymbol{Y}}^{-1}\|_F^2 \ + \ \|\sqrt{\boldsymbol{X}}^{-1}(d\boldsymbol{X}' + d\tilde{\boldsymbol{X}}')\sqrt{\boldsymbol{Y}}\|_F^2 \\
& + 2d\boldsymbol{Y}' \bullet d\boldsymbol{X}' + 2d\tilde{\boldsymbol{X}}' \bullet d\tilde{\boldsymbol{Y}}' \\
& (\text{since } d\boldsymbol{Y}' \bullet d\tilde{\boldsymbol{X}}' = d\tilde{\boldsymbol{Y}}' \bullet d\boldsymbol{X}' = 0 \text{ from (18)}) \\
\geq {} & \|\sqrt{\boldsymbol{X}}(d\boldsymbol{Y}' + d\tilde{\boldsymbol{Y}}')\sqrt{\boldsymbol{Y}}^{-1}\|_F^2 \ + \ \|\sqrt{\boldsymbol{X}}^{-1}(d\boldsymbol{X}' + d\tilde{\boldsymbol{X}}')\sqrt{\boldsymbol{Y}}\|_F^2 \\
& (\text{since } d\boldsymbol{Y}' \bullet d\boldsymbol{X}' \geq 0 \text{ and } d\tilde{\boldsymbol{X}}' \bullet d\tilde{\boldsymbol{Y}}' \geq 0).
\end{aligned}
$$

Hence we see that $\|\sqrt{\boldsymbol{X}}(d\boldsymbol{Y}' + d\tilde{\boldsymbol{Y}}')\sqrt{\boldsymbol{Y}}^{-1}\|_F^2 = 0$ and $\|\sqrt{\boldsymbol{X}}^{-1}(d\boldsymbol{X}' + d\tilde{\boldsymbol{X}}')\sqrt{\boldsymbol{Y}}\|_F^2 = 0$. This implies that $\sqrt{\boldsymbol{X}}(d\boldsymbol{Y}' + d\tilde{\boldsymbol{Y}}')\sqrt{\boldsymbol{Y}}^{-1} = \boldsymbol{O}$ and $\sqrt{\boldsymbol{X}}^{-1}(d\boldsymbol{X}' + d\tilde{\boldsymbol{X}}')\sqrt{\boldsymbol{Y}} = \boldsymbol{O}$. By the nonsingularity of $\sqrt{\boldsymbol{X}}$ and $\sqrt{\boldsymbol{Y}}$, we obtain that $d\boldsymbol{Y}' + d\tilde{\boldsymbol{Y}}' = \boldsymbol{O}$ and $d\boldsymbol{X}' + d\tilde{\boldsymbol{X}}' = \boldsymbol{O}$. We see from (18) that $d\boldsymbol{X}' \bullet d\tilde{\boldsymbol{X}}' = d\boldsymbol{Y}' \bullet d\tilde{\boldsymbol{Y}}' = 0$. Hence the equalities above imply that

$$
(\boldsymbol{O}, \boldsymbol{O}) = (d\boldsymbol{X}', d\boldsymbol{Y}') = \sum_{i=1}^{p} c_i'(\boldsymbol{M}^i, \boldsymbol{N}^i) \ \text{ and } (\boldsymbol{O}, \boldsymbol{O}) = (d\tilde{\boldsymbol{X}}', d\tilde{\boldsymbol{Y}}') = \sum_{j=1}^{\tilde{p}} \tilde{c}_j'(\tilde{\boldsymbol{M}}^j, \tilde{\boldsymbol{N}}^j).
$$

Recall that $\{(\boldsymbol{M}^i, \boldsymbol{N}^i) \in \mathcal{S}^2 \ (i = 1, 2, \ldots, p)\}$ and $\{(\tilde{\boldsymbol{M}}^j, \tilde{\boldsymbol{N}}^j) \in \tilde{\mathcal{S}}^2 \ (j = 1, 2, \ldots, \tilde{p})\}$ are bases of $\mathcal{F}^0$ and $\tilde{\mathcal{F}}^0$, respectively. Hence $c_i' = 0 \ (i = 1, 2, \ldots, p)$ and $\tilde{c}_j' = 0 \ (j = 1, 2, \ldots, \tilde{p})$. This means that the set of n^2 matrices given in (20) is linearly independent and forms a basis of the n^2-dimensional linear space $\hat{\mathcal{S}}$. This completes the proof of Theorem 4.2.

We can rewrite the Newton equation (19) as

$$
(\boldsymbol{X} + d\hat{\boldsymbol{X}}, \boldsymbol{Y} + d\hat{\boldsymbol{Y}}) \in \mathcal{F} + \tilde{\mathcal{F}}^0 \ \text{ and } \boldsymbol{X} d\hat{\boldsymbol{Y}} + d\hat{\boldsymbol{X}} \boldsymbol{Y} = \boldsymbol{Q}. \tag{23}
$$

In fact,

(i) if $(d\boldsymbol{X}, d\boldsymbol{Y}, d\tilde{\boldsymbol{X}}, d\tilde{\boldsymbol{Y}}) \in \mathcal{S}^2 \times \tilde{\mathcal{S}}^2$ is a solution of (19), then $(d\hat{\boldsymbol{X}}, d\hat{\boldsymbol{Y}}) = (d\boldsymbol{X} + d\tilde{\boldsymbol{X}}, d\boldsymbol{Y} + d\tilde{\boldsymbol{Y}})$ is a solution of (23).

(ii) if $(d\hat{\boldsymbol{X}}, d\hat{\boldsymbol{Y}}) \in \hat{\mathcal{S}}^2$ is a solution of (23), then

$$
((d\hat{\boldsymbol{X}} + d\hat{\boldsymbol{X}}^T)/2, (d\hat{\boldsymbol{Y}} + d\hat{\boldsymbol{Y}}^T)/2, (d\hat{\boldsymbol{X}} - d\hat{\boldsymbol{X}}^T)/2, (d\hat{\boldsymbol{Y}} - d\hat{\boldsymbol{Y}}^T)/2)
$$

is a solution of (19).

The proof of Theorem 4.2 and the argument above do not depend on the specific matrix $\boldsymbol{Q}$ of the right-hand side of the Newton equation (19) or (23). They remain valid for any $\boldsymbol{Q} \in \hat{\mathcal{S}}$. Thus we have the following corollary which will be utilized in our succeeding discussions.

COROLLARY 4.3. *Let $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}^2_{++}$ and $\boldsymbol{Q} \in \hat{\mathcal{S}}$.*

1. *The system of equations*

$$
(d\hat{\boldsymbol{X}}, d\hat{\boldsymbol{Y}}) \in \mathcal{F}^0 + \tilde{\mathcal{F}}^0 \ \text{ and } \boldsymbol{X} d\hat{\boldsymbol{Y}} + d\hat{\boldsymbol{X}} \boldsymbol{Y} = \boldsymbol{Q} \tag{24}
$$

has a unique solution $(d\hat{\boldsymbol{X}}, d\hat{\boldsymbol{Y}}) \in \hat{\mathcal{S}}^2$.

2. *The solution $(d\hat{\boldsymbol{X}}, d\hat{\boldsymbol{Y}}) \in \mathcal{F}^0 + \tilde{\mathcal{F}}^0$ satisfies*

$$
\sqrt{\boldsymbol{X}} d\hat{\boldsymbol{Y}} \sqrt{\boldsymbol{Y}}^{-1} + \sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}} = \sqrt{\boldsymbol{X}}^{-1} \boldsymbol{Q} \sqrt{\boldsymbol{Y}}^{-1}
$$

and

$$\sqrt{\boldsymbol{X}}^{-1} \hat{d\boldsymbol{X}} \sqrt{\boldsymbol{Y}} \bullet \sqrt{\boldsymbol{X}} \hat{d\boldsymbol{Y}} \sqrt{\boldsymbol{Y}}^{-1} \geq 0.$$

Proof. To prove assertion 1, let $(\boldsymbol{X}^0, \boldsymbol{Y}^0) \in \mathcal{F}$ and $\boldsymbol{Q}' = \boldsymbol{Q} - \boldsymbol{X}(\boldsymbol{Y} - \boldsymbol{Y}^0) - (\boldsymbol{X} - \boldsymbol{X}^0)\boldsymbol{Y}$. In view of the discussion above, there exists a unique solution $(\hat{d\boldsymbol{X}}', \hat{d\boldsymbol{Y}}') \in \hat{\mathcal{S}}^2$ of the system of equations

$$(\boldsymbol{X} + \hat{d\boldsymbol{X}}', \boldsymbol{Y} + \hat{d\boldsymbol{Y}}') \ \in \ \mathcal{F} + \tilde{\mathcal{F}}^0 \ \text{ and } \ \boldsymbol{X}\hat{d\boldsymbol{Y}}' + \hat{d\boldsymbol{X}}'\boldsymbol{Y} = \boldsymbol{Q}'.$$

We can rewrite this system of equations as

$$\begin{aligned}&(\boldsymbol{X} + \hat{d\boldsymbol{X}}' - \boldsymbol{X}^0, \boldsymbol{Y} + \hat{d\boldsymbol{Y}}' - \boldsymbol{Y}^0) \in \mathcal{F}^0 + \tilde{\mathcal{F}}^0 \text{ and}\\ &\boldsymbol{X}(\boldsymbol{Y} + \hat{d\boldsymbol{Y}}' - \boldsymbol{Y}^0) + (\boldsymbol{X} + \hat{d\boldsymbol{X}}' - \boldsymbol{X}^0)\boldsymbol{Y} = \boldsymbol{Q}.\end{aligned}$$

Letting $(\hat{d\boldsymbol{X}}, \hat{d\boldsymbol{Y}}) = (\boldsymbol{X} + \hat{d\boldsymbol{X}}' - \boldsymbol{X}^0, \boldsymbol{Y} + \hat{d\boldsymbol{Y}}' - \boldsymbol{Y}^0)$, we see that $(\hat{d\boldsymbol{X}}, \hat{d\boldsymbol{Y}})$ is a unique solution of (24). Multiplying both sides of the last equality in the system (24) of equations by $\sqrt{\boldsymbol{X}}^{-1}$ from the left and $\sqrt{\boldsymbol{Y}}^{-1}$ from the right, we have the first relation of assertion 2. The second relation of 2 follows from Conditions 1.2 and 4.1 and the relation (18). □

5. A generic interior-point method.

Generic IP method.

Step 0: Choose $(\boldsymbol{X}^0, \boldsymbol{Y}^0) \in \mathcal{S}^2_{++}$. Let $r = 0$.

Step 1: Let $(\boldsymbol{X}, \boldsymbol{Y}) = (\boldsymbol{X}^r, \boldsymbol{Y}^r)$ and $\mu = (\boldsymbol{X} \bullet \boldsymbol{Y})/n$.

Step 2: Choose *a direction parameter* $\beta \geq 0$.

Step 3: Compute a solution $(\hat{d\boldsymbol{X}}, \hat{d\boldsymbol{Y}}) \in \hat{\mathcal{S}}^2$ of the system (23) of equations with $\boldsymbol{Q} = \beta\mu\boldsymbol{I} - \boldsymbol{X}\boldsymbol{Y}$.

Step 4: Let $d\boldsymbol{X} = (\hat{d\boldsymbol{X}} + \hat{d\boldsymbol{X}}^T)/2$ and $d\boldsymbol{Y} = (\hat{d\boldsymbol{Y}} + \hat{d\boldsymbol{Y}}^T)/2$.

Step 5: Choose *a step size parameter* $\alpha \geq 0$ such that

$$(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}) = (\boldsymbol{X}, \boldsymbol{Y}) + \alpha(d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{S}^2_{++}. \tag{25}$$

Let $(\boldsymbol{X}^{r+1}, \boldsymbol{Y}^{r+1}) = (\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}})$.

Step 6: Replace r by $r + 1$ and go to Step 1.

The generic IP method involves two parameters: a search direction parameter $\beta \geq 0$ and a step size parameter $\alpha \geq 0$. If we choose an initial point $(\boldsymbol{X}^0, \boldsymbol{Y}^0) \in \mathcal{S}^2_{++}$, which can be infeasible, and if we specify $\beta \geq 0$ and $\alpha \geq 0$ satisfying (25) in each iteration, the method consistently generates a sequence $\{(\boldsymbol{X}^r, \boldsymbol{Y}^r)\}$ in $\mathcal{S}^2_{++}$. The lemma below is useful to determine a legitimate step size parameter α satisfying (25) and is closely related to the generalized eigenvalue problem of the matrix pencil (see, for example, [8]).

LEMMA 5.1. *Suppose that* $\boldsymbol{X} \in \mathcal{S}_{++}$, $d\boldsymbol{X} \in \mathcal{S}$, *and* $\alpha \geq 0$. *Let* ξ_{min} *be the minimum eigenvalue of the matrix* $\boldsymbol{X}^{-1}d\boldsymbol{X}$ *and let*

$$\bar{\alpha} = \sup\{\alpha : 1 + \alpha\xi_{min} \geq 0\} = \begin{cases} -1/\xi_{min} & \text{if } \xi_{min} < 0,\\ +\infty & \text{otherwise.}\end{cases}$$

Then $\boldsymbol{X} + \alpha d\boldsymbol{X} \in \mathcal{S}_{++}$ *if and only if* $\alpha < \bar{\alpha}$.

If we compute the minimum ζ_{min} of all the eigenvalues of $\boldsymbol{X}^{-1}d\boldsymbol{X}$ and $\boldsymbol{Y}^{-1}d\boldsymbol{Y}$ in Step 5 of the generic IP method, then

$$\alpha_{bd} = \begin{cases} -1/\zeta_{min} & \text{if } \zeta_{min} < 0, \\ +\infty & \text{otherwise} \end{cases}$$

gives the upper bound for a step size $\alpha \geq 0$ which satisfies (25).

The generic IP method is analogous to many infeasible interior-point methods ([17, 18, 23, 26, 48], etc.) developed for the monotone LCP (8). Specifically the generic IP method shares with them the features that we can start from an infeasible initial point and that we utilize a Newton direction for approximating a point on the central trajectory. A difference lies in Step 4 of the generic IP method where we symmetrize the Newton direction $(\hat{d\boldsymbol{X}}, \hat{d\boldsymbol{Y}})$ computed at Step 3 to create a symmetric search direction $(d\boldsymbol{X}, d\boldsymbol{Y})$. This ensures that each iterate $(\boldsymbol{X}^r, \boldsymbol{Y}^r)$ runs in the set $\mathcal{S}^2_{++}$ of symmetric positive definite matrices whenever we take an initial point $(\boldsymbol{X}^0, \boldsymbol{Y}^0)$ in the set $\mathcal{S}^2_{++}$. The main reason why we use the symmetrized direction is that handling symmetric matrices is much easier than handling nonsymmetric matrices theoretically and practically. In particular,

(a) The logarithmic barrier function $\phi(\mu, \cdot)$ with a fixed $\mu > 0$ is strictly convex on $\mathcal{S}^2_{++}$ (see Lemma 3.2) but not convex on $\hat{\mathcal{S}}^2_{++}$, where $\hat{\mathcal{S}}_{++} = \{\boldsymbol{X} \in \hat{\mathcal{S}} : \boldsymbol{X} \succ \boldsymbol{O}\}$. In fact, if $(\boldsymbol{X}, \hat{\boldsymbol{Y}}) \in \mathcal{S}_{++} \times \hat{\mathcal{S}}_{++}$ but $\hat{\boldsymbol{Y}} \notin \mathcal{S}_{++}$ then

$$\det \frac{\hat{\boldsymbol{Y}} + \hat{\boldsymbol{Y}}^T}{2} < \det \hat{\boldsymbol{Y}} = \det \hat{\boldsymbol{Y}}^T \text{ (see [38]);}$$

hence

$$\phi\left(\mu, \boldsymbol{X}, \frac{\hat{\boldsymbol{Y}} + \hat{\boldsymbol{Y}}^T}{2}\right) > \frac{1}{2}\phi(\mu, \boldsymbol{X}, \hat{\boldsymbol{Y}}) + \frac{1}{2}\phi(\mu, \boldsymbol{X}, \hat{\boldsymbol{Y}}^T).$$

(b) If we confine the sequence $\{(\boldsymbol{X}^r, \boldsymbol{Y}^r)\}$ within $\mathcal{F}_{++} \subset \hat{\mathcal{S}}^2$, the sequence is at least bounded and any accumulation point is a solution of the SDLCP (1) in symmetric matrices as observed in Theorem 3.1.

As special cases of the generic IP method, we present a central trajectory following method in section 8.1 and a potential-reduction method in section 8.2. Both methods may be classified into *feasible interior-point methods*; they start from a feasible interior point $(\boldsymbol{X}^0, \boldsymbol{Y}^0) \in \mathcal{F}_{++}$ and generate a sequence $\{(\boldsymbol{X}^r, \boldsymbol{Y}^r)\}$ in the interior $\mathcal{F}_{++}$ of the feasible region such that $\boldsymbol{X}^r \bullet \boldsymbol{Y}^r \to 0$ as r tends to ∞ . It follows from the monotonicity that

$$\boldsymbol{Y}^0 \bullet \boldsymbol{X}^r + \boldsymbol{X}^0 \bullet \boldsymbol{Y}^r \leq \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0 + \boldsymbol{X}^r \bullet \boldsymbol{Y}^r \text{ for every } r = 0, 1, \dots .$$

Since $\boldsymbol{X}^r \bullet \boldsymbol{Y}^r \to 0$ as r tends to ∞ , the right-hand side is bounded by a positive number, say ω. This implies that the sequence $\{(\boldsymbol{X}^r, \boldsymbol{Y}^r)\}$ lies in a closed and bounded set

$$\{(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}_+ : \boldsymbol{Y}^0 \bullet \boldsymbol{X} + \boldsymbol{X}^0 \bullet \boldsymbol{Y} \leq \omega\}.$$

See Lemma 1.1 for the boundedness of the set. Therefore the sequence $\{(\boldsymbol{X}^r, \boldsymbol{Y}^r)\}$ has at least one accumulation point and every accumulation point is a solution of the SDLCP (1) in symmetric matrices.

When $(\boldsymbol{X},\boldsymbol{Y}) = (\boldsymbol{X}^r,\boldsymbol{Y}^r) \in \mathcal{S}^2_{++}$ is in the interior $\mathcal{F}_{++}$ of the feasible region, the Newton equation (23) turns out to be

$$(d\hat{\boldsymbol{X}}, d\hat{\boldsymbol{Y}}) \in \mathcal{F}^0 + \tilde{\mathcal{F}}^0 \ \text{ and } \boldsymbol{X}d\hat{\boldsymbol{Y}} + d\hat{\boldsymbol{X}}\boldsymbol{Y} = \boldsymbol{Q}, \tag{26}$$

and the search direction $(d\boldsymbol{X}, d\boldsymbol{Y}) = ((d\hat{\boldsymbol{X}} + d\hat{\boldsymbol{X}}^T)/2, (d\hat{\boldsymbol{Y}} + d\hat{\boldsymbol{Y}}^T)/2)$ computed at Step 4 lies in $\mathcal{F}^0$. In this case, (26) coincides with (24). Hence the solution of (26) satisfies item 2 of Corollary 4.3.

6. Some properties of the solution set. It is well known (see, for example, [6]) that if the feasible region of the monotone LCP (8) in the Euclidean space is nonempty then

(i) the solution set of the LCP (8) is a nonempty convex set,

(ii) there exist subsets I, J of the index set $\{1, 2, \ldots, n\}$ such that

$$\begin{array}{l} I \cup J = \{1,2,\ldots,n\}, \\ \left.\begin{array}{rcl} x_i & = & 0 \ (i \in I) \\ y_i & = & 0 \ (i \in J) \end{array}\right\} \text{for every solution } (\boldsymbol{x},\boldsymbol{y}) \text{ of the LCP (8).} \end{array}$$

We can prove similar results on the monotone SDLCP (1) in symmetric matrices under a slightly stronger assumption that the interior $\mathcal{F}_{++}$ of the feasible region is nonempty.

THEOREM 6.1. *Suppose that the interior $\mathcal{F}_{++}$ of the feasible region of the SDLCP* (1) *in symmetric matrices is nonempty.*

1. *The solution set $\mathcal{F}^*$ of the monotone SDLCP* (1) *is a nonempty convex set.*
2. *There exist subsets I, J of the index set $\{1, 2, \ldots, n\}$ and an orthogonal matrix $\boldsymbol{P}$ such that*

$$\begin{array}{l} I \cup J = \ \{1,2,\ldots,n\}, \\ \left.\begin{array}{rcl} \left[\boldsymbol{P}^T\boldsymbol{X}\boldsymbol{P}\right]_{ij} & = & 0 \ (i \in I \ \text{ or } j \in I) \\ \left[\boldsymbol{P}^T\boldsymbol{Y}\boldsymbol{P}\right]_{ij} & = & 0 \ (i \in J \ \text{ or } j \in J) \end{array}\right\} \text{for every } (\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{F}^*. \end{array}$$

Proof. 1. The nonemptiness of the solution set $\mathcal{F}^*$ follows from Theorem 3.1. Suppose that $(\boldsymbol{X}^1,\boldsymbol{Y}^1),\ (\boldsymbol{X}^2,\boldsymbol{Y}^2) \in \mathcal{F}^*$. By Condition 1.2,

$$0 \geq -(\boldsymbol{X}^2 - \boldsymbol{X}^1) \bullet (\boldsymbol{Y}^2 - \boldsymbol{Y}^1) = \boldsymbol{X}^2 \bullet \boldsymbol{Y}^1 + \boldsymbol{X}^1 \bullet \boldsymbol{Y}^2.$$

Since all matrices $\boldsymbol{X}^1$, $\boldsymbol{X}^2$, $\boldsymbol{Y}^1$, and $\boldsymbol{Y}^2$ are symmetric and positive semidefinite, the inequality above implies that $\boldsymbol{X}^1 \bullet \boldsymbol{Y}^2 = 0$ and $\boldsymbol{X}^2 \bullet \boldsymbol{Y}^1 = 0$. See Lemma 1.1. Hence, for every $\lambda \in [0, 1]$,

$$\begin{array}{l} \lambda(\boldsymbol{X}^1,\boldsymbol{Y}^1) + (1-\lambda)(\boldsymbol{X}^2,\boldsymbol{Y}^2) \in \mathcal{F}_+, \\ \big(\lambda\boldsymbol{X}^1 + (1-\lambda)\boldsymbol{X}^2\big) \bullet \big(\lambda\boldsymbol{Y}^1 + (1-\lambda)\boldsymbol{Y}^2\big) = 0. \end{array}$$

Thus we have shown that the solution set $\mathcal{F}^*$ is convex.

2. Let r be the dimension of the affine subspace spanned by the solution set $\mathcal{F}^*$. Then there exist $r+1$ pairs of matrices $(\boldsymbol{X}^0,\boldsymbol{Y}^0), (\boldsymbol{X}^1,\boldsymbol{Y}^1), \ldots, (\boldsymbol{X}^r,\boldsymbol{Y}^r) \in \mathcal{F}^*$ such that any $(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{F}^*$ can be represented as $(\boldsymbol{X},\boldsymbol{Y}) = \sum_{j=0}^r \alpha_j(\boldsymbol{X}^j,\boldsymbol{Y}^j)$, $\sum_{j=0}^r \alpha_j =$

1 for some $\alpha_0, \alpha_1, \dots, \alpha_r$. Define

$$(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}) = \frac{1}{r+1} \sum_{j=0}^{r} (\boldsymbol{X}^j, \boldsymbol{Y}^j),$$
$$K = \{\boldsymbol{p} \in R^n : \boldsymbol{p}^T \bar{\boldsymbol{X}} \boldsymbol{p} = 0\},$$
$$L = \{\boldsymbol{p} \in R^n : \boldsymbol{p}^T \bar{\boldsymbol{Y}} \boldsymbol{p} = 0\}.$$

(K coincides with the subspace spanned by all the eigenvectors of $\bar{\boldsymbol{X}}$ associated with its zero eigenvalue, and L coincides with the subspace spanned by all the eigenvectors of $\bar{\boldsymbol{Y}}$ associated with its zero eigenvalue.) Then for every $\boldsymbol{p} \in K$,

$$0 = \boldsymbol{p}^T \bar{\boldsymbol{X}} \boldsymbol{p} = \frac{1}{r+1} \sum_{j=0}^{r} \boldsymbol{p}^T \boldsymbol{X}^j \boldsymbol{p}.$$

Since each $\boldsymbol{X}^j$ is positive semidefinite, we have $\boldsymbol{X}^j \boldsymbol{p} = \boldsymbol{0}$ $(j = 0, 1, \dots, r)$. Hence

$$\boldsymbol{X}\boldsymbol{p} = \boldsymbol{0} \ \text{ for every } (\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}^* \ \text{ and every } \boldsymbol{p} \in K. \tag{27}$$

Similarly we have that

$$\boldsymbol{Y}\boldsymbol{p} = \boldsymbol{0} \ \text{ for every } (\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}^* \ \text{ and every } \boldsymbol{p} \in L. \tag{28}$$

Let $\xi_1, \xi_2, \dots, \xi_n$ denote the eigenvalues of $\bar{\boldsymbol{X}}$ and $\bar{\boldsymbol{p}}^1, \bar{\boldsymbol{p}}^2, \dots, \bar{\boldsymbol{p}}^n$ the eigenvectors corresponding to them. We may assume that the eigenvectors form a normalized orthogonal basis of R^n. Define $I = \{j : \xi_j = 0\}$ and $J = \{j : \xi_j > 0\}$. Then $\{\bar{\boldsymbol{p}}^j : j \in I\}$ forms a basis of the subspace K. It follows from (27) that

$$\boldsymbol{X}\bar{\boldsymbol{p}}^j = \boldsymbol{0} \ \text{ for every } (\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}^* \ \text{ and every } j \in I. \tag{29}$$

On the other hand, $\bar{\boldsymbol{X}}\bar{\boldsymbol{Y}} = \boldsymbol{O}$ because $(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}) \in \mathcal{F}^*$. Hence

$$0 = (\bar{\boldsymbol{p}}^j)^T \bar{\boldsymbol{X}} \bar{\boldsymbol{Y}} \bar{\boldsymbol{p}}^j = \xi_j (\bar{\boldsymbol{p}}^j)^T \bar{\boldsymbol{Y}} \bar{\boldsymbol{p}}^j \ \text{ for every } j \in J,$$

which together with $\xi_j > 0$ $(j \in J)$ implies that $(\bar{\boldsymbol{p}}^j)^T \bar{\boldsymbol{Y}} \bar{\boldsymbol{p}}^j = 0$ for every $j \in J$, or equivalently $\bar{\boldsymbol{p}}^j \in L$ for every $j \in J$. By (28), we then see that

$$\boldsymbol{Y}\bar{\boldsymbol{p}}^j = \boldsymbol{0} \ \text{ for every } (\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}^* \ \text{ and every } j \in J. \tag{30}$$

Letting $\boldsymbol{P} = (\bar{\boldsymbol{p}}^1, \bar{\boldsymbol{p}}^2, \dots, \bar{\boldsymbol{p}}^n)$, we obtain the desired result from the definition of I, J, (29), and (30). □

Remark 6.2.

(a) In item 1 of Theorem 6.1, the condition $\mathcal{F}_{++} \neq \emptyset$ cannot be weakened. It is well known that the nonemptiness of the feasible region gives a necessary and sufficient condition for the existence of a solution of the monotone LCP (8) in the Euclidean space (see, for example, [6]). In contrast with that case, however, the weaker condition $\mathcal{F}_{+} \neq \emptyset$ does not imply the solvability of the SDLCP(1). This is due to the fact that the cone of positive semidefinite matrices is not polyhedral. (See Gowda and Seidman [10].) There is a similar gap in the case of the monotone nonlinear complementarity problem [29].

(b) Result 2 of Theorem 6.1 is closely related to the complementary slackness theorem (Corollary 2.11 of [2]): it says that all solutions $(\boldsymbol{X}, \boldsymbol{Y})$s of the SDP(2) share a system of eigenvectors and their eigenvalues are complementary in the sense of LCP.

7. Basic lemmas. In this section we prepare basic lemmas which play important roles in what follows.

LEMMA 7.1. *Suppose that* $(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{S}^2_{++}$. *Let* λ_{min} *and* λ_{max} *denote the minimum and the maximum eigenvalues of* $\boldsymbol{XY}$, *respectively.*

1. *Let* $(d\hat{\boldsymbol{X}}, d\hat{\boldsymbol{Y}}) \in \hat{\mathcal{S}}^2$. *Then*

$$\|\sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{X}}^{-1}\|_F \leq \frac{\|\sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}}\|_F}{\sqrt{\lambda_{min}}}, \tag{31}$$

$$\|\sqrt{\boldsymbol{Y}}^{-1} d\hat{\boldsymbol{Y}} \sqrt{\boldsymbol{Y}}^{-1}\|_F \leq \frac{\|\sqrt{\boldsymbol{X}} d\hat{\boldsymbol{Y}} \sqrt{\boldsymbol{Y}}^{-1}\|_F}{\sqrt{\lambda_{min}}}, \tag{32}$$

$$\|\sqrt{\boldsymbol{Y}} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}}\|_F \leq \sqrt{\lambda_{max}} \|\sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}}\|_F, \tag{33}$$

$$\|\sqrt{\boldsymbol{X}} d\hat{\boldsymbol{Y}} \sqrt{\boldsymbol{X}}\|_F \leq \sqrt{\lambda_{max}} \|\sqrt{\boldsymbol{X}} d\hat{\boldsymbol{Y}} \sqrt{\boldsymbol{Y}}^{-1}\|_F. \tag{34}$$

2. *Let* $(d\hat{\boldsymbol{X}}, d\hat{\boldsymbol{Y}}) \in \hat{\mathcal{S}}^2$, $d\boldsymbol{X} = (d\hat{\boldsymbol{X}} + d\hat{\boldsymbol{X}}^T)/2$, $d\boldsymbol{Y} = (d\hat{\boldsymbol{Y}} + d\hat{\boldsymbol{Y}}^T)/2$, $\xi_1, \xi_2, \ldots, \xi_n$ *be the eigenvalues of* $\boldsymbol{X}^{-1} d\boldsymbol{X}$, *and* $\eta_1, \eta_2, \ldots, \eta_n$ *be the eigenvalues of* $\boldsymbol{Y}^{-1} d\boldsymbol{Y}$. *Then*

$$\sum_{j=1}^n \xi_j^2 \leq \|\sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{X}}^{-1}\|_F^2 \leq \frac{\|\sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}}\|_F^2}{\lambda_{min}}, \tag{35}$$

$$\sum_{j=1}^n \eta_j^2 \leq \|\sqrt{\boldsymbol{Y}}^{-1} d\hat{\boldsymbol{Y}} \sqrt{\boldsymbol{Y}}^{-1}\|_F^2 \leq \frac{\|\sqrt{\boldsymbol{X}} d\hat{\boldsymbol{Y}} \sqrt{\boldsymbol{Y}}^{-1}\|_F^2}{\lambda_{min}}. \tag{36}$$

3. (*Extension of the inequalities given in Lemma* 4.20 *of* [19]). *Let* $\boldsymbol{Q} \in \hat{\mathcal{S}}$ *and let* $(d\hat{\boldsymbol{X}}, d\hat{\boldsymbol{Y}})$ *be a solution of the system* (24) *of equations. Then*

$$\begin{aligned} &\|\sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}}\|_F^2 + \|\sqrt{\boldsymbol{X}} d\hat{\boldsymbol{Y}} \sqrt{\boldsymbol{Y}}^{-1}\|_F^2 \\ &\quad = \|\sqrt{\boldsymbol{X}}^{-1} \boldsymbol{Q} \sqrt{\boldsymbol{Y}}^{-1}\|_F^2 - 2 d\hat{\boldsymbol{X}} \bullet d\hat{\boldsymbol{Y}}, \end{aligned} \tag{37}$$

$$0 \leq d\hat{\boldsymbol{X}} \bullet d\hat{\boldsymbol{Y}} \leq \frac{\|\sqrt{\boldsymbol{X}}^{-1} \boldsymbol{Q} \sqrt{\boldsymbol{Y}}^{-1}\|_F^2}{4}. \tag{38}$$

Proof. 1. In general, the inequalities $\nu_{min}(\boldsymbol{A})\|\boldsymbol{B}\|_F^2 \leq \mathrm{Tr}\ \boldsymbol{B}^T\boldsymbol{A}\boldsymbol{B} \leq \nu_{max}(\boldsymbol{A})\|\boldsymbol{B}\|_F^2$ hold for every $\boldsymbol{A} \in \mathcal{S}_{++}$ and every $\boldsymbol{B} \in \hat{\mathcal{S}}$, where $\nu_{min}(\boldsymbol{A})$ and $\nu_{max}(\boldsymbol{A})$ denote the minimum and the maximum eigenvalues of $\boldsymbol{A}$, respectively. On the other hand, λ_{min} and $1/\lambda_{max}$ are the minimum eigenvalues of the matrix $\sqrt{\boldsymbol{X}}\boldsymbol{Y}\sqrt{\boldsymbol{X}}$ and its inverse $(\sqrt{\boldsymbol{X}}\boldsymbol{Y}\sqrt{\boldsymbol{X}})^{-1}$, respectively. Hence

$$\begin{aligned} \|\sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}}\|_F^2 &= \mathrm{Tr}\ (\sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{X}}^{-1})^T \sqrt{\boldsymbol{X}}\boldsymbol{Y}\sqrt{\boldsymbol{X}} (\sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{X}}^{-1}) \\ &\geq \lambda_{min} \|\sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{X}}^{-1}\|_F^2, \\ \|\sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}}\|_F^2 &= \mathrm{Tr}\ (\sqrt{\boldsymbol{Y}} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}})^T (\sqrt{\boldsymbol{Y}}\boldsymbol{X}\sqrt{\boldsymbol{Y}})^{-1} (\sqrt{\boldsymbol{Y}} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}}) \\ &\geq \frac{1}{\lambda_{max}} \|\sqrt{\boldsymbol{Y}} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}}\|_F^2. \end{aligned}$$

Thus we have shown (31) and (33). The proof of (32) and (34) is quite similar.

2. We only show that the inequality (35) holds. The inequality (36) can be proven similarly. Since the symmetric matrix $\sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X} \sqrt{\boldsymbol{X}}^{-1}$ has the same eigenvalues

$\xi_1, \xi_2, \ldots, \xi_n$ as the matrix $\boldsymbol{X}^{-1} d\boldsymbol{X}$, we have that

$$\begin{aligned}
\sum_{j=1}^{n} \xi_j^2 &= \|\sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X} \sqrt{\boldsymbol{X}}^{-1}\|_F^2 \\
&= \left\| \sqrt{\boldsymbol{X}}^{-1} \left(\frac{d\hat{\boldsymbol{X}} + d\hat{\boldsymbol{X}}^T}{2} \right) \sqrt{\boldsymbol{X}}^{-1} \right\|_F^2 \quad (\text{since } d\boldsymbol{X} = (d\hat{\boldsymbol{X}} + d\hat{\boldsymbol{X}}^T)/2) \\
&\leq \left(\frac{\|\sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{X}}^{-1}\|_F}{2} + \frac{\|\sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{X}}^{-1}\|_F}{2} \right)^2 \\
&= \|\sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{X}}^{-1}\|_F^2 \\
&\leq \frac{\|\sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}}\|_F^2}{\lambda_{min}}.
\end{aligned}$$

Here the last inequality follows from (31).

3. By Corollary 4.3, we have $\sqrt{\boldsymbol{X}} d\hat{\boldsymbol{Y}} \sqrt{\boldsymbol{Y}}^{-1} + \sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}} = \sqrt{\boldsymbol{X}}^{-1} \boldsymbol{Q} \sqrt{\boldsymbol{Y}}^{-1}$. Hence

$$\begin{aligned}
&\|\sqrt{\boldsymbol{X}}^{-1} \boldsymbol{Q} \sqrt{\boldsymbol{Y}}^{-1}\|_F^2 \\
&= (\sqrt{\boldsymbol{X}} d\hat{\boldsymbol{Y}} \sqrt{\boldsymbol{Y}}^{-1} + \sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}}) \bullet (\sqrt{\boldsymbol{X}} d\hat{\boldsymbol{Y}} \sqrt{\boldsymbol{Y}}^{-1} + \sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}}) \\
&= \|\sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}}\|_F^2 + \|\sqrt{\boldsymbol{X}} d\hat{\boldsymbol{Y}} \sqrt{\boldsymbol{Y}}^{-1}\|_F^2 + 2 d\hat{\boldsymbol{X}} \bullet d\hat{\boldsymbol{Y}}.
\end{aligned}$$

Thus we have shown (37). Since the linear subspaces $\mathcal{F}^0$ and $\tilde{\mathcal{F}}^0$ of $\hat{\mathcal{S}}^2$ are orthogonal to each other (see (18)), the first inequality of (38) follows directly from Conditions 1.2 and 4.1. We also see that

$$\begin{aligned}
&d\hat{\boldsymbol{X}} \bullet d\hat{\boldsymbol{Y}} \\
&= \frac{1}{4} \left\{ \|\sqrt{\boldsymbol{X}} d\hat{\boldsymbol{Y}} \sqrt{\boldsymbol{Y}}^{-1} + \sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}}\|_F^2 - \|\sqrt{\boldsymbol{X}} d\hat{\boldsymbol{Y}} \sqrt{\boldsymbol{Y}}^{-1} - \sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}}\|_F^2 \right\} \\
&\leq \frac{1}{4} \left\{ \|\sqrt{\boldsymbol{X}} d\hat{\boldsymbol{Y}} \sqrt{\boldsymbol{Y}}^{-1} + \sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}}\|_F^2 \right\} \\
&= \frac{\|\sqrt{\boldsymbol{X}}^{-1} \boldsymbol{Q} \sqrt{\boldsymbol{Y}}^{-1}\|_F^2}{4}.
\end{aligned}$$

Thus we have shown (38). □

Lemma 7.2. *(Extension of Lemma 4.16 of [19]). Let $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}_{++}^2$, $\mu = \boldsymbol{X} \bullet \boldsymbol{Y}/n$, $0 \leq \beta \leq 1$ and let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of the matrix $\boldsymbol{X}\boldsymbol{Y}$. Define the $n \times n$ matrix $\boldsymbol{H}(\beta)$ as $\boldsymbol{H}(\beta) = \beta\mu \sqrt{\boldsymbol{X}}^{-1} \sqrt{\boldsymbol{Y}}^{-1} - \sqrt{\boldsymbol{X}}\sqrt{\boldsymbol{Y}}$.*

1. *Define $\boldsymbol{\Lambda} =$ diag $(\lambda_1, \lambda_2, \ldots, \lambda_n)$ to be the $n \times n$ diagonal matrix with the coordinates $\lambda_1, \lambda_2, \ldots, \lambda_n$. Then*

$$\|\boldsymbol{H}(\beta)\|_F^2 = \|\beta\mu\sqrt{\boldsymbol{\Lambda}}^{-1} - \sqrt{\boldsymbol{\Lambda}}\|_F^2. \tag{39}$$

2. *Assume that $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{N}(\gamma)$ for some $\gamma \in (0, \sqrt{n}]$. Here*

$$\mathcal{N}(\gamma) = \left\{ (\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}_{++} : \begin{array}{l} \left(\displaystyle\sum_{j=1}^{n} (\lambda_j - \mu)^2 \right)^{1/2} \leq \gamma\mu, \quad \textit{where } \mu = \dfrac{\boldsymbol{X} \bullet \boldsymbol{Y}}{n}, \\ \textit{and } \lambda_1, \ldots, \lambda_n \ \textit{ denote the eigenvalues of } \boldsymbol{X}\boldsymbol{Y} \end{array} \right\}$$

$$= \left\{ (\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}_{++} : \|\sqrt{\boldsymbol{X}}\boldsymbol{Y}\sqrt{\boldsymbol{X}} - \mu \boldsymbol{I}\|_F \leq \gamma\mu, \quad \textit{where } \mu = \frac{\boldsymbol{X} \bullet \boldsymbol{Y}}{n} \right\}.$$

Then

$$(1-\gamma)\mu \leq \lambda_{min} \leq \lambda_j \leq \lambda_{max} \leq (1+\gamma)\mu \quad \textit{for every } j = 1, 2, \ldots, n, \tag{40}$$

$$\|\boldsymbol{H}(\beta)\|_F \leq \min\left\{ \frac{((1-\beta)\sqrt{n}+\gamma)\mu}{\sqrt{\lambda_{min}}}, \ \frac{\sqrt{2n}\mu}{\sqrt{\lambda_{min}}} \right\}.$$

Proof. 1. By the definition, we see that

$$\begin{aligned} \|\boldsymbol{H}(\beta)\|_F^2 &= \|\beta\mu\sqrt{\boldsymbol{X}}^{-1}\sqrt{\boldsymbol{Y}}^{-1} - \sqrt{\boldsymbol{X}}\sqrt{\boldsymbol{Y}}\|_F^2 \\ &= \sum_{j=1}^{n} \left(\frac{\beta\mu}{\sqrt{\lambda_j}} - \sqrt{\lambda_j} \right)^2 \\ &= \|\beta\mu\sqrt{\boldsymbol{\Lambda}}^{-1} - \sqrt{\boldsymbol{\Lambda}}\|_F^2. \end{aligned}$$

Thus we have shown the equality (39).

2. We see from $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{N}(\gamma)$ that

$$\left(\sum_{j=1}^{n} (\lambda_j - \mu)^2 \right)^{1/2} \leq \gamma\mu, \tag{41}$$

which implies (40). Hence

$$\begin{aligned} \|\boldsymbol{H}(\beta)\|_F &= \|\beta\mu\sqrt{\boldsymbol{\Lambda}}^{-1} - \sqrt{\boldsymbol{\Lambda}}\|_F \ \text{(by (39))} \\ &\leq (1-\beta)\mu\|\sqrt{\boldsymbol{\Lambda}}^{-1}\|_F + \|\mu\sqrt{\boldsymbol{\Lambda}}^{-1} - \sqrt{\boldsymbol{\Lambda}}\|_F \\ &\leq (1-\beta)\mu \left(\sum_{j=1}^{n} \frac{1}{\lambda_{min}} \right)^{1/2} + \left(\sum_{j=1}^{n} \left(\frac{\mu - \lambda_j}{\sqrt{\lambda_{min}}} \right)^2 \right)^{1/2} \quad \text{(by (40))} \\ &\leq (1-\beta)\mu \cdot \frac{\sqrt{n}}{\sqrt{\lambda_{min}}} + \frac{\gamma\mu}{\sqrt{\lambda_{min}}} \quad \text{(by (41))} \\ &= \frac{((1-\beta)\sqrt{n}+\gamma)\,\mu}{\sqrt{\lambda_{min}}}. \end{aligned}$$

We also see that

$$\begin{aligned} \|\boldsymbol{H}(\beta)\|_F^2 &= \|\beta\mu\sqrt{\boldsymbol{\Lambda}}^{-1} - \sqrt{\boldsymbol{\Lambda}}\|_F^2 \ \text{(by (39))} \\ &\leq \sum_{j=1}^{n} \left(\left(\frac{\beta\mu}{\sqrt{\lambda_j}} \right)^2 + \left(\sqrt{\lambda_j} \right)^2 \right) \\ &\leq \frac{n\mu^2}{\lambda_{min}} + n\mu \left(\text{since } 0 \leq \beta \leq 1 \text{ and } \sum_{j=1}^{n} \lambda_j = n\mu \right) \\ &\leq \frac{2n\mu^2}{\lambda_{min}} \ (\text{since } 1 \leq \mu/\lambda_{min}) \\ &\leq \frac{2n\mu^2}{\lambda_{min}}. \end{aligned}$$

Thus we have shown assertion 2. □

Lemma 7.3. *(See* [16, 42]*, etc.)*

1. *If* $1+\xi>0$ *then* $\log(1+\xi)\le\xi$.

2. *If* $\boldsymbol{\xi}\in R^n$ *satisfies* $\|\boldsymbol{\xi}\|_\infty\le\tau<1$ *then* $\sum_{j=1}^n\log(1+\xi_j)\ge\sum_{j=1}^n\xi_j-\dfrac{\|\boldsymbol{\xi}\|^2}{2(1-\tau)}$.

For every $(\boldsymbol{X},\boldsymbol{Y})\in\mathcal{F}_{++}$, define the potential function

$$f(\boldsymbol{X},\boldsymbol{Y})=(n+\nu)\log\boldsymbol{X}\bullet\boldsymbol{Y}-\log\det\boldsymbol{X}\boldsymbol{Y}-n\log n.$$

Here $\nu\ge 0$ is a parameter. This potential function is the same as the one used in the paper [44] where Vandenberghe and Boyd proposed a potential-reduction method for the primal-dual pair (2) of SDPs. (See also [2, 36].)

Lemma 7.4. *Let* $(\boldsymbol{X},\boldsymbol{Y})\in\mathcal{F}_{++}$, $(d\boldsymbol{X},d\boldsymbol{Y})\in\mathcal{S}^2$, $\mu=\boldsymbol{X}\bullet\boldsymbol{Y}/n$, $0<\tau<1$, *and* $\nu\ge0$. *Let* $\xi_1,\xi_2,\ldots,\xi_n$ *be the eigenvalues of the matrix* $\boldsymbol{X}^{-1}d\boldsymbol{X}$ *and* $\eta_1,\eta_2,\ldots,\eta_n$ *be the eigenvalues of the matrix* $d\boldsymbol{Y}\boldsymbol{Y}^{-1}$. *Let* α *be a positive number such that* $|\alpha\xi_j|\le\tau$ *and* $|\alpha\eta_j|\le\tau$ *for every* $j=1,2,\ldots,n$.

1. $f(\boldsymbol{X}+\alpha d\boldsymbol{X},\boldsymbol{Y}+\alpha d\boldsymbol{Y})-f(\boldsymbol{X},\boldsymbol{Y})\le\alpha G_1(d\boldsymbol{X},d\boldsymbol{Y})+\alpha^2G_2(d\boldsymbol{X},d\boldsymbol{Y})$, *where*

$$G_1(d\boldsymbol{X},d\boldsymbol{Y})=Tr\left(\frac{n+\nu}{n\mu}\boldsymbol{I}-\boldsymbol{Y}^{-1}\boldsymbol{X}^{-1}\right)(d\boldsymbol{X}\boldsymbol{Y}+\boldsymbol{X}d\boldsymbol{Y}),$$
$$G_2(d\boldsymbol{X},d\boldsymbol{Y})=\frac{(n+\nu)d\boldsymbol{X}\bullet d\boldsymbol{Y}}{n\mu}+\frac{\sum_{j=1}^n(\xi_j^2+\eta_j^2)}{2(1-\tau)}.$$

2. *(Extension of Lemma* 2.5 *of* [22]*). Let* $\lambda_1,\lambda_2,\ldots,\lambda_n$ *be the eigenvalues of the matrix* $\boldsymbol{X}\boldsymbol{Y}$ *and* $\lambda_{min}=\min\{\lambda_1,\lambda_2,\ldots,\lambda_n\}$. *Assume that* $\beta=n/(n+\nu)$ *and that* $(d\hat{\boldsymbol{X}},d\hat{\boldsymbol{Y}})$ *is a solution of the Newton equation* (23) *with* $\boldsymbol{Q}=\beta\mu\boldsymbol{I}-\boldsymbol{X}\boldsymbol{Y}$. *Let* $d\boldsymbol{X}=(d\hat{\boldsymbol{X}}+d\hat{\boldsymbol{X}}^T)/2$ *and* $d\boldsymbol{Y}=(d\hat{\boldsymbol{Y}}+d\hat{\boldsymbol{Y}}^T)/2$. *Then*

$$G_1(d\boldsymbol{X},d\boldsymbol{Y})=-\frac{1}{\beta\mu}\|\boldsymbol{H}(\beta)\|_F^2,\tag{42}$$
$$G_2(d\boldsymbol{X},d\boldsymbol{Y})\le\frac{\|\boldsymbol{H}(\beta)\|_F^2}{2(1-\tau)\lambda_{min}}+\frac{1}{\lambda_{min}}\left(\frac{n+\nu}{n}-\frac{1}{1-\tau}\right)d\boldsymbol{X}\bullet d\boldsymbol{Y}.\tag{43}$$

If in addition, $\nu\ge\sqrt{n}$ *then*

$$\frac{\sqrt{\lambda_{min}}}{\beta\mu}\|\boldsymbol{H}(\beta)\|_F\ge\frac{\sqrt{3}}{2}.\tag{44}$$

Proof. 1. The desired inequality follows from the calculation below.

$$\begin{aligned}
&f(\boldsymbol{X}+\alpha d\boldsymbol{X},\boldsymbol{Y}+\alpha d\boldsymbol{Y})-f(\boldsymbol{X},\boldsymbol{Y})\\
&=(n+\nu)\log\left(1+\frac{\alpha\mathrm{Tr}\ (d\boldsymbol{X}\boldsymbol{Y}+\boldsymbol{X}d\boldsymbol{Y})}{n\mu}+\frac{\alpha^2d\boldsymbol{X}\bullet d\boldsymbol{Y}}{n\mu}\right)\\
&\quad-\sum_{j=1}^n(\log(1+\alpha\xi_j)+\log(1+\alpha\eta_j))\\
&\le(n+\nu)\left(\frac{\alpha\mathrm{Tr}\ (d\boldsymbol{X}\boldsymbol{Y}+\boldsymbol{X}d\boldsymbol{Y})}{n\mu}+\frac{\alpha^2d\boldsymbol{X}\bullet d\boldsymbol{Y}}{n\mu}\right)\\
&\quad-\left(\alpha\sum_{j=1}^n(\xi_j+\eta_j)-\alpha^2\frac{\sum_{j=1}^n(\xi_j^2+\eta_j^2)}{2(1-\tau)}\right)\ \text{(by Lemma 7.3)}
\end{aligned}$$

$$
\begin{aligned}
&= \alpha \mathrm{Tr}\ \left(\frac{n+\nu}{n\mu}\boldsymbol{I} - \boldsymbol{Y}^{-1}\boldsymbol{X}^{-1}\right)(d\boldsymbol{X}\boldsymbol{Y} + \boldsymbol{X}d\boldsymbol{Y}) \\
&\quad +\alpha^2\left(\frac{(n+\nu)d\boldsymbol{X} \bullet d\boldsymbol{Y}}{n\mu} + \frac{\sum_{j=1}^n(\xi_j^2+\eta_j^2)}{2(1-\tau)}\right).
\end{aligned}
$$

2. By the definition of G_1,

$$
\begin{aligned}
G_1(d\boldsymbol{X}, d\boldsymbol{Y}) &= \mathrm{Tr}\ \left(\frac{n+\nu}{n\mu}\boldsymbol{I} - \boldsymbol{Y}^{-1}\boldsymbol{X}^{-1}\right)(d\boldsymbol{X}\boldsymbol{Y} + \boldsymbol{X}d\boldsymbol{Y}) \\
&= \mathrm{Tr}\ \left(\frac{n+\nu}{n\mu}\boldsymbol{I} - \boldsymbol{Y}^{-1}\boldsymbol{X}^{-1}\right)(d\hat{\boldsymbol{X}}\boldsymbol{Y} + \boldsymbol{X}d\hat{\boldsymbol{Y}}) \\
&= \mathrm{Tr}\ \left(\frac{1}{\beta\mu}\boldsymbol{I} - \boldsymbol{Y}^{-1}\boldsymbol{X}^{-1}\right)(\beta\mu\boldsymbol{I} - \boldsymbol{X}\boldsymbol{Y}) \\
&\qquad (\text{since } \beta = n/(n+\nu) \text{ and } d\hat{\boldsymbol{X}}\boldsymbol{Y} + \boldsymbol{X}d\hat{\boldsymbol{Y}} = \beta\mu\boldsymbol{I} - \boldsymbol{X}\boldsymbol{Y}) \\
&= \mathrm{Tr}\ \left(\frac{1}{\beta\mu}\boldsymbol{I} - (\sqrt{\boldsymbol{X}}\boldsymbol{Y}\sqrt{\boldsymbol{X}})^{-1}\right)(\beta\mu\boldsymbol{I} - \sqrt{\boldsymbol{X}}\boldsymbol{Y}\sqrt{\boldsymbol{X}}) \\
&= \mathrm{Tr}\ \left(\frac{1}{\beta\mu}\sqrt{\boldsymbol{\Lambda}} - \sqrt{\boldsymbol{\Lambda}}^{-1}\right)(\beta\mu\sqrt{\boldsymbol{\Lambda}}^{-1} - \sqrt{\boldsymbol{\Lambda}}) \\
&= -\frac{1}{\beta\mu}\|\boldsymbol{H}(\beta)\|_F^2 \ (\text{by Lemma 7.2}).
\end{aligned}
$$

Thus we have shown (42).

By the definition of G_2, $0 < \lambda_{min} \leq \mu$, and Lemma 7.1, we see that

$$
\begin{aligned}
G_2(d\boldsymbol{X}, d\boldsymbol{Y}) &= \frac{(n+\nu)d\boldsymbol{X} \bullet d\boldsymbol{Y}}{n\mu} + \frac{\sum_{j=1}^n(\xi_j^2+\eta_j^2)}{2(1-\tau)} \\
&\leq \frac{\|\boldsymbol{H}(\beta)\|_F^2}{2(1-\tau)\lambda_{min}} + \frac{n+\nu}{\lambda_{min}n}d\boldsymbol{X} \bullet d\boldsymbol{Y} - \frac{1}{\lambda_{min}(1-\tau)}d\hat{\boldsymbol{X}} \bullet d\hat{\boldsymbol{Y}} \\
&\leq \frac{\|\boldsymbol{H}(\beta)\|_F^2}{2(1-\tau)\lambda_{min}} + \frac{1}{\lambda_{min}}\left(\frac{n+\nu}{n} - \frac{1}{1-\tau}\right)d\boldsymbol{X} \bullet d\boldsymbol{Y}.
\end{aligned}
$$

Here the last inequality is due to the fact that $d\boldsymbol{X} \bullet d\boldsymbol{Y} \leq d\hat{\boldsymbol{X}} \bullet d\hat{\boldsymbol{Y}}$. Thus we obtain the inequality (43).

Finally we prove the inequality (44) under the assumption that $\nu \geq \sqrt{n}$.

$$
\begin{aligned}
&\left(\frac{\sqrt{\lambda_{min}}}{\beta\mu}\|\boldsymbol{H}(\beta)\|_F\right)^2 \\
&= \left(\frac{\sqrt{\lambda_{min}}}{\beta\mu}\|\beta\mu\sqrt{\boldsymbol{\Lambda}}^{-1} - \sqrt{\boldsymbol{\Lambda}}\|_F\right)^2 \ (\text{by Lemma 7.2}) \\
&= \lambda_{min}\left\|\frac{n+\nu}{n\mu}\sqrt{\boldsymbol{\Lambda}} - \sqrt{\boldsymbol{\Lambda}}^{-1}\right\|_F^2 \ (\text{since } \beta = n/(n+\nu)) \\
&\geq \lambda_{min}\left\|\frac{\sqrt{\boldsymbol{\Lambda}}}{\sqrt{n}\mu}\right\|_F^2 + \lambda_{min}\left\|\frac{\sqrt{\boldsymbol{\Lambda}}}{\mu} - \sqrt{\boldsymbol{\Lambda}}^{-1}\right\|_F^2 \\
&\qquad \left(\text{since Tr } \sqrt{\boldsymbol{\Lambda}}\left(\frac{\sqrt{\boldsymbol{\Lambda}}}{\mu} - \sqrt{\boldsymbol{\Lambda}}^{-1}\right) = 0 \text{ and } \nu \geq \sqrt{n}\right)
\end{aligned}
$$

$$\begin{aligned}
&\geq \frac{\lambda_{min}}{\mu} + \lambda_{min} \left| \frac{\sqrt{\lambda_{min}}}{\mu} - \frac{1}{\sqrt{\lambda_{min}}} \right|^2 \\
&= \frac{(\mu/2 - \lambda_{min})^2 + 3\mu^2/4}{\mu^2} \\
&\geq \frac{3}{4}. \qquad \square
\end{aligned}$$

In the two lemmas below, we are concerned with the following hypothesis.

Hypothesis. (See Hypothesis 4.1 of [17].) Let $(\boldsymbol{X}^0, \boldsymbol{Y}^0) \in \mathcal{S}^2_{++}$. There exists a solution $(\boldsymbol{X}^*, \boldsymbol{Y}^*)$ of the SDLCP (1) such that

$$\omega^* \boldsymbol{X}^0 \succeq \boldsymbol{X}^* \;\text{ and } \omega^* \boldsymbol{Y}^0 \succeq \boldsymbol{Y}^* \tag{45}$$

for some $\omega^* \geq 1$.

This hypothesis as well as the lemmas will be utilized in section 8.3 where we present an infeasible interior-point potential-reduction method.

LEMMA 7.5. *Let $(\boldsymbol{X}, \boldsymbol{Y})$ and $(\boldsymbol{X}^0, \boldsymbol{Y}^0) \in \mathcal{S}^2_{++}$. Assume that the hypothesis is true. Then*

$$\begin{aligned}
\|\sqrt{\boldsymbol{X}}(\boldsymbol{Y}^0 - \boldsymbol{Y}^*)\sqrt{\boldsymbol{X}}\|_F &\leq \omega^* \boldsymbol{X} \bullet \boldsymbol{Y}^0, \\
\|\sqrt{\boldsymbol{Y}}(\boldsymbol{X}^0 - \boldsymbol{X}^*)\sqrt{\boldsymbol{Y}}\|_F &\leq \omega^* \boldsymbol{X}^0 \bullet \boldsymbol{Y}.
\end{aligned}$$

Proof. From the assumption,

$$\begin{aligned}
\omega^* \sqrt{\boldsymbol{X}} \boldsymbol{Y}^0 \sqrt{\boldsymbol{X}} + \left(\sqrt{\boldsymbol{X}} \boldsymbol{Y}^0 \sqrt{\boldsymbol{X}} - \sqrt{\boldsymbol{X}} \boldsymbol{Y}^* \sqrt{\boldsymbol{X}} \right) \in \mathcal{S}_{++}, \\
\omega^* \sqrt{\boldsymbol{X}} \boldsymbol{Y}^0 \sqrt{\boldsymbol{X}} - \left(\sqrt{\boldsymbol{X}} \boldsymbol{Y}^0 \sqrt{\boldsymbol{X}} - \sqrt{\boldsymbol{X}} \boldsymbol{Y}^* \sqrt{\boldsymbol{X}} \right) \in \mathcal{S}_{++}.
\end{aligned}$$

Hence we have

$$\begin{aligned}
0 \leq \operatorname{Tr}\; &\left(\omega^* \sqrt{\boldsymbol{X}} \boldsymbol{Y}^0 \sqrt{\boldsymbol{X}} + \left(\sqrt{\boldsymbol{X}} \boldsymbol{Y}^0 \sqrt{\boldsymbol{X}} - \sqrt{\boldsymbol{X}} \boldsymbol{Y}^* \sqrt{\boldsymbol{X}} \right) \right)^T \\
&\left(\omega^* \sqrt{\boldsymbol{X}} \boldsymbol{Y}^0 \sqrt{\boldsymbol{X}} - \left(\sqrt{\boldsymbol{X}} \boldsymbol{Y}^0 \sqrt{\boldsymbol{X}} - \sqrt{\boldsymbol{X}} \boldsymbol{Y}^* \sqrt{\boldsymbol{X}} \right) \right) \\
= (\omega^*)^2 &\|\sqrt{\boldsymbol{X}} \boldsymbol{Y}^0 \sqrt{\boldsymbol{X}}\|_F^2 - \|\sqrt{\boldsymbol{X}} \boldsymbol{Y}^0 \sqrt{\boldsymbol{X}} - \sqrt{\boldsymbol{X}} \boldsymbol{Y}^* \sqrt{\boldsymbol{X}}\|_F^2.
\end{aligned}$$

It follows that $\|\sqrt{\boldsymbol{X}} \boldsymbol{Y}^0 \sqrt{\boldsymbol{X}} - \sqrt{\boldsymbol{X}} \boldsymbol{Y}^* \sqrt{\boldsymbol{X}}\|_F \leq \omega^* \|\sqrt{\boldsymbol{X}} \boldsymbol{Y}^0 \sqrt{\boldsymbol{X}}\|_F$. Since the matrix $\sqrt{\boldsymbol{X}} \boldsymbol{Y}^0 \sqrt{\boldsymbol{X}}$ is symmetric and positive definite, we have

$$\|\sqrt{\boldsymbol{X}} \boldsymbol{Y}^0 \sqrt{\boldsymbol{X}}\|_F \leq \sqrt{\boldsymbol{X}} \bullet \boldsymbol{Y}^0 \sqrt{\boldsymbol{X}} = \boldsymbol{X} \bullet \boldsymbol{Y}^0.$$

Thus the first inequality in the lemma follows. We can derive the second inequality of the lemma similarly. $\square$

LEMMA 7.6. *(Lemmas 5.1 and 5.2 of [17]). Let $(\boldsymbol{X}^0, \boldsymbol{Y}^0) \in \mathcal{S}^2_{++}$. Assume that the hypothesis is true and that $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}^2_{++}$ satisfies*

$$\theta \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0 \leq \xi \boldsymbol{X} \bullet \boldsymbol{Y}, \tag{46}$$

$$(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}^0 + \theta(\boldsymbol{X}^0, \boldsymbol{Y}^0) + (1 - \theta)(\boldsymbol{X}^*, \boldsymbol{Y}^*) \tag{47}$$

for some $\xi \geq 1$ and $\theta \in [0, 1]$. Let λ_{min} be the minimum eigenvalue of the matrix $\boldsymbol{XY}$, $\mu = \boldsymbol{X} \bullet \boldsymbol{Y}/n$, $\sigma = 2\omega^*\xi + 1$, $\zeta = 2 + \omega^*\sigma$. Let $(d\hat{\boldsymbol{X}}, d\hat{\boldsymbol{Y}}) \in \hat{\mathcal{S}}^2$ be a solution of the Newton equation (23) with $\boldsymbol{Q} = \beta\mu\boldsymbol{I} - \boldsymbol{XY}$, and $d\boldsymbol{X} = (d\hat{\boldsymbol{X}} + d\hat{\boldsymbol{X}}^T)/2$ and $d\boldsymbol{Y} = (d\hat{\boldsymbol{Y}} + d\hat{\boldsymbol{Y}}^T)/2$.

1. $\theta(\boldsymbol{X}^0 \bullet \boldsymbol{Y} + \boldsymbol{X} \bullet \boldsymbol{Y}^0) \le \sigma \boldsymbol{X} \bullet \boldsymbol{Y}$.

2. $\|\sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}}\|_F,\ \|\sqrt{\boldsymbol{X}} d\hat{\boldsymbol{Y}} \sqrt{\boldsymbol{Y}}^{-1}\|_F \le \|\boldsymbol{H}(\beta)\|_F + \frac{\omega^* \sigma n \mu}{\sqrt{\lambda_{min}}} \le \frac{\zeta n \mu}{\sqrt{\lambda_{min}}}$.

3. *Let $\xi_1, \xi_2, \ldots, \xi_n$ be the eigenvalues of $\boldsymbol{X}^{-1} d\boldsymbol{X}$ and $\eta_1, \eta_2, \ldots, \eta_n$ be the eigenvalues of $\boldsymbol{Y}^{-1} d\boldsymbol{Y}$. Then $\sum_{j=1}^n \xi_j^2,\ \sum_{j=1}^n \eta_j^2 \le \left(\frac{\zeta n \mu}{\lambda_{min}}\right)^2$.*

Proof. 1. By the assumption, there is a solution $(\boldsymbol{X}^*, \boldsymbol{Y}^*)$ of the SDLCP (1) satisfying

$$\omega^* \boldsymbol{X}^0 \succeq \boldsymbol{X}^* \text{ and } \omega^* \boldsymbol{Y}^0 \succeq \boldsymbol{Y}^*.$$

It follows that

$$\boldsymbol{X}^0 \bullet \boldsymbol{Y}^* \le \omega^* \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0 \text{ and } \boldsymbol{X}^* \bullet \boldsymbol{Y}^0 \le \omega^* \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0. \tag{48}$$

Let $(\boldsymbol{X}', \boldsymbol{Y}') = \theta(\boldsymbol{X}^0, \boldsymbol{Y}^0) + (1-\theta)(\boldsymbol{X}^*, \boldsymbol{Y}^*)$. By the relation (47) which we have assumed, we then see that

$$(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}^0 + \theta(\boldsymbol{X}^0, \boldsymbol{Y}^0) + (1-\theta)(\boldsymbol{X}^*, \boldsymbol{Y}^*) = \mathcal{F}^0 + (\boldsymbol{X}', \boldsymbol{Y}').$$

Hence $(\boldsymbol{X}' - \boldsymbol{X}, \boldsymbol{Y}' - \boldsymbol{Y}) \in \mathcal{F}^0$. By the monotonicity of the affine subspace $\mathcal{F}^0$, $\boldsymbol{X}' \bullet \boldsymbol{Y} + \boldsymbol{X} \bullet \boldsymbol{Y}' \le \boldsymbol{X}' \bullet \boldsymbol{Y}' + \boldsymbol{X} \bullet \boldsymbol{Y}$. It follows that

$$\begin{aligned}
&\theta(\boldsymbol{X}^0 \bullet \boldsymbol{Y} + \boldsymbol{X} \bullet \boldsymbol{Y}^0) \\
&\le \boldsymbol{X}' \bullet \boldsymbol{Y} + \boldsymbol{X} \bullet \boldsymbol{Y}' \text{ (since } \boldsymbol{X}' \succeq \theta \boldsymbol{X}^0 \text{ and } \boldsymbol{Y}' \succeq \theta \boldsymbol{Y}^0) \\
&\le \boldsymbol{X}' \bullet \boldsymbol{Y}' + \boldsymbol{X} \bullet \boldsymbol{Y} \\
&\le \theta^2 \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0 + \theta(1-\theta)(\boldsymbol{X}^0 \bullet \boldsymbol{Y}^* + \boldsymbol{X}^* \bullet \boldsymbol{Y}^0) + \boldsymbol{X} \bullet \boldsymbol{Y} \ \text{ (since } \boldsymbol{X}^* \bullet \boldsymbol{Y}^* = 0) \\
&\le \theta^2 \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0 + 2\theta(1-\theta)\omega^* \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0 + \boldsymbol{X} \bullet \boldsymbol{Y} \ \text{ (by (48))} \\
&\le \left(2\theta\omega^* \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0 + \boldsymbol{X} \bullet \boldsymbol{Y}\right) \ \text{ (since } \omega^* \ge 1 \text{ and } \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0 \ge 0) \\
&\le (2\omega^* \xi \boldsymbol{X} \bullet \boldsymbol{Y} + \boldsymbol{X} \bullet \boldsymbol{Y}) \ \text{ (by (46))} \\
&\le (2\omega^* \xi + 1) \boldsymbol{X} \bullet \boldsymbol{Y}.
\end{aligned}$$

Thus we have shown assertion 1.

2. By assumption (47) and the Newton equation (23) which $(d\hat{\boldsymbol{X}}, d\hat{\boldsymbol{Y}})$ satisfies, we know that

$$\begin{aligned}
&-\theta\left((\boldsymbol{X}^0, \boldsymbol{Y}^0) - (\boldsymbol{X}^*, \boldsymbol{Y}^*)\right) \in \mathcal{F}^0 - (\boldsymbol{X}, \boldsymbol{Y}) + (\boldsymbol{X}^*, \boldsymbol{Y}^*), \\
&(d\hat{\boldsymbol{X}}, d\hat{\boldsymbol{Y}}) \in \mathcal{F}^0 + \tilde{\mathcal{F}}^0 - (\boldsymbol{X}, \boldsymbol{Y}) + (\boldsymbol{X}^*, \boldsymbol{Y}^*).
\end{aligned}$$

Hence, letting

$$(d\hat{\boldsymbol{X}}', d\hat{\boldsymbol{Y}}') = (d\hat{\boldsymbol{X}}, d\hat{\boldsymbol{Y}}) + \theta\left((\boldsymbol{X}^0, \boldsymbol{Y}^0) - (\boldsymbol{X}^*, \boldsymbol{Y}^*)\right) \in \mathcal{F}^0 + \tilde{\mathcal{F}}^0,$$

we obtain the system of equations in the variable matrices $d\hat{\boldsymbol{X}}',\ d\hat{\boldsymbol{Y}}' \in \hat{\mathcal{S}}$:

$$\left\{\begin{array}{l}
(d\hat{\boldsymbol{X}}', d\hat{\boldsymbol{Y}}') \in \mathcal{F}^0 + \tilde{\mathcal{F}}^0, \\
\boldsymbol{X} d\hat{\boldsymbol{Y}}' + d\hat{\boldsymbol{X}}' \boldsymbol{Y} = \boldsymbol{Q}_1 + \boldsymbol{Q}_2 + \boldsymbol{Q}_3,
\end{array}\right. \tag{49}$$

where $\boldsymbol{Q}_1 = \beta\mu \boldsymbol{I} - \boldsymbol{X}\boldsymbol{Y},\ \boldsymbol{Q}_2 = \theta \boldsymbol{X}(\boldsymbol{Y}^0 - \boldsymbol{Y}^*),\ \boldsymbol{Q}_3 = \theta(\boldsymbol{X}^0 - \boldsymbol{X}^*)\boldsymbol{Y}$. We note that the system (49) of equations has a unique solution in view of Corollary 4.3. Let $(d\hat{\boldsymbol{X}}_j, d\hat{\boldsymbol{Y}}_j)$ $(j = 1, 2, 3)$ denote the solution of the following system of equations:

$$(d\hat{\boldsymbol{X}}_j, d\hat{\boldsymbol{Y}}_j) \in \mathcal{F}^0 + \tilde{\mathcal{F}}^0 \text{ and } \boldsymbol{X} d\hat{\boldsymbol{Y}}_j + d\hat{\boldsymbol{X}}_j \boldsymbol{Y} = \boldsymbol{Q}_j.$$

Then the solution $(d\hat{\boldsymbol{X}}', d\hat{\boldsymbol{Y}}')$ of the system (49) of equations can be represented as

$$(d\hat{\boldsymbol{X}}', d\hat{\boldsymbol{Y}}') = (d\hat{\boldsymbol{X}}_1, d\hat{\boldsymbol{Y}}_1) + (d\hat{\boldsymbol{X}}_2, d\hat{\boldsymbol{Y}}_2) + (d\hat{\boldsymbol{X}}_3, d\hat{\boldsymbol{Y}}_3).$$

On the other hand, we see by the definition of $(d\hat{\boldsymbol{X}}', d\hat{\boldsymbol{Y}}')$ that

$$\begin{aligned}
d\hat{\boldsymbol{X}} &= d\hat{\boldsymbol{X}}' - \theta\left(\boldsymbol{X}^0 - \boldsymbol{X}^*\right) \\
&= d\hat{\boldsymbol{X}}_1 + d\hat{\boldsymbol{X}}_2 + d\hat{\boldsymbol{X}}_3 - \theta\left(\boldsymbol{X}^0 - \boldsymbol{X}^*\right) \\
&= d\hat{\boldsymbol{X}}_1 + d\hat{\boldsymbol{X}}_2 + \boldsymbol{Q}_3\boldsymbol{Y}^{-1} - \boldsymbol{X}d\hat{\boldsymbol{Y}}_3\boldsymbol{Y}^{-1} - \theta\left(\boldsymbol{X}^0 - \boldsymbol{X}^*\right) \\
&= d\hat{\boldsymbol{X}}_1 + d\hat{\boldsymbol{X}}_2 + \left(\theta(\boldsymbol{X}^0 - \boldsymbol{X}^*)\boldsymbol{Y}\right)\boldsymbol{Y}^{-1} - \boldsymbol{X}d\hat{\boldsymbol{Y}}_3\boldsymbol{Y}^{-1} - \theta\left(\boldsymbol{X}^0 - \boldsymbol{X}^*\right) \\
&= d\hat{\boldsymbol{X}}_1 + d\hat{\boldsymbol{X}}_2 - \boldsymbol{X}d\hat{\boldsymbol{Y}}_3\boldsymbol{Y}^{-1}, \\
d\hat{\boldsymbol{Y}} &= d\hat{\boldsymbol{Y}}' - \theta\left(\boldsymbol{Y}^0 - \boldsymbol{Y}^*\right) \\
&= d\hat{\boldsymbol{Y}}_1 + d\hat{\boldsymbol{Y}}_2 + d\hat{\boldsymbol{Y}}_3 - \theta\left(\boldsymbol{Y}^0 - \boldsymbol{Y}^*\right) \\
&= d\hat{\boldsymbol{Y}}_1 + \boldsymbol{X}^{-1}\boldsymbol{Q}_2 - \boldsymbol{X}^{-1}d\hat{\boldsymbol{X}}_2\boldsymbol{Y} + d\hat{\boldsymbol{Y}}_3 - \theta\left(\boldsymbol{Y}^0 - \boldsymbol{Y}^*\right) \\
&= d\hat{\boldsymbol{Y}}_1 + \boldsymbol{X}^{-1}\left(\theta\boldsymbol{X}(\boldsymbol{Y}^0 - \boldsymbol{Y}^*)\right) - \boldsymbol{X}^{-1}d\hat{\boldsymbol{X}}_2\boldsymbol{Y} + d\hat{\boldsymbol{Y}}_3 - \theta\left(\boldsymbol{Y}^0 - \boldsymbol{Y}^*\right) \\
&= d\hat{\boldsymbol{Y}}_1 - \boldsymbol{X}^{-1}d\hat{\boldsymbol{X}}_2\boldsymbol{Y} + d\hat{\boldsymbol{Y}}_3.
\end{aligned}$$

It follows that

$$\begin{aligned}
\sqrt{\boldsymbol{X}}^{-1}d\hat{\boldsymbol{X}}\sqrt{\boldsymbol{Y}} &= \sqrt{\boldsymbol{X}}^{-1}d\hat{\boldsymbol{X}}_1\sqrt{\boldsymbol{Y}} + \sqrt{\boldsymbol{X}}^{-1}d\hat{\boldsymbol{X}}_2\sqrt{\boldsymbol{Y}} - \sqrt{\boldsymbol{X}}^{-1}(\boldsymbol{X}d\hat{\boldsymbol{Y}}_3\boldsymbol{Y}^{-1})\sqrt{\boldsymbol{Y}} \\
&= \sqrt{\boldsymbol{X}}^{-1}d\hat{\boldsymbol{X}}_1\sqrt{\boldsymbol{Y}} + \sqrt{\boldsymbol{X}}^{-1}d\hat{\boldsymbol{X}}_2\sqrt{\boldsymbol{Y}} - \sqrt{\boldsymbol{X}}d\hat{\boldsymbol{Y}}_3\sqrt{\boldsymbol{Y}}^{-1}, \\
\sqrt{\boldsymbol{X}}d\hat{\boldsymbol{Y}}\sqrt{\boldsymbol{Y}}^{-1} &= \sqrt{\boldsymbol{X}}d\hat{\boldsymbol{Y}}_1\sqrt{\boldsymbol{Y}}^{-1} - \sqrt{\boldsymbol{X}}(\boldsymbol{X}^{-1}d\hat{\boldsymbol{X}}_2\boldsymbol{Y})\sqrt{\boldsymbol{Y}}^{-1} + \sqrt{\boldsymbol{X}}d\hat{\boldsymbol{Y}}_3\sqrt{\boldsymbol{Y}}^{-1} \\
&= \sqrt{\boldsymbol{X}}d\hat{\boldsymbol{Y}}_1\sqrt{\boldsymbol{Y}}^{-1} - \sqrt{\boldsymbol{X}}^{-1}d\hat{\boldsymbol{X}}_2\sqrt{\boldsymbol{Y}} + \sqrt{\boldsymbol{X}}d\hat{\boldsymbol{Y}}_3\sqrt{\boldsymbol{Y}}^{-1}.
\end{aligned}$$

Hence, by item 3 of Lemma 7.1 and the definition of $\boldsymbol{H}(\beta)$,

$$\begin{aligned}
&\|\sqrt{\boldsymbol{X}}^{-1}d\hat{\boldsymbol{X}}\sqrt{\boldsymbol{Y}}\|_F \\
&\le \|\sqrt{\boldsymbol{X}}^{-1}d\hat{\boldsymbol{X}}_1\sqrt{\boldsymbol{Y}}\|_F + \|\sqrt{\boldsymbol{X}}^{-1}d\hat{\boldsymbol{X}}_2\sqrt{\boldsymbol{Y}}\|_F + \|\sqrt{\boldsymbol{X}}d\hat{\boldsymbol{Y}}_3\sqrt{\boldsymbol{Y}}^{-1}\|_F \\
&\le \|\sqrt{\boldsymbol{X}}^{-1}\boldsymbol{Q}_1\sqrt{\boldsymbol{Y}}^{-1}\|_F + \|\sqrt{\boldsymbol{X}}^{-1}\boldsymbol{Q}_2\sqrt{\boldsymbol{Y}}^{-1}\|_F + \|\sqrt{\boldsymbol{X}}^{-1}\boldsymbol{Q}_3\sqrt{\boldsymbol{Y}}^{-1}\|_F \\
&\le \|\boldsymbol{H}(\beta)\|_F + \|\sqrt{\boldsymbol{X}}^{-1}\boldsymbol{Q}_2\sqrt{\boldsymbol{Y}}^{-1}\|_F + \|\sqrt{\boldsymbol{X}}^{-1}\boldsymbol{Q}_3\sqrt{\boldsymbol{Y}}^{-1}\|_F, \\
&\|\sqrt{\boldsymbol{X}}d\hat{\boldsymbol{Y}}\sqrt{\boldsymbol{Y}}^{-1}\|_F \\
&\le \|\sqrt{\boldsymbol{X}}d\hat{\boldsymbol{Y}}_1\sqrt{\boldsymbol{Y}}^{-1}\|_F + \|\sqrt{\boldsymbol{X}}^{-1}d\hat{\boldsymbol{X}}_2\sqrt{\boldsymbol{Y}}\|_F + \|\sqrt{\boldsymbol{X}}d\hat{\boldsymbol{Y}}_3\sqrt{\boldsymbol{Y}}^{-1}\|_F \\
&\le \|\sqrt{\boldsymbol{X}}^{-1}\boldsymbol{Q}_1\sqrt{\boldsymbol{Y}}^{-1}\|_F + \|\sqrt{\boldsymbol{X}}^{-1}\boldsymbol{Q}_2\sqrt{\boldsymbol{Y}}^{-1}\|_F + \|\sqrt{\boldsymbol{X}}^{-1}\boldsymbol{Q}_3\sqrt{\boldsymbol{Y}}^{-1}\|_F \\
&\le \|\boldsymbol{H}(\beta)\|_F + \|\sqrt{\boldsymbol{X}}^{-1}\boldsymbol{Q}_2\sqrt{\boldsymbol{Y}}^{-1}\|_F + \|\sqrt{\boldsymbol{X}}^{-1}\boldsymbol{Q}_3\sqrt{\boldsymbol{Y}}^{-1}\|_F.
\end{aligned}$$

Therefore we have shown

$$\begin{aligned}
(50)\qquad &\|\sqrt{\boldsymbol{X}}^{-1}d\hat{\boldsymbol{X}}\sqrt{\boldsymbol{Y}}\|_F,\ \|\sqrt{\boldsymbol{X}}d\hat{\boldsymbol{Y}}\sqrt{\boldsymbol{Y}}^{-1}\|_F \\
&\quad\le \|\boldsymbol{H}(\beta)\|_F + \|\sqrt{\boldsymbol{X}}^{-1}\boldsymbol{Q}_2\sqrt{\boldsymbol{Y}}^{-1}\|_F + \|\sqrt{\boldsymbol{X}}^{-1}\boldsymbol{Q}_3\sqrt{\boldsymbol{Y}}^{-1}\|_F.
\end{aligned}$$

We now evaluate $\|\sqrt{\boldsymbol{X}}^{-1}\boldsymbol{Q}_2\sqrt{\boldsymbol{Y}}^{-1}\|_F$.

$$\|\sqrt{\boldsymbol{X}}^{-1}\boldsymbol{Q}_2\sqrt{\boldsymbol{Y}}^{-1}\|_F$$

$$
\begin{aligned}
&= \left\| \sqrt{\boldsymbol{X}}^{-1} \left(\theta \boldsymbol{X}(\boldsymbol{Y}^0 - \boldsymbol{Y}^*) \right) \sqrt{\boldsymbol{Y}}^{-1} \right\|_F \\
&= \theta \left(\mathrm{Tr}\ (\sqrt{\boldsymbol{X}}(\boldsymbol{Y}^0 - \boldsymbol{Y}^*)\sqrt{\boldsymbol{X}})(\sqrt{\boldsymbol{X}}\boldsymbol{Y}\sqrt{\boldsymbol{X}})^{-1}(\sqrt{\boldsymbol{X}}(\boldsymbol{Y}^0 - \boldsymbol{Y}^*)\sqrt{\boldsymbol{X}}) \right)^{1/2} \\
&\le \frac{\theta}{\sqrt{\lambda_{min}}} \|\sqrt{\boldsymbol{X}}(\boldsymbol{Y}^0 - \boldsymbol{Y}^*)\sqrt{\boldsymbol{X}}\|_F \\
&= \frac{\theta\omega^*}{\sqrt{\lambda_{min}}} \boldsymbol{X} \bullet \boldsymbol{Y}^0 \text{ (by Lemma 7.5).}
\end{aligned}
$$

Thus we have shown that

$$
\|\sqrt{\boldsymbol{X}}^{-1} \boldsymbol{Q}_2 \sqrt{\boldsymbol{Y}}^{-1}\|_F \le \frac{\theta\omega^*}{\sqrt{\lambda_{min}}} \boldsymbol{X} \bullet \boldsymbol{Y}^0. \tag{51}
$$

Similarly we can prove that

$$
\|\sqrt{\boldsymbol{X}}^{-1} \boldsymbol{Q}_3 \sqrt{\boldsymbol{Y}}^{-1}\|_F \le \frac{\theta\omega^*}{\sqrt{\lambda_{min}}} \boldsymbol{X}^0 \bullet \boldsymbol{Y}. \tag{52}
$$

Therefore

$$
\begin{aligned}
&\|\sqrt{\boldsymbol{X}}^{-1} d\hat{\mathbf{X}} \sqrt{\boldsymbol{Y}}\|_F,\ \|\sqrt{\boldsymbol{X}} d\hat{\mathbf{Y}} \sqrt{\boldsymbol{Y}}^{-1}\|_F \\
&\le \|\boldsymbol{H}(\beta)\|_F + \|\sqrt{\boldsymbol{X}}^{-1} \boldsymbol{Q}_2 \sqrt{\boldsymbol{Y}}^{-1}\|_F + \|\sqrt{\boldsymbol{X}}^{-1} \boldsymbol{Q}_3 \sqrt{\boldsymbol{Y}}^{-1}\|_F \quad \text{(by (50))} \\
&\le \|\boldsymbol{H}(\beta)\|_F + \frac{\theta\omega^*}{\sqrt{\lambda_{min}}} (\boldsymbol{X} \bullet \boldsymbol{Y}^0 + \boldsymbol{X}^0 \bullet \boldsymbol{Y}) \quad \text{(by (51) and (52))} \\
&\le \|\boldsymbol{H}(\beta)\|_F + \frac{\omega^* \sigma n \mu}{\sqrt{\lambda_{min}}} \quad \text{(by assertion 1)} \\
&\le \frac{\sqrt{2} n \mu}{\sqrt{\lambda_{min}}} + \frac{\omega^* \sigma n \mu}{\sqrt{\lambda_{min}}} \quad \text{(by Lemma 7.2)} \\
&\le \frac{\zeta n \mu}{\sqrt{\lambda_{min}}}.
\end{aligned}
$$

3. The assertion follows from assertion 2 of both Lemma 7.6 and 7.1. □

By the argument above, it is easily seen that if $(\boldsymbol{X}^0, \boldsymbol{Y}^0) \in \mathcal{F}_{++}$ then we can take $\theta = 0$, which implies that $\boldsymbol{Q}_2 = \boldsymbol{Q}_3 = \boldsymbol{O}$. Hence we can obtain the better evaluation of $\|\sqrt{\boldsymbol{X}}^{-1} d\hat{\mathbf{X}} \sqrt{\boldsymbol{Y}}\|_F,\ \|\sqrt{\boldsymbol{X}} d\hat{\mathbf{Y}} \sqrt{\boldsymbol{Y}}^{-1}\|_F$ in assertion 2 of Lemma 7.6. We state this result as a corollary.

COROLLARY 7.7. *Suppose that all the assumptions of Lemma* 7.6 *hold. In addition, let* $(\boldsymbol{X}^0, \boldsymbol{Y}^0) \in \mathcal{F}_{++}$. *Then* $\|\sqrt{\boldsymbol{X}}^{-1} d\hat{\mathbf{X}} \sqrt{\boldsymbol{Y}}\|_F,\ \|\sqrt{\boldsymbol{X}} d\hat{\mathbf{Y}} \sqrt{\boldsymbol{Y}}^{-1}\|_F \le \|\boldsymbol{H}(\beta)\|_F$.

8. Some interior-point methods. In this section we present three types of interior-point methods, a central trajectory following method, a potential-reduction method, and an infeasible interior-point potential-reduction method as special cases of the generic IP method.

8.1. A central trajectory following method. This method is based on the $O(\sqrt{n}L)$ iteration interior-point method proposed by Kojima–Mizuno–Yoshise [21] for the monotone LCP (8) in the Euclidean space. A horn neighborhood of the central trajectory is defined as

$$
\mathcal{N}(\gamma) = \left\{ \begin{array}{l} (\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}_{++} : \ \left(\displaystyle\sum_{j=1}^{n} (\lambda_j - \mu)^2 \right)^{1/2} \le \gamma\mu, \quad \text{where } \mu = \dfrac{\boldsymbol{X} \bullet \boldsymbol{Y}}{n}, \\ \qquad\qquad\qquad\quad \text{and } \lambda_1, \dots, \lambda_n \ \text{ denote the eigenvalues of } \boldsymbol{X}\boldsymbol{Y} \end{array} \right\}
$$

$$= \left\{ (\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}_{++} : \|\sqrt{\boldsymbol{X}}\boldsymbol{Y}\sqrt{\boldsymbol{X}} - \mu \boldsymbol{I}\|_F \leq \gamma\mu, \quad \text{where } \mu = \frac{\boldsymbol{X} \bullet \boldsymbol{Y}}{n} \right\}.$$

Here $\gamma > 0$ denotes a parameter which determines the width of the neighborhood $\mathcal{N}(\gamma)$. We obtain the central trajectory following method by imposing the following additional restrictions on the generic IP method:

- Let $\tilde{\mathcal{F}}^0 = \boldsymbol{O} \times \tilde{\mathcal{S}}$.
- Let $0 < \gamma \leq 0.1$. Choose an initial point $(\boldsymbol{X}^0, \boldsymbol{Y}^0) \in \mathcal{N}(\gamma)$ in Step 0.
- Let $\beta = 1 - \gamma/\sqrt{n}$ in Step 2.
- Let $\alpha = 1$ in Step 5.

The first restriction needs some explanation. When we compute the solution $(\hat{d\boldsymbol{X}}, \hat{d\boldsymbol{Y}}) \in \hat{\mathcal{S}}^2$ of the Newton equation (26) with $\boldsymbol{Q} = \beta\mu\boldsymbol{I} - \boldsymbol{X}\boldsymbol{Y}$ in Step 3, the restriction above implies $\hat{d\boldsymbol{X}}$ is symmetric so that $d\boldsymbol{X} = (\hat{d\boldsymbol{X}} + \hat{d\boldsymbol{X}}^T)/2 = \hat{d\boldsymbol{X}}$ in Step 4. The fact that $\hat{d\boldsymbol{X}}$ is symmetric is necessary in our proof of the lemma below. The authors tried to employ a general $n(n-1)/2$-dimensional monotone linear subspace $\tilde{\mathcal{F}}^0$ of $\tilde{\mathcal{S}}^2$, but we had some difficulty deriving the inequality (55) in the general case.

THEOREM 8.1. *Let* $\gamma \in (0,\ 0.1]$. *Suppose that* $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{N}(\gamma)$. *Let* $\beta = 1 - \gamma/\sqrt{n}$ *in Step* 2 *and* $\alpha = 1$ *in Step* 5. *Let* $\bar{\mu} = \bar{\boldsymbol{X}} \bullet \bar{\boldsymbol{Y}}/n$. *Then*

$$\begin{aligned} &(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}) = (\boldsymbol{X}, \boldsymbol{Y}) + (d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{N}(\gamma), \\ &\beta\mu \leq \bar{\mu} \leq \left(1 - \frac{\gamma}{2\sqrt{n}}\right)\mu. \end{aligned} \tag{53}$$

Proof. First note that $\hat{d\boldsymbol{X}} = d\boldsymbol{X}$. By the definition of $(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}})$,

$$\begin{aligned} n\bar{\mu} &= \bar{\boldsymbol{X}} \bullet \bar{\boldsymbol{Y}} \\ &= \boldsymbol{X} \bullet \boldsymbol{Y} + \text{Tr}\,(\boldsymbol{X} d\boldsymbol{Y} + d\boldsymbol{X}\boldsymbol{Y}) + d\boldsymbol{X} \bullet d\boldsymbol{Y} \\ &= \boldsymbol{X} \bullet \boldsymbol{Y} + \text{Tr}\,(\boldsymbol{X} \hat{d\boldsymbol{Y}} + d\boldsymbol{X}\boldsymbol{Y}) + d\boldsymbol{X} \bullet d\boldsymbol{Y} \\ &= n\mu + \text{Tr}\,(\beta\mu\boldsymbol{I} - \boldsymbol{X}\boldsymbol{Y}) + d\boldsymbol{X} \bullet d\boldsymbol{Y} \text{ (by the Newton equation (26))} \\ &= n\mu + n\beta\mu - n\mu + d\boldsymbol{X} \bullet d\boldsymbol{Y} \text{ (since } \mu = \boldsymbol{X} \bullet \boldsymbol{Y}/n) \\ &= n\beta\mu + d\boldsymbol{X} \bullet d\boldsymbol{Y}. \end{aligned}$$

In view of item 3 of Lemma 7.1 with $\boldsymbol{Q} = \beta\mu\boldsymbol{I} - \boldsymbol{X}\boldsymbol{Y}$ and item 2 of Lemma 7.2, we see that

$$0 \leq d\boldsymbol{X} \bullet d\boldsymbol{Y} \leq \frac{\|\boldsymbol{H}(\beta)\|_F^2}{4} \leq \frac{\gamma^2\mu}{1-\gamma}.$$

Hence

$$\beta\mu \leq \bar{\mu} = \beta\mu + \frac{d\boldsymbol{X} \bullet d\boldsymbol{Y}}{n} \leq \left(1 - \frac{\gamma}{2\sqrt{n}}\right)\mu.$$

Thus we have shown (53).

By Lemma 7.1 with $\boldsymbol{Q} = \beta\mu\boldsymbol{I} - \boldsymbol{X}\boldsymbol{Y}$ and Lemma 7.2,

$$\sum_{j=1}^{n}(\xi_j^2 + \eta_j^2) \leq \frac{1}{\lambda_{min}}\|\boldsymbol{H}(\beta)\|_F^2 \leq \frac{4\gamma^2}{(1-\gamma)^2} < 1.$$

Hence Lemma 5.1 together with $(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{F}$ and $(d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{F}^0$ ensure that

$$(\bar{\boldsymbol{X}},\bar{\boldsymbol{Y}}) = (\boldsymbol{X},\boldsymbol{Y}) + (d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{F}_{++}. \tag{54}$$

To complete the proof, we only need to show the inequality

$$\|\bar{\boldsymbol{B}}^T\bar{\boldsymbol{Y}}\bar{\boldsymbol{B}} - \bar{\mu}\boldsymbol{I}\|_F \leq \gamma\bar{\mu} \tag{55}$$

for some $\bar{\boldsymbol{B}}$ such that $\bar{\boldsymbol{X}} = \bar{\boldsymbol{B}}\bar{\boldsymbol{B}}^T$, since (54) and (55) imply that $(\bar{\boldsymbol{X}},\bar{\boldsymbol{Y}}) \in \mathcal{N}(\gamma)$. By Lemma 7.1, with $\boldsymbol{Q} = \beta\mu\boldsymbol{I} - \boldsymbol{X}\boldsymbol{Y}$, Lemma 7.2, and $0 < \gamma \leq 0.1$,

$$\|\sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X}\sqrt{\boldsymbol{X}}^{-1}\|_F \leq \frac{1}{\sqrt{(1-\gamma)\mu}} \cdot \frac{2\gamma\sqrt{\mu}}{\sqrt{(1-\gamma)}} = \frac{2\gamma}{1-\gamma} < 1, \tag{56}$$

$$\|\sqrt{\boldsymbol{X}} d\hat{\boldsymbol{Y}}\sqrt{\boldsymbol{X}}\|_F \leq \sqrt{(1+\gamma)\mu} \cdot \frac{2\gamma\sqrt{\mu}}{\sqrt{(1-\gamma)}} = \frac{2\mu\gamma\sqrt{1+\gamma}}{\sqrt{1-\gamma}}. \tag{57}$$

We see from the definition of $(\bar{\boldsymbol{X}},\bar{\boldsymbol{Y}})$ that

$$\bar{\boldsymbol{X}} = \boldsymbol{X} + d\boldsymbol{X} = \sqrt{\boldsymbol{X}}\sqrt{\boldsymbol{X}} + d\boldsymbol{X} = \sqrt{\boldsymbol{X}}(\boldsymbol{I} + \sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X}\sqrt{\boldsymbol{X}}^{-1})\sqrt{\boldsymbol{X}}.$$

Since the inequality (56) implies that the absolute values of all the eigenvalues of the symmetric matrix $\sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X}\sqrt{\boldsymbol{X}}^{-1}$ are less than 1, the symmetric matrix $\boldsymbol{I} + \sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X}\sqrt{\boldsymbol{X}}^{-1}$ is positive definite. Hence it can be represented as $\boldsymbol{I}+\sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X}\sqrt{\boldsymbol{X}}^{-1} = \boldsymbol{P}\boldsymbol{\varXi}\boldsymbol{P}^T$ for an orthogonal matrix $\boldsymbol{P}$ and a diagonal matrix $\boldsymbol{\varXi} =$ diag $(\bar{\xi}_1, \bar{\xi}_2, \ldots, \bar{\xi}_n)$ of its eigenvalues $\bar{\xi}_1, \bar{\xi}_2, \ldots, \bar{\xi}_n$ such that

$$\frac{1-3\gamma}{1-\gamma} = 1 - \frac{2\gamma}{1-\gamma} \leq \bar{\xi}_j \leq 1 + \frac{2\gamma}{1-\gamma} = \frac{1+\gamma}{1-\gamma} \text{ for every } j = 1, 2, \ldots, n. \tag{58}$$

Letting $\bar{\boldsymbol{B}} = \sqrt{\boldsymbol{X}}\boldsymbol{P}\sqrt{\boldsymbol{\varXi}}$, we obtain

$$\bar{\boldsymbol{X}} = \sqrt{\boldsymbol{X}}(\boldsymbol{I} + \sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X}\sqrt{\boldsymbol{X}}^{-1})\sqrt{\boldsymbol{X}} = \sqrt{\boldsymbol{X}}\boldsymbol{P}\boldsymbol{\varXi}\boldsymbol{P}^T\sqrt{\boldsymbol{X}} = \bar{\boldsymbol{B}}\bar{\boldsymbol{B}}^T.$$

We also see from the Newton equation (26) that

$$\bar{\boldsymbol{X}}(\boldsymbol{Y} + d\hat{\boldsymbol{Y}}) - \beta\mu\boldsymbol{I} = (\boldsymbol{X} + d\boldsymbol{X})(\boldsymbol{Y} + d\hat{\boldsymbol{Y}}) - \beta\mu\boldsymbol{I} = d\boldsymbol{X} d\hat{\boldsymbol{Y}};$$

hence $\bar{\boldsymbol{B}}^T(\boldsymbol{Y} + d\hat{\boldsymbol{Y}})\bar{\boldsymbol{B}} - \beta\mu\boldsymbol{I} = \bar{\boldsymbol{B}}^{-1} d\boldsymbol{X} d\hat{\boldsymbol{Y}}\bar{\boldsymbol{B}}$. Now we are ready to evaluate $\|\bar{\boldsymbol{B}}^T\bar{\boldsymbol{Y}}\bar{\boldsymbol{B}} - \bar{\mu}\boldsymbol{I}\|_F$ to derive the inequality (55):

$$\begin{aligned}
&\|\bar{\boldsymbol{B}}^T\bar{\boldsymbol{Y}}\bar{\boldsymbol{B}} - \bar{\mu}\boldsymbol{I}\|_F \\
&\leq \|\bar{\boldsymbol{B}}^T\bar{\boldsymbol{Y}}\bar{\boldsymbol{B}} - \beta\mu\boldsymbol{I}\|_F \text{ (since } \|\bar{\boldsymbol{B}}^T\bar{\boldsymbol{Y}}\bar{\boldsymbol{B}} - \bar{\mu}\boldsymbol{I}\|_F = \min\nolimits_{\nu\in R} \|\bar{\boldsymbol{B}}^T\bar{\boldsymbol{Y}}\bar{\boldsymbol{B}} - \nu\boldsymbol{I}\|_F) \\
&= \|\bar{\boldsymbol{B}}^T \left(\frac{\boldsymbol{Y} + d\hat{\boldsymbol{Y}} + \boldsymbol{Y}^T + d\hat{\boldsymbol{Y}}^T}{2}\right)\bar{\boldsymbol{B}} - \beta\mu\boldsymbol{I}\|_F \\
&\leq \frac{\|\bar{\boldsymbol{B}}^T(\boldsymbol{Y} + d\hat{\boldsymbol{Y}})\bar{\boldsymbol{B}} - \beta\mu\boldsymbol{I}\|_F}{2} + \frac{\|\bar{\boldsymbol{B}}^T(\boldsymbol{Y} + d\hat{\boldsymbol{Y}})^T\bar{\boldsymbol{B}} - \beta\mu\boldsymbol{I}\|_F}{2} \\
&= \|\bar{\boldsymbol{B}}^T(\boldsymbol{Y} + d\hat{\boldsymbol{Y}})\bar{\boldsymbol{B}} - \beta\mu\boldsymbol{I}\|_F \\
&= \|\bar{\boldsymbol{B}}^{-1} d\boldsymbol{X} d\hat{\boldsymbol{Y}}\bar{\boldsymbol{B}}\|_F
\end{aligned}$$

$$
\begin{aligned}
&= \|\sqrt{\boldsymbol{\Xi}}^{-1}\boldsymbol{P}^T\sqrt{\boldsymbol{X}}^{-1}d\boldsymbol{X}\sqrt{\boldsymbol{X}}^{-1}\sqrt{\boldsymbol{X}}d\hat{\boldsymbol{Y}}\sqrt{\boldsymbol{X}}\boldsymbol{P}\sqrt{\boldsymbol{\Xi}}\|_F \text{ (since } \bar{\boldsymbol{B}} = \sqrt{\boldsymbol{X}}\boldsymbol{P}\sqrt{\boldsymbol{\Xi}}\text{)}\\
&\leq \frac{\sqrt{(1+\gamma)/(1-\gamma)}}{\sqrt{(1-3\gamma)/(1-\gamma)}}\|\boldsymbol{P}^T\sqrt{\boldsymbol{X}}^{-1}d\boldsymbol{X}\sqrt{\boldsymbol{X}}^{-1}\sqrt{\boldsymbol{X}}d\hat{\boldsymbol{Y}}\sqrt{\boldsymbol{X}}\boldsymbol{P}\|_F \text{ (by (58))}\\
&\leq \frac{\sqrt{1+\gamma}}{\sqrt{1-3\gamma}}\|\sqrt{\boldsymbol{X}}^{-1}d\boldsymbol{X}\sqrt{\boldsymbol{X}}^{-1}\|_F \cdot \|\sqrt{\boldsymbol{X}}d\hat{\boldsymbol{Y}}\sqrt{\boldsymbol{X}}\|_F\\
&\leq \frac{\sqrt{1+\gamma}}{\sqrt{1-3\gamma}} \cdot \frac{2\gamma}{1-\gamma} \cdot \frac{2\mu\gamma\sqrt{1+\gamma}}{\sqrt{1-\gamma}} \quad \text{(by (56) and (57))}\\
&= \frac{4\gamma^2(1+\gamma)\mu}{(1-\gamma)^{3/2}\sqrt{1-3\gamma}}\\
&\leq \frac{4\gamma^2(1+\gamma)\bar{\mu}}{(1-\gamma)^{3/2}\sqrt{1-3\gamma}\beta} \quad \text{(since } \beta\mu \leq \bar{\mu} \text{ by (53))}\\
&\leq \frac{4\gamma^2(1+\gamma)}{(1-\gamma)^{3/2}\sqrt{1-3\gamma}(1-\gamma)}\bar{\mu} \quad \text{(since } 1-\gamma \leq \beta\text{)}.
\end{aligned}
$$

Hence

$$\|\bar{\boldsymbol{B}}^T\bar{\boldsymbol{Y}}\bar{\boldsymbol{B}} - \bar{\mu}\boldsymbol{I}\|_F \leq \frac{4\gamma(1+\gamma)}{(1-\gamma)^{5/2}\sqrt{1-3\gamma}}\gamma\bar{\mu} \leq \gamma\bar{\mu}.$$

Here the last inequality follows from $\gamma \in (0, 0.1]$. Thus we have shown the inequality (55). □

Let $\epsilon > 0$. In view of the theorem above, the central trajectory following method generates a sequence $\{(\boldsymbol{X}^r, \boldsymbol{Y}^r)\}$ such that

$$(\boldsymbol{X}^r, \boldsymbol{Y}^r) \in \mathcal{N}(\gamma) \text{ and } \boldsymbol{X}^r \bullet \boldsymbol{Y}^r \leq \left(1 - \frac{\gamma}{2\sqrt{n}}\right)^r \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0$$

for every $r = 0, 1, \ldots$. Hence if

$$r \geq \frac{2\sqrt{n}}{\gamma}\log\frac{\boldsymbol{X}^0 \bullet \boldsymbol{Y}^0}{\epsilon} = O\left(\sqrt{n}\log\frac{\boldsymbol{X}^0 \bullet \boldsymbol{Y}^0}{\epsilon}\right)$$

then $(\boldsymbol{X}^r, \boldsymbol{Y}^r)$ gives an approximate solution of the SDLCP (1) in symmetric matrices such that

$$(\boldsymbol{X}^r, \boldsymbol{Y}^r) \in \mathcal{F}_{++} \text{ and } \boldsymbol{X}^r \bullet \boldsymbol{Y}^r \leq \epsilon. \tag{59}$$

8.2. A potential-reduction method. For every $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}_{++}$, define the potential function

$$f(\boldsymbol{X}, \boldsymbol{Y}) = (n + \nu)\log \boldsymbol{X} \bullet \boldsymbol{Y} - \log\det \boldsymbol{X}\boldsymbol{Y} - n\log n.$$

Here $\nu \geq 0$ is a parameter. This potential function is the same as the one used in the paper [44] by Vandenberghe and Boyd. Our potential-reduction method described below is different from their method in search directions. Our method may be regarded as an extension of the Kojima–Mizuno–Yoshise potential-reduction method [22] for the monotone LCP (8) in the Euclidean space (see also [19]).

The potential function f defined above enjoys similar properties as the potential function [40, 42] used for the monotone LCP (8). In particular, if we rewrite the potential function f as

$$\begin{aligned} f(\boldsymbol{X},\boldsymbol{Y}) &= \nu f_{cp}(\boldsymbol{X},\boldsymbol{Y}) + f_{cen}(\boldsymbol{X},\boldsymbol{Y}), \\ f_{cp}(\boldsymbol{X},\boldsymbol{Y}) &= \log \boldsymbol{X} \bullet \boldsymbol{Y}, \\ f_{cen}(\boldsymbol{X},\boldsymbol{Y}) &= n \log \boldsymbol{X} \bullet \boldsymbol{Y} - \log\det \boldsymbol{X}\boldsymbol{Y} - n\log n, \end{aligned}$$

we have

$$\begin{aligned} & f_{cen}(\boldsymbol{X},\boldsymbol{Y}) \geq 0 \text{ for every } (\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{F}_{++}, \\ & f_{cen}(\boldsymbol{X},\boldsymbol{Y}) = 0 \text{ if and only if } (\boldsymbol{X},\boldsymbol{Y}) \text{ lies in the central trajectory } \mathcal{C}. \end{aligned} \tag{60}$$

See the paper [44].

We impose the following restrictions on the generic IP method:

- Choose an initial point $(\boldsymbol{X}^0, \boldsymbol{Y}^0) \in \mathcal{F}_{++}$ in Step 0.
- Let $\beta = n/(n+\nu)$ in Step 2.
- Let

$$\left\{ \begin{array}{rcl} 0 & < & \tau \ < \ 1, \\ \boldsymbol{H}(\beta) & = & \beta\mu\sqrt{\boldsymbol{X}}^{-1}\sqrt{\boldsymbol{Y}}^{-1} - \sqrt{\boldsymbol{X}}\sqrt{\boldsymbol{Y}}, \\ \lambda_{min} & = & \min\{\lambda_1, \lambda_2, \ldots, \lambda_n\}, \end{array} \right. \tag{61}$$

where $\lambda_1, \lambda_2, \ldots, \lambda_n$ denote the eigenvalues of the matrix $\boldsymbol{X}\boldsymbol{Y}$. Take a step size parameter α in Step 5 such that $\alpha = \tau\sqrt{\lambda_{min}}/\|\boldsymbol{H}(\beta)\|_F$.

We remark here that if $\boldsymbol{X} = \boldsymbol{L}\boldsymbol{L}^T$, $\boldsymbol{L} \in \hat{\mathcal{S}}$, $\boldsymbol{Y} = \boldsymbol{M}\boldsymbol{M}^T$, and $\boldsymbol{M} \in \hat{\mathcal{S}}$, then

$$\|\boldsymbol{H}(\beta)\|_F = \|\beta\mu\boldsymbol{L}^{-1}\boldsymbol{M}^{-T} - \boldsymbol{L}^T\boldsymbol{M}\|_F.$$

This makes the computation of the step length $\alpha = \tau\sqrt{\lambda_{min}}/\|\boldsymbol{H}(\beta)\|_F$ more flexible and efficient.

THEOREM 8.2. *Let $n \geq 3$, $\nu = \sqrt{n}$, $\tau = 0.4$, and $\delta = 0.2$. Suppose that $(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{F}_{++}$. Let $\beta = n/(n+\nu)$ in Step 2 and $\alpha = \tau\sqrt{\lambda_{min}}/\|\boldsymbol{H}(\beta)\|_F$ in Step 5, where τ, λ_{min}, and $\boldsymbol{H}(\beta)$ are given in (61). Then*

$$(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}) = (\boldsymbol{X},\boldsymbol{Y}) + \alpha(d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{F}_{++} \text{ and } f(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}) \leq f(\boldsymbol{X},\boldsymbol{Y}) - \delta.$$

Proof. Let $\xi_1, \xi_2, \ldots, \xi_n$ be the eigenvalues of $\boldsymbol{X}^{-1}d\boldsymbol{X}$ and $\eta_1, \eta_2, \ldots, \eta_n$ be the eigenvalues of $\boldsymbol{Y}^{-1}d\boldsymbol{Y}$. By Lemma 7.1 and $\alpha = \tau\sqrt{\lambda_{min}}/\|\boldsymbol{H}(\beta)\|_F$,

$$\sum_{j=1}^{n} \left((\alpha\xi_j)^2 + (\alpha\eta_j)^2 \right) \leq \alpha^2 \cdot \frac{1}{\lambda_{min}} \|\boldsymbol{H}(\beta)\|_F^2 = \tau^2.$$

Hence $|\alpha\xi_j| \leq \tau = 0.4$ and $|\alpha\eta_j| \leq \tau = 0.4$ for every $j = 1, 2, \ldots, n$. By Lemma 5.1, $(\boldsymbol{X},\boldsymbol{Y}) \in \mathcal{F}$ and $(d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{F}^0$, we obtain that

$$(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}) = (\boldsymbol{X},\boldsymbol{Y}) + \alpha(d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{F}_{++}.$$

It follows from $\tau = 0.4$, $\nu = \sqrt{n}$, and $n \geq 3$ that $(n + \sqrt{n})/n < 1/(1-\tau)$. By Lemma 7.4,

$$G_2(d\boldsymbol{X}, d\boldsymbol{Y}) \leq \frac{\|\boldsymbol{H}(\beta)\|_F^2}{2(1-\tau)\lambda_{min}}. \tag{62}$$

Thus we consequently obtain

$$\begin{aligned}
&f(\boldsymbol{X}+\alpha d\boldsymbol{X}, \boldsymbol{Y}+\alpha d\boldsymbol{Y}) - f(\boldsymbol{X},\boldsymbol{Y}) \\
&\leq \alpha G_1(d\boldsymbol{X}, d\boldsymbol{Y}) + \alpha^2 G_2(d\boldsymbol{X}, d\boldsymbol{Y}) \quad \text{(by 1 of Lemma 7.4)} \\
&\leq -\frac{1}{\beta\mu}\|\boldsymbol{H}(\beta)\|_F^2 \alpha + \frac{\|\boldsymbol{H}(\beta)\|^2}{2(1-\tau)\lambda_{min}}\alpha^2 \quad \text{(by (42) and (62))} \\
&= -\frac{\tau\sqrt{\lambda_{min}}}{\beta\mu}\|\boldsymbol{H}(\beta)\|_F + \frac{\tau^2}{2(1-\tau)} \quad (\text{since } \alpha = \tau\sqrt{\lambda_{min}}/\|\boldsymbol{H}(\beta)\|_F) \\
&\leq -\frac{\sqrt{3}\tau}{2} + \frac{\tau^2}{2(1-\tau)} \quad \text{(by 2 of Lemma 7.4)} \\
&\leq -0.2. \ (\text{since } \tau = 0.4).
\end{aligned}$$

This completes the proof of Theorem 8.2. □

Let $\epsilon > 0$. By Theorem 8.2, the potential-reduction method generates a sequence $\{(\boldsymbol{X}^r, \boldsymbol{Y}^r)\}$ such that $(\boldsymbol{X}^r, \boldsymbol{Y}^r) \in \mathcal{F}_{++}$ and $f(\boldsymbol{X}^r, \boldsymbol{Y}^r) \leq f(\boldsymbol{X}^0, \boldsymbol{Y}^0) - r\delta$ for every $r = 1, 2, \ldots$. Hence if $r \geq (f(\boldsymbol{X}^0, \boldsymbol{Y}^0) - \sqrt{n}\log\epsilon)/\delta$, then $(\boldsymbol{X}^r, \boldsymbol{Y}^r)$ gives an approximate solution of the SDLCP (1) satisfying (59). If in addition $f_{cen}(\boldsymbol{X}^0, \boldsymbol{Y}^0)$ is bounded by a constant independent of n, the right-hand side of the inequality above is of $O(\sqrt{n}\log(\boldsymbol{X}^0 \bullet \boldsymbol{Y}^0/\epsilon))$, the same order as the one in the case of the central trajectory following method described in section 8.1.

8.3. An infeasible interior-point potential-reduction method. The IIP potential-reduction method presented below is based on the $O(n^{2.5}L)$ iteration constrained potential-reduction algorithm (Algorithm I of [32]) for linear programs and its modification [17]. We can start the IIP potential-reduction method from any $(\boldsymbol{X}^0, \boldsymbol{Y}^0) \in \mathcal{S}^2_{++}$. Before running the method, we assume that the hypothesis given in section 7 holds.

As we will see below, the IIP potential-reduction method either detects in a finite number of iterations that the hypothesis is false (i.e., there is no solution $(\boldsymbol{X}^*, \boldsymbol{Y}^*)$ of the SDLCP (1) satisfying (45)) or reduces the potential function by at least a given constant δ at every iteration.

We impose some additional requirements on Steps 0, 2, and 5 of the generic IP method to describe the IIP potential-reduction method:

IIP potential-reduction method.

Step 0_{iip}: Choose an initial point $(\boldsymbol{X}^0, \boldsymbol{Y}^0) \in \mathcal{S}^2_{++}$ and two parameters $\nu \geq \sqrt{n}$, $\xi \geq 1$. Let $\sigma = 2\omega^*\xi + 1, \zeta = 2 + \omega^*\sigma$, and $\delta = 1/(10\zeta^2(n+\nu)^2)$. Let $\theta^0 = 1$ and $r = 0$.

Step 1: Let $(\boldsymbol{X}, \boldsymbol{Y}) = (\boldsymbol{X}^r, \boldsymbol{Y}^r)$ and

$$\mu = \frac{\boldsymbol{X} \bullet \boldsymbol{Y}}{n}.$$

Step 2_{iip}: If

$$\theta^r(\boldsymbol{X}^0 \bullet \boldsymbol{Y} + \boldsymbol{X} \bullet \boldsymbol{Y}^0) \leq \sigma \boldsymbol{X} \bullet \boldsymbol{Y} \tag{63}$$

does not hold then stop; in this case there is no solution of the SDLCP (1) satisfying (45) (see item 2 of Theorem 8.3). Otherwise let $\beta = n/(n+\nu)$.

Step 3: Compute a solution $(\hat{d\boldsymbol{X}}, \hat{d\boldsymbol{Y}}) \in \hat{\mathcal{S}}^2$ of the system (23) of equations with $\boldsymbol{Q} = \beta\mu\boldsymbol{I} - \boldsymbol{X}\boldsymbol{Y}$.

Step 4: Let $d\boldsymbol{X} = (d\hat{\boldsymbol{X}} + d\hat{\boldsymbol{X}}^T)/2$ and $d\boldsymbol{Y} = (d\hat{\boldsymbol{Y}} + d\hat{\boldsymbol{Y}}^T)/2$.

Step 5_{iip}: Choose a step size parameter $\alpha \geq 0$ such that

$$(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}) = (\boldsymbol{X}, \boldsymbol{Y}) + \alpha(d\boldsymbol{X}, d\boldsymbol{Y}) \in \mathcal{S}^2_{++}, \tag{64}$$

$$(1-\alpha)\theta^r \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0 \leq \xi \bar{\boldsymbol{X}} \bullet \bar{\boldsymbol{Y}}, \tag{65}$$

$$f(\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}) \leq f(\boldsymbol{X}, \boldsymbol{Y}) - \delta. \tag{66}$$

Let $(\boldsymbol{X}^{r+1}, \boldsymbol{Y}^{r+1}) = (\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}})$ and $\theta^{r+1} = (1-\alpha)\theta^r$.

Step 6: Replace r by $r+1$ and go to Step 1.

THEOREM 8.3. *Let $(\boldsymbol{X}, \boldsymbol{Y}) = (\boldsymbol{X}^r, \boldsymbol{Y}^r) \in \mathcal{S}^2_{++}$ be the rth iterate generated by the IIP potential-reduction method. Let $\mu = \boldsymbol{X} \bullet \boldsymbol{Y}/n$ and let λ_{min} denote the minimum eigenvalue of the matrix $\boldsymbol{XY}$.*

1. *Let $(\boldsymbol{X}', \boldsymbol{Y}')$ be a pair of matrices in $\mathcal{F}$; for example, we can take an orthogonal projection of $(\boldsymbol{X}^0, \boldsymbol{Y}^0)$ onto $\mathcal{F}$, or a solution $(\boldsymbol{X}^*, \boldsymbol{Y}^*)$ of the SDLCP* (1) *satisfying* (45) *under the hypothesis in section 7. Then*

$$\theta^r \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0 \leq \xi \boldsymbol{X} \bullet \boldsymbol{Y}, \tag{67}$$

$$(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F} + \theta^r \left((\boldsymbol{X}^0, \boldsymbol{Y}^0) - (\boldsymbol{X}', \boldsymbol{Y}')\right). \tag{68}$$

2. *Assume that inequality* (63) *holds at Step 2_{iip}. Then $\alpha = \lambda_{min}/(5\zeta^2(n+\nu)n\mu)$ fulfills all the requirements* (64), (65), *and* (66) *in Step 5_{iip} for a legitimate step size parameter.*

3. *If the inequality* (63) *does not hold at Step 2_{iip} then there is no solution $(\boldsymbol{X}^*, \boldsymbol{Y}^*)$ of the SDLCP* (1) *satisfying* (45).

In order to prove the assertion 1, we need the following lemma.

LEMMA 8.4. *Let $(\boldsymbol{X}, \boldsymbol{Y}) = (\boldsymbol{X}^r, \boldsymbol{Y}^r) \in \mathcal{S}^2_{++}$ be the rth iterate generated by the IIP potential-reduction method, and let $\theta = \theta^r$. Let $(\boldsymbol{X}', \boldsymbol{Y}')$ be a pair of matrices in $\mathcal{F}$. Then*

$$\theta \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0 \leq \xi \boldsymbol{X} \bullet \boldsymbol{Y}, \tag{69}$$

$$(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{F}^0 + \theta(\boldsymbol{X}^0, \boldsymbol{Y}^0) + (1-\theta)(\boldsymbol{X}', \boldsymbol{Y}'). \tag{70}$$

Proof. By the construction, the inequality (69) holds with $\theta = \theta^r \in [0,1]$. Hence it suffices to show by induction that $(\boldsymbol{X}^r, \boldsymbol{Y}^r) \in \mathcal{F}^0 + \theta^r(\boldsymbol{X}^0, \boldsymbol{Y}^0) + (1-\theta^r)(\boldsymbol{X}', \boldsymbol{Y}')$. When $r = 0$, the relation above obviously holds because $\theta^0 = 1$. Assume that the relation holds for $r = k$. Then

$$\begin{aligned}
&(\boldsymbol{X}^{k+1}, \boldsymbol{Y}^{k+1}) \\
&= (1-\alpha)(\boldsymbol{X}^k, \boldsymbol{Y}^k) + \alpha(\boldsymbol{X}^k + d\boldsymbol{X}, \boldsymbol{Y}^k + d\boldsymbol{Y}) \\
&\in (1-\alpha)\left(\mathcal{F}^0 + \theta^k(\boldsymbol{X}^0, \boldsymbol{Y}^0) + (1-\theta^k)(\boldsymbol{X}', \boldsymbol{Y}')\right) + \alpha\left(\mathcal{F}^0 + (\boldsymbol{X}', \boldsymbol{Y}')\right) \\
&= \mathcal{F}^0 + (1-\alpha)\theta^k(\boldsymbol{X}^0, \boldsymbol{Y}^0) + (1-(1-\alpha)\theta^k)(\boldsymbol{X}', \boldsymbol{Y}') \\
&= \mathcal{F}^0 + \theta^{k+1}(\boldsymbol{X}^0, \boldsymbol{Y}^0) + (1-\theta^{k+1})(\boldsymbol{X}', \boldsymbol{Y}').
\end{aligned}$$

Thus we have shown the desired relation for $r = k+1$. □

Proof of Theorem 8.3. Assertion 1 of the theorem follows from Lemma 8.4 since we can rewrite the relation (70) as the relation (68).

To prove assertion 2 of the theorem, let $\alpha = \lambda_{min}/(5\zeta^2 n(n+\nu)\mu)$. By the definition, $\zeta = 2 + \omega^*\sigma \geq 4$, so that we see $\alpha = \lambda_{min}/(5\zeta^2 n(n+\nu)\mu) \leq 1/80$. By Lemma 7.6, we then see that for every $j = 1, 2, \ldots, n$

$$|\alpha\xi_j|,\ |\alpha\eta_j| \leq \alpha \cdot \frac{\zeta n\mu}{\lambda_{min}} = \frac{\lambda_{min}}{5\zeta^2 n(n+\nu)\mu} \cdot \frac{\zeta n\mu}{\lambda_{min}} \leq \frac{1}{20}.$$

Hence, letting $\tau = 1/20$, we obtain that $|\alpha\xi_j|$, $|\alpha\eta_j| \le \tau$ $(j = 1, 2, \ldots, n)$. By Lemma 5.1, $\boldsymbol{X} + \alpha d\boldsymbol{X} \in \mathcal{S}_{++}$ and $\boldsymbol{Y} + \alpha d\boldsymbol{Y} \in \mathcal{S}_{++}$. Thus we have shown the relation (64).

To derive the inequality (65) we observe that

$$\begin{aligned}
&\xi(\boldsymbol{X} + \alpha d\boldsymbol{X}) \bullet (\boldsymbol{Y} + \alpha d\boldsymbol{Y}) \\
&= \xi\left(\boldsymbol{X} \bullet \boldsymbol{Y} + \alpha \mathrm{Tr}\,(\boldsymbol{X} d\boldsymbol{Y} + d\boldsymbol{X}\boldsymbol{Y}) + \alpha^2 d\boldsymbol{X} \bullet d\boldsymbol{Y}\right) \\
&= \xi\left(\boldsymbol{X} \bullet \boldsymbol{Y} + \alpha \mathrm{Tr}\,(\boldsymbol{X} d\hat{\boldsymbol{Y}} + d\hat{\boldsymbol{X}}\boldsymbol{Y}) + \alpha^2 d\boldsymbol{X} \bullet d\boldsymbol{Y}\right) \\
&\quad \text{(since } \boldsymbol{X} \text{ and } \boldsymbol{Y} \text{ are symmetric)} \\
&= \xi\left(\boldsymbol{X} \bullet \boldsymbol{Y} + \alpha \mathrm{Tr}\,(\beta\mu\boldsymbol{I} - \boldsymbol{X}\boldsymbol{Y}) + \alpha^2 \mathrm{Tr}\, \sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X} \sqrt{\boldsymbol{Y}} \sqrt{\boldsymbol{Y}}^{-1} d\boldsymbol{Y} \sqrt{\boldsymbol{X}}\right) \\
&\quad \text{(by the Newton equation (23))} \\
&\ge \xi\left((1 - (1-\beta)\alpha)\boldsymbol{X} \bullet \boldsymbol{Y} - \alpha^2 \|\sqrt{\boldsymbol{X}}^{-1} d\boldsymbol{X} \sqrt{\boldsymbol{Y}}\|_F \cdot \|\sqrt{\boldsymbol{Y}}^{-1} d\boldsymbol{Y} \sqrt{\boldsymbol{X}}\|_F\right) \\
&\quad \text{(since } n\mu = \boldsymbol{X} \bullet \boldsymbol{Y}\text{)} \\
&\ge \xi\left((1 - (1-\beta)\alpha)\boldsymbol{X} \bullet \boldsymbol{Y} - \alpha^2 \|\sqrt{\boldsymbol{X}}^{-1} d\hat{\boldsymbol{X}} \sqrt{\boldsymbol{Y}}\|_F \cdot \|\sqrt{\boldsymbol{Y}}^{-1} d\hat{\boldsymbol{Y}} \sqrt{\boldsymbol{X}}\|_F\right) \\
&\ge \xi\left((1 - (1-\beta)\alpha)\boldsymbol{X} \bullet \boldsymbol{Y} - \alpha^2 \left(\frac{\zeta n\mu}{\sqrt{\lambda_{min}}}\right)^2\right) \quad \text{(by 2 of Lemma 7.6)} \\
&= (1-\alpha)\xi \boldsymbol{X} \bullet \boldsymbol{Y} + \alpha\xi\left(\beta n\mu - \alpha \cdot \frac{\zeta^2 n^2 \mu^2}{\lambda_{min}}\right) \\
&= (1-\alpha)\xi \boldsymbol{X} \bullet \boldsymbol{Y} + \alpha\xi\left(\frac{n^2\mu}{n+\nu} - \frac{\lambda_{min}}{5\zeta^2 n(n+\nu)\mu} \cdot \frac{\zeta^2 n^2 \mu^2}{\lambda_{min}}\right) \\
&\quad \left(\text{since } \beta = \frac{n}{n+\nu} \text{ and } \alpha = \frac{\lambda_{min}}{5\zeta^2 n(n+\nu)\mu}\right) \\
&= (1-\alpha)\xi \boldsymbol{X} \bullet \boldsymbol{Y} + \alpha\xi\left(\frac{n^2\mu}{n+\nu} - \frac{n\mu}{5(n+\nu)}\right) \\
&\ge (1-\alpha)\xi \boldsymbol{X} \bullet \boldsymbol{Y} \\
&\ge (1-\alpha)\theta^r \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0 \quad \text{(by assertion 1 of the theorem).}
\end{aligned}$$

Thus we have shown the inequality (65).

Now we derive the inequality (66):

$$\begin{aligned}
&f(\boldsymbol{X} + \alpha d\boldsymbol{X}, \boldsymbol{Y} + \alpha d\boldsymbol{Y}) - f(\boldsymbol{X}, \boldsymbol{Y}) \\
&\le \alpha G_1(d\boldsymbol{X}, d\boldsymbol{Y}) + \alpha^2 G_2(d\boldsymbol{X}, d\boldsymbol{Y}) \text{ (by 1 of Lemma 7.4)} \\
&= -\frac{\alpha}{\beta\mu}\|\boldsymbol{H}(\beta)\|_F^2 + \frac{\alpha^2(n+\nu) d\boldsymbol{X} \bullet d\boldsymbol{Y}}{n\mu} + \frac{\alpha^2 \sum_{j=1}^n (\xi_j^2 + \eta_j^2)}{2(1-\tau)} \\
&\quad \text{(by 1 of Lemma 7.4 with } \tau = 1/20\text{)} \\
&\le -\frac{\alpha}{\beta\mu}\|\boldsymbol{H}(\beta)\|_F^2 + \frac{\alpha^2(n+\nu) d\hat{\boldsymbol{X}} \bullet d\hat{\boldsymbol{Y}}}{n\mu} + \frac{\alpha^2 \sum_{j=1}^n (\xi_j^2 + \eta_j^2)}{2(1-\tau)} \\
&\le -\frac{\alpha}{\beta\mu}\|\boldsymbol{H}(\beta)\|_F^2 + \frac{\alpha^2}{4\beta\mu}\|\boldsymbol{H}(\beta)\|_F^2 + \frac{2\alpha^2(\zeta n\mu)^2)}{2(1-1/20)\lambda_{min}^2} \\
&\quad \text{(by 3 of Lemma 7.1 with } \boldsymbol{Q} = \beta\mu\boldsymbol{I} - \boldsymbol{X}\boldsymbol{Y}\text{, 3 of Lemma 7.6,} \\
&\qquad \text{and } \beta = n/(n+\nu)\text{)}
\end{aligned}$$

$$
\begin{aligned}
&\leq -\frac{\alpha}{\beta\mu}\|\boldsymbol{H}(\beta)\|_F^2 \cdot \left(1-\frac{\alpha}{4}\right) + \frac{\alpha^2(\zeta n\mu)^2)}{(1-1/20)\lambda_{min}^2} \\
&\leq -\frac{\lambda_{min}}{5\zeta^2\beta(n+\nu)^2\mu} \cdot \frac{1}{\beta\mu}\|\boldsymbol{H}(\beta)\|_F^2 \cdot \left(1-\frac{1}{320}\right) \\
&\quad + \left(\frac{\lambda_{min}}{5\zeta^2 n(n+\nu)\mu}\right)^2 \cdot \frac{20(\zeta n\mu)^2}{19\lambda_{min}^2} \ \left(\text{since } \alpha = \frac{\lambda_{min}}{5\zeta^2 n(n+\nu)\mu} \leq \frac{1}{80}\right) \\
&= -\frac{1}{5\zeta^2(n+\nu)^2} \cdot \left(\frac{\sqrt{\lambda_{min}}}{\beta\mu}\|\boldsymbol{H}(\beta)\|_F\right)^2 \cdot \frac{319}{320} \\
&\quad + \frac{1}{5^2\zeta^2(n+\nu)^2} \cdot \frac{20}{19} \\
&\leq -\frac{1}{5\zeta^2(n+\nu)^2} \cdot \frac{3}{4} \cdot \frac{319}{320} + \frac{1}{25\zeta^2(n+\nu)^2} \cdot \frac{20}{19} \ \text{(by 2 of Lemma 7.4)} \\
&\leq -\frac{1}{10\zeta^2(n+\nu)^2}.
\end{aligned}
$$

Thus we have shown the inequality (66).

Assertion 2 follows directly from 1 of Lemma 7.6. This completes the proof of Theorem 8.3.

Assume that the hypothesis in section 7 is true. Then Theorem 8.3 ensures that the IIP potential-reduction method generates an infinite sequence $\{(\boldsymbol{X}^r, \boldsymbol{Y}^r) \in \mathcal{S}^2_{++}\}$ satisfying

$$
\left\{
\begin{array}{l}
f(\boldsymbol{X}^r, \boldsymbol{Y}^r) \leq f(\boldsymbol{X}^0, \boldsymbol{Y}^0) - r\delta, \\
\theta^r \boldsymbol{X}^0 \bullet \boldsymbol{Y}^0 \leq \xi \boldsymbol{X}^r \bullet \boldsymbol{Y}^r, \\
(\boldsymbol{X}^r, \boldsymbol{Y}^r) \in \mathcal{F} + \theta^r \left((\boldsymbol{X}^0, \boldsymbol{Y}^0) - (\boldsymbol{X}', \boldsymbol{Y}')\right)
\end{array}
\right. \tag{71}
$$

for every $r = 0, 1, 2, \ldots$. Thus we may regard $(\boldsymbol{X}^r, \boldsymbol{Y}^r)$ with any sufficiently large r as an approximate solution of the SDLCP (1). More precisely, for any given $\epsilon > 0$ we have $\boldsymbol{X}^r \bullet \boldsymbol{Y}^r \leq \epsilon$ and $\theta^r \leq \xi\epsilon/(\boldsymbol{X}^0 \bullet \boldsymbol{Y}^0)$ if $r \geq (f(\boldsymbol{X}^0, \boldsymbol{Y}^0) - \nu \log \epsilon)/\delta$. If in addition $\nu = \sqrt{n}$ and $(\boldsymbol{X}^0, \boldsymbol{Y}^0) = \rho(\boldsymbol{I}, \boldsymbol{I})$ for some $\rho > 0$, then the right-hand side of the inequality above is of $O(n^{2.5} \log(n\rho/\epsilon))$, the same order as the constrained potential-reduction algorithm (Algorithm I) proposed by Mizuno, Kojima, and Todd [32]. Besides the constrained potential-reduction algorithm, Mizuno, Kojima, and Todd [32] also presented a pure potential-reduction algorithm (Algorithm II) and its $O(nL)$-iteration variant (Algorithm III). We could also modify the IIP potential-reduction method to develop such variants, but the details are omitted here.

9. Concluding remarks. There remain many theoretical and practical issues to be studied further on the monotone SDLCP (1) in symmetric matrices and interior-point methods for solving it. In particular, the authors are interested in feasible and infeasible interior-point methods using a wide neighborhood of the central trajectory. The central trajectory following method in section 8.1 is mainly of theoretical importance. We need to prepare an initial feasible interior point $(\boldsymbol{X}^0, \boldsymbol{Y}^0)$ in the narrow neighborhood $\mathcal{N}(\gamma)$ with $\gamma = 0.1$ and confine the generated sequence in the neighborhood. Even when we know such an initial feasible interior point, we should take a smaller search direction parameter β and a larger step size parameter α to increase the computational efficiency. Instead of $\mathcal{N}(\gamma)$, we could consider a wider neighborhood

$$
\mathcal{N}_\infty(\pi) = \left\{ (\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{S}^2_{++} : \begin{array}{l} \lambda_{min} \geq \pi \mathrm{Tr}\ \boldsymbol{X}\boldsymbol{Y}/n, \text{ where } \lambda_{min} \text{ denotes} \\ \text{the minimum eigenvalue of } \boldsymbol{X}\boldsymbol{Y} \end{array} \right\}
$$

(for infeasible interior-point methods)

or

$\mathcal{N}_\infty(\pi) \cap \mathcal{F}_{++}$ (for feasible interior-point methods)

of the central trajectory. Here $\pi > 0$. This type of neighborhood has been successfully utilized in many feasible and infeasible interior-point methods ([18, 20, 26, 33, 48], etc.) for linear programs in the Euclidean space. The authors tried to extend the Kojima–Megiddo–Mizuno infeasible interior-point method [18] to the SDLCP (1) in symmetric matrices but encountered some difficulties in analyzing the step length parameter α, which keeps the next iterate remaining in the neighborhood $\mathcal{N}_\infty(\pi)$.

Nesterov and Nemirovskii [36] discussed variational inequalities with monotone operators and presented a path-following method for solving them.

In their recent paper [37], Nesterov and Todd presented a quite general theoretical foundation for interior-point algorithms for a wide class of nonlinear programs in conic form including a primal-dual pair (2) of semidefinite programs as a special case. Among others, they proposed a joint scaling primal-dual interior-point method for linear programs in conic form, which is an extension of the $O(\sqrt{n}L)$ iteration potential-reduction algorithm given by Kojima–Mizuno–Yoshise [22]. Our current paper has been written independently of their paper [37].

Acknowledgment. The authors wish to thank Dr. M. Shida and the anonymous referees for very thorough reading of the manuscript and many valuable comments and suggestions to improve the paper.

REFERENCES

[1] F. Alizadeh, *Optimization over the positive-definite cone: Interior point methods and combinatorial applications*, in Advances in Optimization and Parallel Computing, P. Pardalos, ed., North-Holland, Amsterdam, 1992, pp. 1–25.

[2] F. Alizadeh, *Interior point methods in semidefinite programming with application to combinatorial optimization*, SIAM J. Optim., 5 (1995), pp. 13–51.

[3] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, *Primal-dual interior-point methods for semidefinite programming*, Presented at the Mathematical Programming Symposium, Ann Arbor, MI, 1994, manuscript.

[4] J. F. Bonnans and C. C. Gonzaga, *Convergence of interior point algorithms for the monotone linear complementarity problem*, Technical Report 2074, INRIA, Rocquencourt, France, 1993.

[5] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia, PA, 1994.

[6] R. W. Cottle, J. S. Pang, and R. E. Stone, *The Linear Complementarity Problem*, Academic Press, New York, 1992.

[7] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley & Sons, New York, 1968.

[8] G. H. Golub and C. H. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, London, 1989.

[9] C. C. Gonzaga and M. J. Todd, *An $O(\sqrt{n}L)$-iteration large-step primal-dual affine algorithm for linear programming problems*, SIAM J. Optim., 2 (1992), pp. 349–359.

[10] M. S. Gowda and T. I. Seidman, *Generalized linear complementarity problems*, Math. Programming, 46 (1990), pp. 329–340.

[11] O. Güller, *Generalized linear complementarity problems and interior point algorithms for their solutions*, Internal Report, Dept. of Industrial Engineering and Operations Research, University of California, Berkeley, CA, 1993.

[12] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz, *An interior-point method for semidefinite programming*, Program in Statistics and Operations Research, Princeton University, Princeton, NJ, 1994.

[13] R. D. Hill and S. R. Waters, *On the cone of positive semidefinite matrices*, Linear Algebra Appl., 90 (1987), pp. 81–88.

[14] M. W. Hirsch and S. Smale, *Differential Equations, Dynamical Systems, and Linear Algebra*, Academic Press, New York, 1974.

[15] F. Jarre, *An interior-point method for minimizing the maximum eigenvalue of a linear combination of matrices*, SIAM J. Control Optim., 31 (1993), pp. 1360–1377.

[16] N. Karmarkar, *A New Polynomial-Time Algorithm for Linear Programming*, Combinatorica, 4 (1984), pp. 373–395.

[17] M. Kojima, *Basic lemmas in polynomial-time infeasible-interior-point methods for linear programs*, Ann. Oper. Res., 62 (1996), pp. 1–28.

[18] M. Kojima, N. Megiddo, and S. Mizuno, *A primal-dual infeasible-interior-point algorithm for linear programming*, Math. Programming, 61 (1993), pp. 263–280.

[19] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise, *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems*, Lecture Notes in Computer Science 538, Springer-Verlag, New York, 1991.

[20] 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.

[21] M. Kojima, S. Mizuno, and A. Yoshise, *A polynomial-time algorithm for a class of linear complementary problems*, Math. Programming, 44 (1989), pp. 1–26.

[22] M. Kojima, S. Mizuno, and A. Yoshise, *An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems*, Math. Programming, 50 (1991), pp. 331–342.

[23] M. Kojima, T. Noma, and A. Yoshise, *Global convergence in infeasible-interior-point algorithms*, Math. Programming, 65 (1994), pp. 43–72.

[24] M. Kojima, M. Shida, and S. Shindoh, *Reduction of monotone linear complementarity problems over cones to linear programs over cones*, Research Report #296, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro, Tokyo 152, Japan, 1995.

[25] I. J. Lustig, *Feasibility issues in a primal-dual interior-point method for linear programming*, Math. Programming, 49 (1990/91), pp. 145–162.

[26] I. J. Lustig, R. E. 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.

[27] R. Marsten, R. Subramanian, M. Saltzman, I. J. Lustig, and D. Shanno, *Interior point methods for linear programming: Just call Newton, Lagrange, and Fiacco and McCormick!*, Interfaces, 20 (1990), pp. 105–116.

[28] L. McLinden, *The complementarity problem for maximal monotone multifunctions*, in Variational Inequalities and Complementarity Problems, R. W. Cottle, F. Giannessi, and J.-L. Lions, eds., John Wiley & Sons, New York, 1980, pp. 251–270.

[29] N. Megiddo, *A monotone complementarity problem with feasible solutions but not complementarity solutions*, Math. Programming, 12 (1977), pp. 131–132.

[30] 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.

[31] S. Mizuno, F. Jarre, and J. Stoer, *A unified approach to infeasible-interior-point algorithms via geometrical linear complementarity problems*, Inst. Statist. Math., 1994.

[32] S. Mizuno, M. Kojima, and M. J. Todd, *Infeasible-interior-point primal-dual potential-reduction algorithms for linear programming*, SIAM J. Optim., 5 (1995), pp. 52–67.

[33] 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.

[34] R. D. C. Monteiro and I. Adler, *Interior path following primal-dual algorithms. Part* I: *Linear programming*, Math. Programming, 44 (1989), pp. 27–41.

[35] Ju. E. Nesterov and A. S. Nemirovskii, *Self-concordant functions and polynomial-time methods in convex programming*, Report, Central Economical and Mathematical Institute, USSR Acad. Sci., Moscow, USSR, 1989.

[36] Ju. E. Nesterov and A. S. Nemirovskii, *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*, SIAM, Philadelphia, PA, 1994.

[37] Ju. E. Nesterov and M. J. Todd, *Self-Scaled Cones and Interior-Point Methods in Nonlinear Programming*, Working Paper, CORE, Catholic University of Louvain, Louvain-la-Neuve, Belgium, 1994.

[38] A. M. Ostrowski and O. Taussky, *On the variation of the determinant of a positive definite matrix*, Nederl. Akad. Wetensch, Proc. Ser. A, 54 (1951), pp. 383–385.

[39] R. Sznajder and M. S. Gowda, *Generalizations of P_0- and P- properties; extended vertical and horizontal LCP's*, Linear Algebra Appl., to appear.

[40] K. Tanabe, *Centered Newton method for mathematical programming*, in System Modeling and Optimization, M. Iri and K. Yajima, eds., Springer-Verlag, New York, 1988, pp. 197–206.

[41] M. J. Todd, *Recent developments and new directions in linear programming*, in Recent Developments and Applications, M. Iri and K. Tanabe, eds., Kluwer Academic Publishers, London, 1989, pp. 109–157.

[42] M. J. Todd and Y. Ye, *A centered projective algorithm for linear programming*, Math. Oper. Res., 15 (1990), pp. 508–529.

[43] R. H. Tütüncü and M. J. Todd, *Reducing horizontal linear complementarity problems*, Technical Report, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1993.

[44] L. Vandenberghe and S. Boyd, *A primal-dual potential reduction method for problems involving matrix inequalities*, Math. Programming, Series B, 69 (1995), pp. 205–236.

[45] L. Vandenberghe and S. Boyd, *Semidefinite programming*, Working Paper, Information Systems Laboratory, Stanford University, Stanford, CA, 1995.

[46] Y. Ye, *An $O(n^3L)$ potential reduction algorithm for linear programming*, Math. Programming, 50 (1991), pp. 239–258.

[47] Y. Ye, *A fully polynomial-time approximation algorithm for computing a stationary point of the general linear complementarity problem*, Math. Oper. Res., 18 (1993), pp. 334–345.

[48] Y. Zhang, *On the convergence of a class of infeasible interior-point algorithms for the horizontal linear complementarity problem*, SIAM J. Optim., 4 (1994), pp. 208–227.

[49] Y. Zhang, R. A. Tapia, and F. Potra, *On the superlinear convergence of interior point algorithms for a general class of problems*, SIAM J. Optim., 3 (1993), pp. 413–422.

SIAM J. OPTIM.
Vol. 7, No. 1, pp. 126–140, February 1997

007

A FAMILY OF POLYNOMIAL AFFINE SCALING ALGORITHMS FOR POSITIVE SEMIDEFINITE LINEAR COMPLEMENTARITY PROBLEMS[*]

B. JANSEN[†], C. ROOS[†], AND T. TERLAKY[†]

Abstract. In this paper the new polynomial affine scaling algorithm of Jansen, Roos, and Terlaky for linear programming (LP) is extended to positive semidefinite (PSD) linear complementarity problems. The algorithm is immediately further generalized to allow higher order scaling. These algorithms are also new for the LP case. The analysis is based on Ling's proof for the LP case; hence, it allows an arbitrary interior feasible pair to start with. With the scaling of Jansen, Roos, and Terlaky, the complexity of the algorithm is

$$\mathcal{O}\left(\frac{n}{\rho^2(1-\rho^2)}\ln\frac{(x^{(0)})^T s^{(0)}}{\epsilon}\right),$$

where ρ^2 is a uniform bound for the ratio of the smallest and largest coordinate of the iterates in the primal-dual space.

We also show that Monteiro, Adler, and Resende's polynomial complexity result for the classical primal-dual affine scaling algorithm can easily be derived from our analysis. In addition, our result is valid for arbitrary, not necessarily centered, initial points.

Finally, some computational results are presented, which indicate the influence of the order of the scaling on the numerical performance.

Key words. interior-point method, affine scaling method, linear complementarity problem

AMS subject classifications. 90C33, 90C20, 90C05

PII. S1052623493262348

1. Introduction.

In this paper we consider the positive semidefinite (PSD) linear complementarity problem (LCP) as follows:

$$-Mx + s = q, \quad x \geq 0, \quad s \geq 0, \quad xs = 0, \tag{1.1}$$

where M is a given $n \times n$ real PSD matrix, $q \in \mathbb{R}^n$, and xs denotes the componentwise product of the vectors x and s. The set of feasible and positive feasible vectors is denoted, respectively, by

$$\mathcal{F} = \{\,(x,s) \mid -Mx + s = q,\, x \geq 0,\, s \geq 0\,\},$$
$$\mathcal{F}^0 = \{\,(x,s) \mid -Mx + s = q,\, x > 0,\, s > 0\,\}.$$

We shall assume throughout that $\mathcal{F}^0 \neq \emptyset$.

Scaling is one of the most important techniques in modern polynomial time optimization methods. The first affine scaling algorithm of Dikin [2] remained unnoticed for a long time. After Karmarkar [9] initiated the dynamically developing field of interior point methods (IPMs), affine scaling became one of the basic concepts in IPMs. Primal or dual affine scaling methods were studied by, e.g., Barnes [1], Vanderbei, Meketon, and Freedman [17], Tsuchiya and Muramatsu [16], and Saigal [15].

[*]Received by the editors December 14, 1993; accepted for publication (in revised form) September 18, 1995. This work was completed with the support of a research grant from Shell.

http://www.siam.org/journals/siopt/7-1/26234.html

[†]Faculty of Technical Mathematics and Informatics, Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands (b.jansen@twi.tudelft.nl, c.roos@twi.tudelft.nl, t.terlaky@twi.tudelft.nl). The research of the first author was supported by the Dutch Organization for Scientific Research (NWO) under grant 611-304-028.

A primal-dual affine scaling algorithm for linear programming (LP) was analyzed by Monteiro, Adler, and Resende [14]. For a general framework of IPMs for LCP, see [10].

Recently, the authors proposed a new primal-dual affine scaling method for LP [8]. Given a nearly centered primal-dual interior feasible pair, they define an affine scaling direction (d_x, d_s) as the steepest descent in the norm induced by Dikin's ellipsoid in the primal-dual space. Provided that the step along this direction is small enough, the next iterate is still nearly centered. This is due to the fact that this new affine scaling direction has a centering component; more precisely, it is tangent to a curve that tends to the central path. Having a well-centered initial pair, the complexity of the algorithm is proved to be $\mathcal{O}(nL)$ iterations (compared with the $\mathcal{O}(nL^2)$ complexity bound of the classical primal-dual affine scaling algorithm [14]). Ling [11] gave a new analysis of the new affine scaling method, allowing any interior starting point and proving that the complexity of the algorithm is $\mathcal{O}(\frac{n}{\rho^2}L)$, where ρ^2 is uniform bound for the ratio of the smallest and largest coordinate of the product $x^{(k)}s^{(k)}$ during the algorithm.

The aim of this paper is to generalize the approach in [8] in two ways. First, we consider the LCP instead of the LP problem. Secondly, we analyze a family of affine scaling methods of which the algorithm in [8] is just a special case. It is also shown that the classical affine scaling algorithm can be derived as the limiting case of our family, and the use of our analysis provides a new, simple proof for the polynomial complexity (see also Monteiro, Adler, and Resende [14] and Mizuno and Nagasawa [13]) of the classical affine scaling algorithm, using an arbitrary initial interior point. Hence, as in [8], we apply and further extend Dikin's original scaling approach in the primal-dual (xs)-space. It may be recalled from [8] that the resulting search directions are not a linear combination of the classical affine scaling and centering directions, as is usual in the context of IPMs (see [7]).

The paper is organized as follows. The generalized Dikin-type search directions are derived and discussed in section 2. The algorithmic frame is presented in section 3. Some general results, true for all $r \geq 0$, are proved in section 4. The proof of polynomial convergence for $r > 0$ is presented in section 5. Then, in section 6 we derive the polynomial complexity of the classical primal-dual affine scaling algorithm with suitable step size. Finally, section 7 contains some illustrative computational results.

Throughout, we shall use $\|\cdot\|_p$ ($p \in [1, \infty]$) to denote the l_p norm on $\mathbb{R}^n$, with $\|\cdot\|$ denoting the Euclidean norm $\|\cdot\|_2$. E will denote the identity matrix; e will be used to denote the vector which has all its components equal to one. Given an n-dimensional vector d, we denote by D the $n \times n$ diagonal matrix whose diagonal entries are the coordinates d_j of d. If x, $s \in \mathbb{R}^n$ then $x^T s$ denotes the dot product of the two vectors. Further, xs and x^α for $\alpha \in \mathbb{R}$ will denote the vector resulting from componentwise operations.

2. The search directions. In [8], the new affine scaling direction $(\Delta x, \Delta s)$ for LP is obtained by minimizing the duality gap over a suitable ellipsoid which is called the primal-dual Dikin ellipsoid. In this paper we generalize this approach to LCPs, and we also generalize the scaling by introducing a parameter $r > 0$ which is called the degree of scaling. We remark that $r = 1$ will give the algorithm studied in [8] for LP and the classical affine scaling algorithm is obtained with the value $r = 0$.

Let a strictly feasible pair $(x, s) \in \mathcal{F}^0$ be given. To determine the search direction

$(\Delta x, \Delta s)$ for a fixed value of r we consider the following problem:

$$\begin{array}{lrl} & \text{minimize } ((xs)^r)^T(x^{-1}\Delta x + s^{-1}\Delta s) & \\ \text{(2.1)} & \text{subject to } \quad -M\Delta x + \Delta s & = 0, \\ & \|x^{-1}\Delta x + s^{-1}\Delta s\| & \leq 1. \end{array}$$

This minimization problem has a unique solution, as we now will show. It is convenient to introduce some notations. For each positive primal-dual pair (x, s), define

$$v := (xs)^{\frac{1}{2}} \quad \text{and} \quad d := (xs^{-1})^{\frac{1}{2}}.$$

Hence we have

$$x = dv \quad \text{and} \quad s = d^{-1}v.$$

Further, let us denote

$$\begin{array}{lrclcrcl} \text{(2.2)} & p_x & := & d^{-1}\Delta x & \text{and} & p_s & := & d\Delta s, \\ & p_v & := & p_x + p_s & \text{and} & \overline{M} & := & DMD. \end{array}$$

These relations imply that

$$x\Delta s + s\Delta x = xd^{-1}p_s + sdp_x = v(p_x + p_s) = vp_v,$$

hence

$$x^T\Delta s + s^T\Delta x = v^T p_v,$$

and

$$x^{-1}\Delta x + s^{-1}\Delta s = (xs)^{-1}(x\Delta s + s\Delta x) = v^{-2}vp_v = v^{-1}p_v.$$

Using these notations problem (2.1) can be reformulated as follows:

$$\begin{array}{lrl} & \text{minimize } (v^{2r-1})^T(p_x + p_s) & \\ \text{(2.3)} & \text{subject to } \quad -\overline{M}p_x + p_s & = 0, \\ & \|v^{-1}(p_x + p_s)\| & \leq 1. \end{array}$$

We can eliminate p_s from (2.3) by using $p_s = \overline{M}p_x$. Now, using (2.2),

$$p_v = p_x + p_s = (E + \overline{M})p_x.$$

Hence (2.3) is equivalent to

$$\begin{array}{rl} \text{minimize } (v^{2r-1})^T p_v & \\ \text{subject to } \quad \|v^{-1}p_v\| & \leq 1. \end{array}$$

This is a trivial optimization problem with a unique solution; namely,

$$p_v = -\frac{v^{2r+1}}{\|v^{2r}\|}. \tag{2.4}$$

From now on, p_v will have the meaning shown in (2.4). Thus we find

$$p_x = (E+\overline{M})^{-1}p_v \quad \text{and} \quad p_s = \overline{M}(E+\overline{M})^{-1}p_v. \tag{2.5}$$

Rescaling to Δx and Δs we thus find

$$\Delta x = D(E+DMD)^{-1}p_v; \qquad \Delta s = MD(E+DMD)^{-1}p_v. \tag{2.6}$$

We can now also calculate the optimal value of (2.1); namely,

$$((xs)^r)^T(x^{-1}\Delta x + s^{-1}\Delta s) = (v^{2r-1})^T p_v = -\frac{e^T v^{4r}}{\|v^{2r}\|} = -\|v^{2r}\|.$$

Note, that $(\Delta x, \Delta s)$ is the unique solution of the system of equations

$$\begin{aligned} -M\Delta x + \Delta s &= 0, \\ s\Delta x + x\Delta s &= -\tfrac{v^{2r+2}}{\|v^{2r}\|}. \end{aligned}$$

If we compare this with the classical equation system of the primal-dual affine scaling method, we see that at the right-hand side of the second equation we have $-\frac{v^{2r+2}}{\|v^{2r}\|}$ instead of $-v^2$. See, e.g., [10]. So classical affine scaling occurs for $r=0$, whereas $r=1$ gives the new affine scaling direction proposed in [8].

3. The algorithm. The algorithm is initialized with $(x^{(0)}, s^{(0)}) \in \mathcal{F}^0$ and repeatedly makes steps in the direction $(\Delta x, \Delta s)$, using a fixed step size θ, until the error in complementarity reaches some prescribed value ϵ. For each degree of scaling $r > 0$ one can define an algorithm, which formally is stated as follows.

Algorithm.

```
Input
    $(x^0, s^0)$: the initial pair of interior feasible solutions;
    $r > 0$: the degree of scaling;

Parameters
    $\varepsilon$ is the accuracy parameter;
    $\theta$ is the step size;
begin
    $x := x^{(0)}$; $s := s^{(0)}$;
    while  $x^T s > \varepsilon$  do
        calculate $\Delta x$ and $\Delta s$ from (2.4) and (2.6);
        $x := x + \theta\Delta x$;
        $s := s + \theta\Delta s$;
    end
end.
```

4. General results for $r \geq 0$. Given (x, s), the new iterates will be denoted by $\hat{x} = x + \theta\Delta x$ and $\hat{s} = s + \theta\Delta s$, respectively, where θ is the step size. So we have

$$\hat{v}^2 := \hat{x}\hat{s} = xs + \theta(x\Delta s + s\Delta x) + \theta^2\Delta x\Delta s = v^2 + \theta v p_v + \theta^2 p_x p_s.$$

As a consequence, the new error in complementarity is given by

$$\hat{x}^T\hat{s} = e^T\hat{v}^2 = e^T v^2 + \theta v^T p_v + \theta^2 p_x^T p_s = e^T v^2 - \theta\frac{e^T v^{2r+2}}{\|v^{2r}\|} + \theta^2 p_x^T p_s. \tag{4.1}$$

To be able to bound the error in complementarity we need to bound the last two terms in (4.1). Note that due to matrix M being PSD, one has that $\overline{M}$ is also PSD; hence,

$$p_x^T p_s = p_x^T \overline{M} p_x \geq 0.$$

From (4.1) it is clear that for an estimate of the error in complementarity after one iteration, we need an estimate for $p_x^T p_s$. Later on we also need an estimate for $\|p_x p_s\|_\infty$. For both purposes the following lemma is useful.

LEMMA 4.1. *Let p_x, p_s, and p_v be defined as in* (2.4) *and* (2.5). *One has*

(i) $\|p_v\| \leq \|v\|_\infty \leq \|v\|$,

(ii) $0 \leq \Delta x^T \Delta s = p_x^T p_s \leq \frac{\|p_v\|^2}{4}$,

(iii) $\|\Delta x \Delta s\|_\infty = \|p_x p_s\|_\infty \leq \frac{\|p_v\|^2}{4}$.

Proof. (i): since $p_v = -\frac{v^{2r+1}}{\|v^{2r}\|}$, the inequalities $\|p_v\| \leq \|v\|_\infty \leq \|v\|$ are obvious.

(ii): to prove the other inequalities, we introduce the notation $q_v = p_x - p_s$. One has

$$\|q_v\|^2 = \|p_x\|^2 + \|p_s\|^2 - 2p_x^T p_s = \|p_v\|^2 - 4p_x^T p_s \leq \|p_v\|^2. \tag{4.2}$$

Consequently,

$$\Delta x^T \Delta s = p_x^T p_s = \tfrac{1}{4}(\|p_v\|^2 - \|q_v\|^2) \leq \tfrac{1}{4}\|p_v\|^2,$$

which proves (ii).

(iii): using that

$$p_x p_s = \tfrac{1}{4}(p_v^2 - q_v^2),$$

we write

$$\|p_x p_s\|_\infty \leq \tfrac{1}{4}\max(\|p_v\|_\infty^2, \|q_v\|_\infty^2) \leq \tfrac{1}{4}\max(\|p_v\|^2, \|q_v\|^2) = \tfrac{1}{4}\|p_v\|^2.$$

The last inequality follows from (4.2). This completes the proof of the lemma. □

We introduce some further notations. Since our algorithm can start in any interior feasible point, the complexity will depend on the ratio between the smallest and largest coordinate of v (cf. [11]). This ratio will be denoted by $\omega(v)$. So we define

$$\omega(v) := \frac{\min(v)}{\max(v)},$$

where $\max(v)$ denotes the largest coordinate of v and $\min(v)$ denotes the smallest coordinate of v. If $\omega(v) \geq \rho$, there are α, $\beta \in (0, \infty)$ such that

$$\alpha e \leq v^2 \leq \beta e, \quad \text{with} \quad \frac{\alpha}{\beta} = \rho^2. \tag{4.3}$$

A crucial part of the analysis is to give an upper bound for the second term in (4.1):

$$\vartheta(r) := v^T p_v = -\frac{e^T v^{2r+2}}{\|v^{2r}\|},$$

where v is given. Later on we also will give conditions which ensure that $\omega(v)$ will remain bounded during the algorithm.

LEMMA 4.2. *Let $v \in \mathbb{R}^n_+$ be an arbitrary vector. Depending on the value of r, the following bounds hold for $\vartheta(r)$.*

(i) *If* $0 \le r \le 1$, *then* $\vartheta(r) \le -\frac{\|v\|^2}{\sqrt{n}}$.

(ii) *If* $1 \le r$ *and* $\omega(v) \ge \rho$, *then* $\vartheta(r) \le -\frac{\rho^{2r-2}}{\sqrt{n}}\|v\|^2$.

Proof. (i): it is obvious that $\vartheta(0) = -\frac{\|v\|^2}{\sqrt{n}}$. Hence it is enough to show that the derivative of $\vartheta(r)$ is nonpositive as long as $0 \le r \le 1$. Let us first differentiate the nominator and denominator separately.

$$(e^T v^{2r+2})' = 2\sum_{i=1}^n v_i^{2r+2} \ln v_i.$$

$$\|v^{2r}\|' = \left(\sqrt{\sum_{i=1}^n v_i^{4r}}\right)' = \frac{4}{2\|v^{2r}\|}\sum_{i=1}^n v_i^{4r} \ln v_i = \frac{2\sum_{i=1}^n v_i^{4r} \ln v_i}{\|v^{2r}\|}.$$

The sign of $\vartheta'(r)$ is determined by the nominator of the derivative, which is given by

$$\begin{aligned} &-\left(2\sum_{i=1}^n v_i^{2r+2}\|v^{2r}\| \ln v_i - \frac{2}{\|v^{2r}\|}\sum_{i=1}^n v_i^{4r}\|v^{r+1}\|^2 \ln v_i\right) \\ &= \frac{2}{\|v^{2r}\|}\sum_{i=1}^n \left(v_i^{4r}\|v^{r+1}\|^2 - v_i^{2r+2}\|v^{2r}\|^2\right)\ln v_i. \end{aligned}$$

Now we may write

$$\begin{aligned} 2\sum_{i=1}^n \left(v_i^{4r}\|v^{r+1}\|^2 - v_i^{2r+2}\|v^{2r}\|^2\right)\ln v_i &= 2\sum_{i=1}^n\sum_{j=1}^n \left(v_i^{4r}v_j^{2r+2} - v_i^{2r+2}v_j^{4r}\right)\ln v_i \\ &= 2\sum_{i,j=1}^n (v_iv_j)^{2r+2}(v_i^{2r-2} - v_j^{2r-2})\ln v_i \\ &= \sum_{i,j=1}^n (v_iv_j)^{2r+2}\left((v_i^{2r-2} - v_j^{2r-2})\ln v_i + (v_j^{2r-2} - v_i^{2r-2})\ln v_j\right) \\ &= \sum_{i,j=1}^n (v_iv_j)^{2r+2}(v_i^{2r-2} - v_j^{2r-2})\ln\frac{v_i}{v_j}. \end{aligned}$$

The last expression is nonpositive for $r \le 1$ and nonnegative for $r > 1$, hence $\vartheta(r)$ is monotone nonincreasing if $0 \le r \le 1$ and monotone nondecreasing if $r > 1$. Since $\vartheta(0) = -\frac{\|v\|^2}{\sqrt{n}}$ we have $\vartheta(r) \le -\frac{\|v\|^2}{\sqrt{n}}$ if $0 \le r \le 1$. The first part of the lemma is proved.

(ii): using (4.3), one has

$$\begin{aligned} -\vartheta(r) &= \frac{e^T v^{2r+2}}{\|v^{2r}\|\|v\|^2}\|v\|^2 \\ &\ge \frac{\alpha^{r-1}e^T v^4}{\beta^{r-1}\|v^2\|\|v\|^2}\|v\|^2 \\ &\ge \frac{\rho^{2r-2}}{\sqrt{n}}\|v\|^2. \end{aligned}$$

The last inequality follows from

$$\frac{e^T v^4}{\|v^2\|\|v\|^2} = \frac{\|v^2\|}{\|v\|^2} \geq \frac{1}{\sqrt{n}},$$

where the Cauchy–Schwartz inequality is used. The proof is completed. □

Remark. Observe that the above lemma is trivial if $r = 1$. In that case, $\vartheta(1) \leq -\frac{\|v\|^2}{\sqrt{n}}$ is an immediate consequence of the Cauchy–Schwartz inequality. This was the last step in the above proof.

Now we can guarantee a decrease in the error of complementarity.

LEMMA 4.3. (i): *if* $0 \leq r \leq 1$ *and* $\theta \leq \frac{2}{\sqrt{n}}$ *then*

$$\hat{x}^T\hat{s} = \|\hat{v}\|^2 \leq \left(1 - \frac{\theta}{2\sqrt{n}}\right)\|v\|^2. \tag{4.4}$$

(ii): *if* $1 \leq r$ *and* $\theta \leq \frac{2\rho^{2r-2}}{\sqrt{n}}$ *then*

$$\hat{x}^T\hat{s} = \|\hat{v}\|^2 \leq \left(1 - \frac{\theta\rho^{2r-2}}{2\sqrt{n}}\right)\|v\|^2. \tag{4.5}$$

Proof. From (4.1) one has the following:

$$\|\hat{v}\|^2 = \|v\|^2 - \theta\frac{e^T v^{2r+2}}{\|v^{2r}\|} + \theta^2 p_x^T p_s = \|v\|^2 + \vartheta(r) + \theta^2 p_x^T p_s.$$

(i): using Lemmas 4.1 and 4.2 we obtain

$$\|\hat{v}\|^2 \leq \left(1 - \frac{\theta}{\sqrt{n}} + \frac{\theta^2}{4}\right)\|v\|^2.$$

Since $\theta \leq \frac{2}{\sqrt{n}}$, it follows that

$$1 - \frac{\theta}{\sqrt{n}} + \frac{\theta^2}{4} \leq 1 - \frac{\theta}{\sqrt{n}} + \frac{\theta}{2\sqrt{n}} = 1 - \frac{\theta}{2\sqrt{n}}.$$

This proves the first part.

(ii): we use Lemmas 4.1 and 4.2 again to get

$$\|\hat{v}\|^2 \leq \left(1 - \frac{\theta\rho^{2r-2}}{\sqrt{n}} + \frac{\theta^2}{4}\right)\|v\|^2.$$

Since $\theta \leq \frac{2\rho^{2r-2}}{\sqrt{n}}$, it follows that

$$1 - \frac{\theta\rho^{2r-2}}{\sqrt{n}} + \frac{\theta^2}{4} \leq 1 - \frac{\theta\rho^{2r-2}}{\sqrt{n}} + \frac{\theta\rho^{2r-2}}{2\sqrt{n}} = 1 - \frac{\theta\rho^{2r-2}}{2\sqrt{n}}.$$

The lemma is proved. □

Lemma 4.3 makes clear that the algorithm will converge if the step size θ can be bounded away from zero, since this will guarantee a fixed reduction of $\|v\|^2$. If the lower bound for θ is sufficiently large, then the algorithm will be polynomial.

We proceed with a condition on the step size that guarantees feasibility of the new iterates. Let us say that the step size θ is feasible, if the new iterates are positive. Then we may state the following result.

LEMMA 4.4. *Let* $0 \leq \tau$, $x(\tau) = x + \tau\Delta x$, $s(\tau) = s + \tau\Delta s$, *and* $v^2(\tau) = x(\tau)s(\tau)$. *If* $\bar{\tau}$ *is such that* $v^2(\tau) > 0$ *for all* τ *satisfying* $0 \leq \tau \leq \bar{\tau}$, *then the step size* $\bar{\tau}$ *is feasible.*

Proof. If $\bar{\tau}$ satisfies the hypothesis of the lemma, then $x(\bar{\tau})$ and $s(\bar{\tau})$ cannot vanish for any $\tau \in [0, \bar{\tau}]$. Hence, by continuity, $x(\tau)$ and $s(\tau)$ must be positive for any such τ. □

5. The proof of polynomial complexity if $r > 0$. The next theorem makes clear that with a suitable step size, the new iterates not only stay feasible but also that the ratio of the smallest and largest coordinate of v will remain bounded by ρ; i.e., $\omega(v) \geq \rho$ stays valid for all the iterates. The proof goes along the same lines as the proof of Theorem 3 in [11] for the LP case with $r = 1$.

THEOREM 5.1. *If* $(x, s) \in \mathcal{F}^0$, $0 < \rho < 1$, $r > 0$, $\omega(v) \geq \rho$, *and*

$$0 \leq \theta \leq \min\left(2\rho\left(\sqrt{1 + \frac{\rho^2}{n}} - \frac{\rho}{\sqrt{n}}\right), \frac{\rho^{2r}\sqrt{n}}{r+1}, \frac{4\rho^2(1-\rho^{2r})}{(1+\rho^2)\sqrt{n}}\right), \tag{5.1}$$

then $(\hat{x}, \hat{s}) \in \mathcal{F}^0$ *and* $\rho \leq \omega(\hat{v})$.

Proof. The hypothesis of the theorem provides three upper bounds for the step size θ. As we will see below, the first upper bound guarantees feasibility of the new iterates and the last guarantees that $\omega(\hat{v}) \geq \rho$, both under the premise that the second bound holds.

Let α and β be defined as in (4.3). We remind that by (4.1),

$$\hat{v}^2 = \hat{x}\hat{s} = v^2 - \theta\frac{v^{2r+2}}{\|v^{2r}\|} + \theta^2 p_x p_s.$$

One easily verifies that the function

$$\varphi(t) = t - \theta\frac{t^{r+1}}{\|v^{2r}\|}$$

is monotonically increasing on the interval $[0, \beta]$ if $\theta \leq \frac{\|v^{2r}\|}{(r+1)\beta^r}$. The second upper bound for θ now guarantees the monotonicity of φ, because

$$\frac{\|v^{2r}\|}{(r+1)\beta^r} \geq \frac{\alpha^r\|e\|}{(r+1)\beta^r} = \frac{\rho^{2r}\sqrt{n}}{r+1} \geq \theta.$$

Using this monotonicity property together with (4.3), one has the inequalities

$$\left(\alpha - \theta\frac{\alpha^{r+1}}{\|v^{2r}\|}\right)e \leq v^2 - \theta\frac{v^{2r+2}}{\|v^{2r}\|} \leq \left(\beta - \theta\frac{\beta^{r+1}}{\|v^{2r}\|}\right)e.$$

So the minimal and maximal coordinates of $\hat{v}^2$ are bounded by

$$\begin{array}{rcl} \min(\hat{v}^2) & \geq & \alpha - \theta\frac{\alpha^{r+1}}{\|v^{2r}\|} - \theta^2\|p_x p_s\|_\infty, \\ \max(\hat{v}^2) & \leq & \beta - \theta\frac{\beta^{r+1}}{\|v^{2r}\|} + \theta^2\|p_x p_s\|_\infty. \end{array} \tag{5.2}$$

Now, by applying Lemma 4.1 and observing that $\|p_v\|^2 \leq \beta$, we see that

$$\|p_x p_s\|_\infty \leq \tfrac{1}{4}\beta. \tag{5.3}$$

Hence from (5.2) and (5.3),

$$\begin{array}{rcl} \min(\hat{v}^2) & \geq & \alpha - \theta\frac{\alpha^{r+1}}{\|v^{2r}\|} - \frac{1}{4}\theta^2\beta, \\ \max(\hat{v}^2) & \leq & \beta - \theta\frac{\beta^{r+1}}{\|v^{2r}\|} + \frac{1}{4}\theta^2\beta. \end{array} \tag{5.4}$$

Lemma 4.4 implies that the new iterates will be feasible if $\min(\hat{v}^2) > 0$. After dividing by α, this amounts to the following condition on θ:

$$1 - \frac{\theta\alpha^r}{\|v^{2r}\|} - \frac{\theta^2}{4\rho^2} \geq 0.$$

Since

$$\frac{\alpha^r}{\|v^{2r}\|} = \frac{\|\alpha^r e\|}{\sqrt{n}\,\|v^{2r}\|} \leq \frac{1}{\sqrt{n}},$$

this certainly holds if

$$1 - \frac{\theta}{\sqrt{n}} - \frac{\theta^2}{4\rho^2} \geq 0.$$

Elementary calculations make clear that this condition is satisfied, due to the first upper bound on θ in the theorem. So the new iterates are feasible.

Now $\omega(\hat{v}) \geq \rho$ will certainly hold if

$$\beta - \theta\frac{\beta^{r+1}}{\|v^{2r}\|} + \tfrac{1}{4}\theta^2\beta \leq \frac{1}{\rho^2}\left(\alpha - \theta\frac{\alpha^{r+1}}{\|v^{2r}\|} - \tfrac{1}{4}\theta^2\beta\right).$$

On dividing this by $\beta = \frac{\alpha}{\rho^2}$, we see that this is equivalent to

$$1 - \theta\frac{\beta^r}{\|v^{2r}\|} + \tfrac{1}{4}\theta^2 \leq 1 - \theta\frac{\alpha^r}{\|v^{2r}\|} - \frac{\theta^2}{4\rho^2}.$$

By rearranging, one has

$$\theta\frac{1+\rho^2}{\rho^2} \leq \frac{4(\beta^r - \alpha^r)}{\|v^{2r}\|};$$

that is,

$$\theta \leq \frac{4\rho^2(\beta^r - \alpha^r)}{(1+\rho^2)\|v^{2r}\|}. \tag{5.5}$$

From (4.3), the definition of α and β,

$$\alpha^r\sqrt{n} \leq \|v^{2r}\| \leq \beta^r\sqrt{n}.$$

Hence

$$\frac{\beta^r - \alpha^r}{\|v^{2r}\|} \geq \frac{1-\rho^{2r}}{\sqrt{n}},$$

so (5.5) will certainly hold if

$$\theta \leq \frac{4\rho^2(1-\rho^{2r})}{(1+\rho^2)\sqrt{n}},$$

which is guaranteed by the third upper bound for θ in the theorem. Hence it is proved that the new iterate is at least as well centered as the old one. □

We have that for each $0 < \rho < 1$ there exists a $0 < \theta$ such that all the iterates of the algorithm give a feasible primal-dual solution for which $\omega(\hat{v}) \geq \rho$. Now we are ready to present the complexity of the algorithms. The proof can be done in the usual way.

THEOREM 5.2. *If $\epsilon > 0$ is given, $(x^{(0)}, s^{(0)}) \in \mathcal{F}^0$, and θ satisfies the conditions of Lemma* 4.3 *and Theorem* 5.1, *then the algorithm stops, with a solution (x^*, s^*) for which $(x^*)^T s^* \leq \epsilon$ and $\omega(v^*) \geq \rho$ holds, after at most*

(i) $2\frac{\sqrt{n}}{\theta} \ln \frac{(x^{(0)})^T s^{(0)}}{\epsilon}$ *iterations if* $0 < r \leq 1$,

(ii) $2\frac{\sqrt{n}}{\theta\rho^{2r-2}} \ln \frac{(x^{(0)})^T s^{(0)}}{\epsilon}$ *iterations if* $1 < r$.

Proof. (i): by Lemma 4.3, in each iteration the duality gap reduces by at least the factor

$$1 - \frac{\theta}{2\sqrt{n}}. \tag{5.6}$$

So, after k steps the error in complementarity will be less than ϵ if

$$\left(1 - \frac{\theta}{2\sqrt{n}}\right)^k (x^{(0)})^T s^{(0)} \leq \epsilon.$$

Taking logarithms, one obtains

$$k \ln\left(1 - \frac{\theta}{2\sqrt{n}}\right) \leq \ln \frac{\epsilon}{(x^{(0)})^T s^{(0)}},$$

which is certainly true if

$$-k\frac{\theta}{2\sqrt{n}} \leq \ln \frac{\epsilon}{(x^{(0)})^T s^{(0)}},$$

or, equivalently,

$$k \geq \frac{2\sqrt{n}}{\theta} \ln \frac{(x^{(0)})^T s^{(0)}}{\epsilon}. \tag{5.7}$$

This proves the first part of the theorem. The proof of part (ii) is analogous. □

In the following corollaries we will use the notation $\omega_0^2 = \omega(x^{(0)} s^{(0)})$.

COROLLARY 5.3. *Let us take $(x^{(0)}, s^{(0)})$ such that $\omega_0 \geq \rho = \frac{1}{\sqrt{2}}$ holds.*

(i) *If* $0 < r \leq 1$ *and* $n \geq 4$ *then we may choose* $\theta = \frac{4(1-2^{-r})}{3\sqrt{n}}$, *hence the complexity of our algorithm is* $\mathcal{O}(\frac{n}{1-2^{-r}} \ln \frac{(x^{(0)})^T s^{(0)}}{\epsilon})$.

(ii) *If* $r = 1$ *and* $n \geq 4$ *then we may choose* $\theta = \frac{1}{2\sqrt{n}}$, *hence the complexity of our algorithm is* $\mathcal{O}(n \ln \frac{(x^{(0)})^T s^{(0)}}{\epsilon})$.

(iii) *If* $1 < r$ *and* n *is sufficiently large then we may choose* $\theta = \frac{4}{2^r\sqrt{n}}$, *hence the complexity of the algorithm is* $\mathcal{O}(2^{2r-2} n \ln \frac{(x^{(0)})^T s^{(0)}}{\epsilon})$.

COROLLARY 5.4. *Let us take* $(x^{(0)}, s^{(0)})$ *such that* $\omega_0 \geq \rho$ *holds. If* $r = 1$ *and* n *is sufficiently large then we may choose* $\theta = \frac{2\rho^2(1-\rho^2)}{\sqrt{n}}$, *hence the complexity of the algorithm is* $\mathcal{O}(\frac{n}{\rho^2(1-\rho^2)} \ln \frac{(x^{(0)})^T s^{(0)}}{\epsilon})$.

Before proving polynomial complexity of our algorithm for $r = 0$, note that for all $r \geq 0$ the iterates stay in a fixed neighborhood of the central path. Thus, using the results of Güler and Ye [5] we conclude that our algorithm produces a maximally complementary solution.

6. Polynomial complexity if $r = 0$. In this section we show that, with suitable step-size, the classical primal-dual affine scaling algorithm is polynomial. The obtained complexity bound, $\mathcal{O}(nL^2)$, is the same as obtained by Monteiro, Adler, and Resende [14] and by Mizuno and Nagasawa [13]. Our approach enjoys the advantages of these two results. In the case of LP and convex quadratic programming problems, our complexity result is the same as in the above mentioned papers. We use a fixed step-size as it is in Monteiro, Adler, and Resende's paper [14], but we do not use any potential function which determines the actual step-size as presented by Mizuno and Nagasawa [13]. Contrary to the assumptions in [14], as in [13] our analysis allows an arbitrary, not necessarily centered, starting point. So from now on we assume that $r = 0$. For keeping the discussion simple we assume, as in the previous section, that $n \geq 4$. It is easily verified that Lemmas 4.3 and 4.4 still apply in the present case. Theorem 5.1, however, is not valid if $r = 0$. In fact, by taking the limit of the bounds in Theorem 5.1 as r tends to zero, one obtains that the step size θ also tends to zero. Below we show that by making a positive step (i.e., $\theta > 0$), $\omega(v)$ may well decrease, but the decrease can be bounded from below. In fact, this is the content of the next lemma.

LEMMA 6.1. *If* $(x, s) \in \mathcal{F}^0$ *and*

$$0 \leq \theta \leq \min\left(2\omega(v)\left(\sqrt{1 + \frac{\omega(v)^2}{n}} - \frac{\omega(v)}{\sqrt{n}}\right),\ \sqrt{n}\right) \tag{6.1}$$

then $(\hat{x}, \hat{s}) \in \mathcal{F}^0$ *and*

$$1 + \omega(\hat{v}^2) \geq \frac{1 + \omega(v^2)}{1 + \frac{\theta^2\sqrt{n}}{4(\sqrt{n}-\theta)}}. \tag{6.2}$$

Proof. It may be clear from the proof of Theorem 5.1 that the given bounds (6.1) on θ guarantee the feasibility of the new iterate $(\hat{x}, \hat{s})$. So it remains to show that (6.2) holds. First observe that (5.4) holds also for $r = 0$. Hence, by using the notation $\omega^2 = \omega(v^2) = \frac{\alpha}{\beta}$ with α and β such that $\alpha e \leq xs \leq \beta e$, one has

$$\omega(\hat{v}^2) \geq \frac{(1 - \frac{\theta}{\sqrt{n}})\alpha - \frac{\theta^2\beta}{4}}{(1 - \frac{\theta}{\sqrt{n}})\beta + \frac{\theta^2\beta}{4}} = \frac{4\omega^2(\sqrt{n} - \theta) - \theta^2\sqrt{n}}{4(\sqrt{n} - \theta) + \theta^2\sqrt{n}}.$$

By rearranging the terms, the inequality (6.2) directly follows. □

Now we are ready to prove the polynomial complexity of the classical primal-dual affine scaling algorithm for positive semidefinite LCPs. We will denote by $(x^{(k)}, s^{(k)})$ the iterate after k iterations and for simplicity we use the notation $\omega_k^2 := \omega(x^{(k)} s^{(k)})$.

THEOREM 6.2. *Let an initial interior point* $(x^{(0)}, s^{(0)}) \in \mathcal{F}^0$ *with* $1 \geq \omega_0$ *and* $0 < \epsilon < (x^{(0)})^T s^{(0)}$ *be given. We define parameters* $\tilde{L}$ *and* τ *as follows:*

$$\tilde{L} := \ln \frac{(x^{(0)})^T s^{(0)}}{\epsilon}, \qquad \tau := \frac{2}{\omega_0^2} + \frac{1}{n\tilde{L}},$$

and we assume that $\sqrt{2n}\tilde{L} \geq \omega_0$. *Then, taking* $\theta = \frac{1}{t\sqrt{n}\tilde{L}}$, *where* t *is the (unique) real number in the interval* $[\tau, \tau + \frac{1}{2n\tilde{L}^2})$ *such that* $2tn\tilde{L}^2$ *is integral, after* $2tn\tilde{L}^2 = \mathcal{O}(\frac{n\tilde{L}^2}{\omega_0^2})$ *iterations the algorithm yields a solution* (x^*, s^*) *such that* $(x^*)^T s^* \leq \epsilon$ *and* $\omega(x^* s^*) \geq \frac{\omega_0^2}{2}$.

Proof. For the moment we make the assumption that in each iteration of the algorithm the step size $\theta = \frac{1}{t\sqrt{n}\tilde{L}}$ satisfies the conditions of Lemma 6.1. Later on we will justify this assumption. Taking logarithms in (6.2) and substituting the given value of θ, we obtain

$$\ln \frac{1+\omega_0^2}{1+\omega_k^2} \leq k \ln \left(1 + \frac{\theta^2\sqrt{n}}{4(\sqrt{n}-\theta)}\right) \leq \frac{k\theta^2\sqrt{n}}{4(\sqrt{n}-\theta)} = \frac{k}{4t^2n\tilde{L}^2 - 4t\tilde{L}} = \frac{k}{4t\tilde{L}\left(tn\tilde{L}-1\right)}.$$

Hence we have $\omega_k^2 \geq \frac{\omega_0^2}{2}$ as long as

$$\frac{k}{4t\tilde{L}\left(tn\tilde{L}-1\right)} \leq \ln \frac{1+\omega_0^2}{1+\frac{\omega_0^2}{2}}. \tag{6.3}$$

Since $f(\tau) := \ln((1+\tau)/(1+\tau/2))$ is a concave function and $f(0) = 0,\ f(1) \geq \frac{1}{4}$, one has

$$\ln \frac{1+\omega_0^2}{1+\frac{\omega_0^2}{2}} \geq \frac{\omega_0^2}{4}.$$

As a consequence, the inequality (6.3) is certainly satisfied if

$$k \leq \omega_0^2 t\tilde{L}\left(tn\tilde{L}-1\right). \tag{6.4}$$

We conclude that, to maintain the inequality $\omega_k^2 \geq \frac{\omega_0^2}{2}$, the total number of iterations must satisfy (6.4).

Since Lemma 4.3 is valid, the proof of Theorem 5.2 makes clear (see (5.6) and (5.7)) that the algorithm stops after at most k iterations, where

$$k \geq 2\frac{\sqrt{n}}{\theta} \ln \frac{(x^{(0)})^T s^{(0)}}{\epsilon} = 2tn\tilde{L}^2,$$

and then we have $(x^{(k)})^T s^{(k)} \leq \epsilon$. (Note that the definition of t guarantees that $2tn\tilde{L}^2$ is integral.) So, as far as the gap reduction is concerned, the algorithm needs not more than $2tn\tilde{L}^2$ iterations. This number of iterations will respect the bound (6.4) if

$$2tn\tilde{L}^2 \leq \omega_0^2 t\tilde{L}\left(tn\tilde{L}-1\right).$$

Dividing both sides by $\omega_0^2 tn\tilde{L}^2$, this reduces to the inequality $t \geq \tau$, which clearly is satisfied by the value assigned to t in the theorem.

It remains to show that in each iteration of the algorithm the specified step size θ satisfies condition (6.1) of Lemma 6.1. First, observe that $\theta \leq \sqrt{n}$ is equivalent to $tn\tilde{L} \geq 1$. Since $t \geq \tau$ and $\tau n\tilde{L} \geq 1$, we have $\theta \leq \sqrt{n}$. It remains to deal with the condition that for each k, with $1 \leq k \leq 2tn\tilde{L}^2$,

$$\theta \leq 2\omega_k \left(\sqrt{1 + \frac{\omega_k^2}{n}} - \frac{\omega_k}{\sqrt{n}} \right).$$

Using $n \geq 4$, we have $\frac{\omega_k}{\sqrt{n}} \leq \frac{\omega_k}{2} \leq \frac{1}{2}$. Therefore, since

$$2\left(\sqrt{1+\sigma^2} - \sigma\right) > 1 \quad \text{if} \quad 0 \leq \sigma < \tfrac{3}{4},$$

it is sufficient that $\theta \leq \omega_k$ for each k. As we have seen before, for the given step size we have $\omega_k \geq \frac{\omega_0}{\sqrt{2}}$ for each k. So is it sufficient that θ satisfies $\theta \leq \frac{\omega_0}{\sqrt{2}}$. This amounts to $\omega_0 t\sqrt{n}\tilde{L} \geq \sqrt{2}$. Due to the assumption in the theorem that $\sqrt{2n}\tilde{L} \geq \omega_0$, this certainly holds if t satisfies $\omega_0^2 t \geq 2$. Since $\omega_0^2 \tau \geq 2$ and $t \geq \tau$ it is obvious that t satisfies this inequality. Hence the proof of the theorem is complete. □

Remarks. We can make the results of Theorem 6.2 more concrete as follows.

• If we choose a centered starting point, i.e., $\omega_0 = 1$, then $2 < \tau \leq 3$ and so also $2 < t \leq 3$. Hence in that case the algorithm needs at most $6n\tilde{L}^2$ iterations.

• Let L denote the size of the LCP (1.1). If we assume that for the initial point $(x^{(0)})^T s^{(0)} = \mathcal{O}(2^L)$ and $\epsilon = \mathcal{O}(2^{-L})$, then we solve the LCP in $\mathcal{O}(2tnL^2)$ iterations. If in addition the starting point is centered, then we have $\mathcal{O}(6nL^2)$ complexity.

• Finally, note that although our analysis proves the polynomial complexity of the classical primal-dual affine scaling algorithm for arbitrary, not necessarily centered, starting points, the constant t depends on the (non)centrality of the initial point. A less centered initial point results in a larger t value. Clearly, as ω_0 tends to zero, then t goes to infinity. This behavior is in conformance with the results of section 5.

7. Computational results. To illustrate the effect of the parameter r on the numerical performance of the algorithm, we solved a series of convex quadratic programming problems (reformulated as linear complementarity problems). The problems are coming from statistics and known as *convex regression* problems [3, 6]. We have been given two n-dimensional vectors a and c. The values a_i are the sample points and the values c_i are the observed function (distribution) values. The problem is to find a convex regression function with function values y_i in the sample points a_i, where the distance between the observation (c) and function values (y) is minimal. We have the following formal quadratic programming problem:

$$\begin{array}{ll} \min & \displaystyle\sum_{i=1}^{n} (y_i - c_i)^2 \\ \text{s.t.} & \dfrac{y_{i+1} - y_i}{a_{i+1} - a_i} \geq \dfrac{y_i - y_{i-1}}{a_i - a_{i-1}} \text{ for all } \quad i = 2, 3, \ldots, n-1. \end{array}$$

In our test set the values a_i are random numbers from a uniform distribution on the interval [-1,1] and $c_i = (a_i - 0.5)^2 + e_i$, where e_i has a normal distribution with zero mean and 0.1 variance.

We have implemented our family of algorithms in MATLAB (version 4.0) with efficient sparse matrix handling facility [12, 4]. For the computation we used a 90Mhz

Table 7.1

	The order of scaling: r										
n	0.0	0.01	0.1	0.5	1	2	5	10	15	20	25
10	62.7	48.4	28.4	27.0	26.2	28.3	38.1	50.9	62.0	69.8	73.5^{1}
25	120.1	59.6	36.8	33.7	33.0	34.3	46.8	66.3	83.3	95.8^{1}	107.7^{4}
50	37.6	35.7	40.5	37.7	37.8	40.0	49.7	69.5	88.0	105.4	115.8^{6}
75	44.9	67.8	55.9	49.0	48.8	50.3	60.5	82.8	103.0	120.2^{1}	126.5^{8}
100	144.8	54.6	68.0	57.9	57.4	58.7	69.5	94.4	111.8	131.1^{1}	137.8^{9}
200	68.9	57.7	205.5	107.2	99.9	102.4	115.6	138.0	160.7	176.3^{4}	$* * *^{10}$
300	71.5	69.4	524.4	189.4	154.9	157.3	171.0	195.3	218.5	232.3^{6}	$* * *^{10}$
400	258.1^{3}	199.4^{3}	1078.4	308.5	227.2	221.7	235.6	259.3	282.7	301.6^{5}	319.5^{8}
500	159.8^{2}	209.6^{2}	761.7^{2}	458.6	319.0	301.4	309.6	332.2	355.6	374.5^{5}	394.0^{9}

Table 7.2

	The order of scaling: r										
n	0.0	0.01	0.1	0.5	1	2	5	10	15	20	25
$10^{\max}$	114	65	32	31	30	32	42	59	72	83	87
$10^{\min}$	26	26	22	22	22	24	32	43	52	58	59
$25^{\max}$	653	151	54	44	43	45	60	89	110	103	116
$25^{\min}$	28	28	31	29	29	30	42	60	75	89	102
$50^{\max}$	61	60	44	41	42	44	54	75	95	114	121
$50^{\min}$	30	29	37	36	36	38	47	65	80	96	109
$75^{\max}$	72	293	66	54	53	54	66	93	116	129	127
$75^{\min}$	37	36	50	46	45	47	56	77	95	111	126
$100^{\max}$	980	154	76	62	61	62	76	122	123	138	137
$100^{\min}$	37	36	62	55	54	56	66	87	105	121	137
$200^{\max}$	100	87	303	111	104	106	122	147	174	190	***
$200^{\min}$	41	40	146	105	97	99	110	130	150	169	***
$300^{\max}$	96	113	736	193	159	160	176	203	229	239	***
$300^{\min}$	48	49	318	186	151	153	164	185	204	222	***
$400^{\max}$	1314	904	1598	315	231	227	245	272	297	305	322
$400^{\min}$	56	64	890	304	224	218	231	252	276	297	317
$500^{\max}$	508	833	1813	466	323	304	314	340	364	383	394
$500^{\min}$	66	72	1337	454	316	299	306	328	349	370	394

Pentium processor PC with 32 MB memory. Besides the order of scaling (r), there is one more important parameter in the algorithm, the step-size. As is always the case, the theoretical step-size is too pessimistic in practice. After some experiments we used $\frac{2}{3}$ of the maximal possible step in all the experiments.

In Table 7.1 the average number of iterations needed to reach a 10^{-5} precision in the duality gap is presented for a series of test problems with n taking values in the range from 10 to 1000 and for 11 different values of r in the range from 0 to 25. We have solved 10 problems of each size and with each r. The maximum number of iterations was set to 2500. Only the successful runs were taken into account in calculating the average. If a failure (no solution in 2500 iterations) occurred in solving a certain set of problems then the number of failures is indicated by superscripts.

In Table 7.2 the maximal and the minimal number of iterations of the successful runs are reported. From the computational results presented in Tables 7.1 and 7.2 we draw the following conclusions.

- If r is too small (close to zero or equal to zero), the algorithm becomes very unstable. Sometimes very low iteration numbers occur, while in a slightly different case the iteration number is high or the algorithm fails. This underlines the common knowledge that centering is needed to stabilize IPMs.

Similar problems occur when r is too big (greater than 15). Then the difficulties are due to the high exponents which make sufficiently precise calculations impossible. It seems that there is no universal best r value. The selection of the best r value depends on the dimension.

• Looking at the r values ($0.1 \leq r \leq 15$), one can observe that lowest iteration numbers are consistently obtained for slightly increasing r values as the dimension increases. This means that as the dimension increases, the importance of centering increases as well.

• For fixed r the number of iterations increases almost linearly in the dimension. In efficient IPMs for LP the number of iterations is almost constant, which is not the case with this algorithm. Here the gap between the theoretically worst case and the practical behavior seems to be smaller than in general.

REFERENCES

[1] E. Barnes, *A variation on Karmarkar's algorithm for solving linear programming problems*, Math. Programming, 36 (1986), pp. 174–182.

[2] I. Dikin, *Iterative solution of problems of linear and quadratic programming*, Dokl. Akad. Nauk SSSR, 174 (1967), pp. 747–748 (translated in Soviet Math. Dokl., 8 (1967), pp. 674–675).

[3] R. L. Dykstra, *An algorithm for restricted least squares regression*, J. Amer. Statist. Assoc., 78 (1983), pp. 837–842.

[4] J. R. Gilbert, C. Moiler, and R. Schreiber, *Sparse matrices in Matlab: Design and implementation*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 333–356.

[5] O. Güler and Y. Ye, *Convergence behavior of some interior–point algorithms*, Math. Programming, 60 (1993), pp. 215–228.

[6] M. Hage, *Estimation of Convex Decreasing Densities and Convex Regression Functions*, M.Sc. Thesis, Faculty of Technical Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands, September, 1994.

[7] D. Hertog and C. Roos, *A survey of search directions in interior point methods for linear programming*, Math. Programming, 52 (1991), pp. 481–509.

[8] B. Jansen, C. Roos, and T. Terlaky, *A polynomial primal-dual dikin–type algorithm for linear programming*, Math. Oper. Res., 21 (1996), pp. 341–353.

[9] N. K. Karmarkar, *A new polynomial–time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[10] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise, *A unified approach to interior point algorithms for linear complementarity problems*, Lecture Notes in Computer Science 538, Springer-Verlag, Berlin, New York, 1991.

[11] P. D. Ling, *A new proof of convergence for the new primal–dual affine scaling interior–point algorithm of Jansen, Roos and Terlaky*, Technical report, University of East Anglia, Norwich, August, 1993.

[12] *Matlab Users Guide*, MathWorks Inc., 1993.

[13] S. Mizuno and A. Nagasawa, *A primal–dual affine–scaling potential–reduction algorithm for linear programming*, Math. Programming, 62 (1993), pp. 119–131.

[14] R. Monteiro, I. Adler, and M. 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] R. Saigal, *A three step quadratically convergent implementation of the primal affine scaling method*, Technical report 93–9, Dept. of Industrial and Operational Engineering, University of Michigan, Ann Arbor, MI, February, 1993.

[16] T. Tsuchiya and M. Muramatsu, *Global convergence of the long–step affine scaling algorithm for degenerate linear programming problems*, SIAM J. Optim., 5 (1995), pp. 525–551.

[17] R. Vanderbei, M. Meketon, and B. Freedman, *A modification of Karmarkar's linear programming algorithm*, Algorithmica, 1 (1986), pp. 395–407.

SIAM J. Optim.
Vol. 7, No. 1, pp. 141–161, February 1997

MINIMIZATION OF A LARGE-SCALE QUADRATIC FUNCTION SUBJECT TO A SPHERICAL CONSTRAINT[*]

D. C. SORENSEN[†]

Abstract. An important problem in linear algebra and optimization is the *trust-region subproblem*: minimize a quadratic function subject to an ellipsoidal or spherical constraint. This basic problem has several important large-scale applications including seismic inversion and forcing convergence in optimization methods. Existing methods to solve the trust-region subproblem require matrix factorizations, which are not feasible in the large-scale setting. This paper presents an algorithm for solving the large-scale trust-region subproblem that requires a fixed-size limited storage proportional to the order of the quadratic and that relies only on matrix-vector products. The algorithm recasts the trust-region subproblem in terms of a parameterized eigenvalue problem and adjusts the parameter with a superlinearly convergent iteration to find the optimal solution from the eigenvector of the parameterized problem. Only the smallest eigenvalue and corresponding eigenvector of the parameterized problem needs to be computed. The implicitly restarted Lanczos method is well suited to this subproblem.

Key words. Krylov methods, regularization, constrained quadratic optimization, trust-region, Lanczos method

AMS subject classifications. Primary, 65F15; Secondary, 65G05

PII. S1052623494274374

1. Introduction.

An important problem in linear algebra and optimization is the *trust-region subproblem*: minimize a quadratic function subject to an ellipsoidal constraint. A mathematical statement of the problem is

$$\min\ \tfrac{1}{2}x^TAx + g^Tx \ \ \text{subject to } \|Cx\| \le \Delta,$$

where A is an $n \times n$ symmetric matrix, g is an n vector, x is the unknown n vector, C is a nonsingular matrix, and Δ is a given positive number. The norm is the standard 2-norm, T denotes transpose, and all quantities are real.

This basic problem has many applications. The regularization or smoothing of discrete forms of ill-posed problems such as those arising in seismic inversion and the trust-region mechanism used to force convergence in optimization methods are two examples of significant computational importance. Discussions of the problem of minimizing a quadratic function subject to a quadratic constraint may be found in [5], [6], [10]. Applications to unconstrained optimization algorithms are given in [9], [10], [15], and applications to constrained optimization algorithms are discussed in [1], [2], [4], [13]. For applications to seismic inversion, see [8], [17].

A solution x to the problem must satisfy a relation of the form

$$(A + \mu C^TC)x = -g,$$

with $\|Cx\| = \Delta$. The parameter μ is the regularization parameter for ill-posed problems, and it is the Levenberg–Marquardt parameter in optimization. C is often constructed to impose a smoothness condition on the solution x for ill-posed problems,

[*] Received by the editors September 1, 1994; accepted for publication (in revised form) September 4, 1995. This work was supported in part by DOE contract DE-FG0f-91ER25103, by NSF cooperative agreement CCR-9120008, and by ARPA contract number DAAL03-91-C-0047 (administered by the U.S. Army Research Office).

http://www.siam.org/journals/siopt/7-1/27437.html

[†] Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005-1892 (sorensen@caam.rice.edu).

and it is used to incorporate scaling of the variables in optimization. With a change of variables one can assume $C = I$ and that the trust-region subproblem will minimize a quadratic function subject to a spherical constraint. This case is considered in the following discussion.

If positive definite matrices of the form $A+\mu I$ can be decomposed into a Cholesky factorization, then the method proposed by Moré and Sorensen [10] can be used to solve the problem. In some important applications, e.g., seismic inversion and large-scale constrained optimization, factoring or even forming these matrices is out of the question. A conjugate–gradient-style method for the large-scale trust-region subproblem requiring only matrix-vector products $w \leftarrow Av$ would be highly desirable.

The purpose of this paper is to present an algorithm for solving the large-scale trust-region subproblem that requires a fixed-size limited storage proportional to n and relies only upon matrix-vector products. In some sense the approach developed here may be viewed as an extension of the methods presented by Steihaug [16] and by Toint [18]. The algorithm recasts the trust-region subproblem in terms of a parameterized eigenvalue problem and adjusts the parameter with a superlinearly convergent iteration to find the optimal vector x from the eigenvector of the parameterized problem. Only the smallest eigenvalue and corresponding eigenvector of the parameterized problem has to be computed. The implicitly restarted Lanczos method (IRLM) as implemented in the ARPACK software [7] is one technique that meets the requirements of limited storage and reliance only on matrix-vector products. An algorithm that is designed to solve the related large-scale quadratically constrained least-squares problem is presented in [6]. The author is not aware of another algorithm that is suitable for the general (indefinite) large-scale case.

2. The trust-region subproblem.

The trust-region subproblem has a very interesting mathematical structure that lends itself to efficient computational techniques once the subtlety of the structure is exposed. In this section and throughout the remainder of the paper $C = I$ is assumed and the problem to be considered is

$$\min \tfrac{1}{2} x^T A x + g^T x \quad \text{subject to} \quad \|x\| \le \Delta. \tag{2.1}$$

The optimality conditions for this problem are interesting and computationally attractive since they are both necessary and sufficient and provide a means to reduce the given n-dimensional constrained optimization problem to a zero-finding problem in a single scalar variable. The conditions are given in the following lemma.

LEMMA 2.1. *The vector x is a solution to* (2.1) *if and only if x is a solution to an equation of the form*

$$(A - \lambda I)x = -g,$$

with $A - \lambda I$ positive semidefinite, $\lambda \le 0$, and $\lambda(\Delta - \|x\|) = 0$.
The statement of these conditions is slightly nonstandard in the use of a negative rather than a positive λ. The reason for this will be made clear shortly. A simple proof of this lemma is given in [14].

The method developed by Moré and Sorensen [10] relies upon the ability to compute a Cholesky factorization

$$R_\lambda^T R_\lambda = A - \lambda I,$$

whenever this matrix is positive definite. For any such λ one can solve

$$R_\lambda^T R_\lambda x_\lambda = -g \quad \text{and then} \quad R_\lambda^T q_\lambda = x_\lambda$$

to evaluate the function

$$\phi(\lambda) \equiv \frac{1}{\Delta} - \frac{1}{\|x_\lambda\|}$$

and its derivative

$$\phi'(\lambda) = \frac{\|q_\lambda\|^2}{\|x_\lambda\|^3}$$

and thus apply Newton's method to find a solution to the equation

$$\phi(\lambda) = 0.$$

This method will rapidly find solutions that are on the boundary of the trust region but it is not appropriate for large-scale problems which do not afford a Cholesky decomposition.

It is possible to reparameterize the trust-region subproblem to obtain a scalar problem that is tractable in the large-scale setting. A motivating observation is that for a given real number α,

$$\tfrac{1}{2}\alpha + \psi(x) = \tfrac{1}{2}(1, x^T) \begin{pmatrix} \alpha & g^T \\ g & A \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix},$$

where $\psi(x) \equiv \frac{1}{2}x^T A x + g^T x$.

For a fixed α the goal is to minimize a vertical translation of the function $\psi(x)$ over the set $\{x : 1 + x^T x = 1 + \Delta^2\}$. This suggests that the solution may be found in terms of an eigenpair of the bordered matrix. An eigenvalue λ and corresponding normalized eigenvector $(1, x^T)^T$ of the bordered matrix will satisfy

$$\begin{pmatrix} \alpha & g^T \\ g & A \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} = \begin{pmatrix} 1 \\ x \end{pmatrix} \lambda, \tag{2.2}$$

and it follows that

$$\alpha - \lambda = -g^T x \quad \text{and} \quad (A - \lambda I)x = -g. \tag{2.3}$$

Hence,

$$\alpha - \lambda = g^T (A - \lambda I)^{-1} g = \sum_{j=1}^{n} \frac{\gamma_j^2}{\delta_j - \lambda}, \tag{2.4}$$

where $\{\delta_j\}$ are the eigenvalues of A and $\{\gamma_j\}$ are the expansion coefficients of g in the eigenvector basis.

The bordered matrix appearing on the left in (2.2) will play a key role, and for future reference this matrix will be denoted as B_α. A moment's reflection on the consequences of (2.4) will reveal some very useful information. This equation shows that the eigenvalues of the matrix A interlace the eigenvalues of the bordered matrix B_α. (This is also a consequence of the Cauchy interlace theorem.) Hence, the smallest eigenvalue λ of B_α satisfies $\lambda \leq \delta_1$, where δ_1 is the smallest eigenvalue of A. This assures that the matrix $A - \lambda I$ is positive semidefinite *regardless* of the value of α. Moreover, as long as g is not orthogonal to the eigenspace corresponding to the smallest eigenvalue of A, then the smallest eigenvalue of B_α is often well separated from the rest of the spectrum of B_α, especially for smaller values of Δ. This can be seen best through a graphical study of equation (2.3) and the relations that follow. In cases where it is well separated, a Lanczos-type algorithm should be quite successful in computing this eigenvalue and the corresponding eigenvector.

Equation (2.3) defines λ and, hence, x implicitly as functions of α. Let the function ϕ be defined by

$$\phi(\lambda) \equiv g^T(A-\lambda I)^{-1}g = -g^T x.$$

Then

$$\phi'(\lambda) = g^T(A-\lambda I)^{-2}g = x^T x,$$

where differentiation is with respect to λ and $(A-\lambda I)x = -g$.

Finding the smallest eigenvalue and corresponding eigenvector of the bordered matrix B_α for a given value of α and then normalizing the eigenvector to have its first component equal to one provides a means to evaluate the rational function ϕ and its derivative at values of $\lambda < \delta_1$, the smallest eigenvalue of A. If α can be adjusted so the corresponding x satisfies $\phi'(\lambda) = x^T x = \Delta^2$ with $\alpha - \lambda = \phi(\lambda)$, then

$$(A-\lambda I)x = -g, \quad \lambda(\Delta - \|x\|) = 0,$$

with $A - \lambda I$ positive semidefinite. If $\lambda \le 0$ then x is optimal and solves the trust-region subproblem. If $\lambda > 0$ is found with $\|x\| < \Delta$ during the course of adjusting α, then A is positive definite and the solution to the trust-region subproblem is the unconstrained minimizer $-A^{-1}g$. The only other possibility is that the eigenvector of the bordered matrix has first component *zero* and thus cannot be normalized to have its first component equal to one. This is equivalent to the so-called *hard case* analyzed in [10]. The hard case is discussed at length in section 5.

This development has led to a reformulation of the trust-region subproblem in terms of a parameterized eigenvalue problem. In fact, a sequence of eigenvalue problems will have to be solved in order to iteratively adjust the parameter α to produce the optimal λ and x. Therefore, if this observation is to be helpful, a rapidly convergent method must be devised to adjust α to the optimal value, and an efficient method for computing the smallest eigenvalue and corresponding eigenvector of the bordered matrix must be available. Keeping in mind the assumption that only matrix-vector products $w \leftarrow Av$ are available, a Lanczos method seems to be a natural choice for an eigenvalue method. A well-suited variant of the Lanczos method is presented in the next section. This will be followed with the development of a rapidly convergent iteration to adjust α.

3. The implicitly restarted Lanczos method (IRLM). Lanczos methods have been used extensively to solve large, sparse symmetric eigenvalue problems $Ax = \lambda x$. In exact arithmetic, the Lanczos process is a scheme to tridiagonalize a symmetric $A \in \mathcal{R}^{n\times n}$. After j-steps of the Lanczos process, an orthonormal $n \times j$ matrix V_j and a symmetric tridiagonal matrix T_j are produced such that

$$AV_j = V_j T_j \quad + f_j e_j^T, \tag{3.1}$$

where f_j is a vector of length n with $V_j^T f_j = 0$, and e_j is the jth coordinate vector of length j. This is easily shown to be a truncation of the complete orthogonal reduction of A to tridiagonal form that typically precedes the implicitly-shifted tridiagonal QR iteration.

The eigenvalues of T_j approximate a subset of eigenvalues of A. If μ, y is an eigenpair for T_j (i.e., $T_j y = y\mu$) then $\mu, x = V_j y$ is an approximate eigenpair for A and the error of approximation is given by

$$\|Ax - x\mu\| = \|f_j\| |e_j^T y|. \tag{3.2}$$

In particular, the approximation is exact when $f_j = 0$. Eigenvalues and eigenvectors of the symmetric tridiagonal matrix T_j may be determined by the symmetric QR method or some other suitable technique.

There are a number of numerical difficulties with the original Lanczos process and these difficulties have been addressed extensively in the literature [12]. The method developed in [14] provides an alternate approach to the classic numerical difficulties associated with the Lanczos process. The underlying idea in [14] is to recognize that the residual vector f_j is a function of the initial starting vector (i.e., the first column of V_j) and to then adjust this starting vector to make the residual vector vanish. The total number of Lanczos steps is limited to a fixed prescribed value k and the starting vector is iteratively updated in a way that forces the norm of the residual vector f_k to converge to zero.

The iteration involves repeated application of *polynomial filters* to the starting vector and an in-place updating of the k-step Lanczos factorization. The iteration repeatedly updates the starting vector: $v_1 \leftarrow \pi(A)v_1$, where the polynomial π is applied implicitly through a mechanism directly related to the implicitly-shifted QR technique. The polynomial is constructed to damp undesirable eigenvector components from the starting vector, forcing it into an invariant subspace. This leads to termination of the Lanczos sequence which begins with this starting vector in precisely k steps with $f_k = 0$. The k eigenvalues of the associated T_k will be the eigenvalues of interest. The construction and application of these polynomials, how to update in-place, and other related details are explained in [14]. The technique is analogous to the implicitly-shifted QR iteration for dense matrices and shares a number of important numerical properties associated with that process.

With respect to the subject of this paper, the major advantage of this implicit restart approach is the following:

- *Fixed space.* In this scheme, the number of Lanczos basis vectors never exceeds a prespecified bound that is proportional to the number of eigenvalues sought. Moreover, as in the basic Lanczos process, only matrix-vector products are required with A. Peripheral storage of basis vectors for eigenvector construction is not required.

By virtue of the fixed modest number of Lanczos basis vectors, it is computationally feasible to maintain full numerical orthogonality among the basis vectors. The maintenance of orthogonality ensures that no spurious eigenvalues are computed. This method is referred to as IRLM.

4. Adjusting alpha. Recasting the trust-region problem as a parameterized eigenvalue problem together with the IRLM provides a viable approach to large-scale problems if the optimal parameter α can be computed rapidly. Recall that the goal is to adjust α so that

$$\alpha - \lambda = \phi(\lambda), \quad \phi'(\lambda) = \Delta^2,$$

where

$$\phi(\lambda) = -g^T x, \quad \phi'(\lambda) = x^T x,$$

with $(A - \lambda I)x = -g$. One possibility would be to apply a standard iteration such as the secant method to the problem

$$\frac{1}{\Delta} - \frac{1}{\|x_{\lambda(\alpha)}\|} = 0.$$

The approach adopted here is to develop a special interpolation-based iteration that takes advantage of the structure of the problem. This interpolation-based iterative method will take the following form: let $\hat{\phi}(\lambda)$ interpolate ϕ and ϕ' at some previous iterate(s).

```
ALGORITHM 1.
  1. Initialize $\alpha \leftarrow 0$ and compute the smallest eigenvalue and corresponding
     normalized eigenvector of $B_\alpha$ to obtain the initial iterates $\lambda$ and $x$;
  2. While $\left(\left|\frac{\|x\|-\Delta}{\Delta}\right| > tol\right)$
       (a) Construct the interpolant $\hat{\phi}$ based on the current and perhaps previous
           iterates;
       (b) Let $\hat{\lambda}$ satisfy $\hat{\phi}'(\hat{\lambda}) = \Delta^2$;
       (c) Put $\alpha_+ = \hat{\lambda} + \hat{\phi}(\hat{\lambda})$;
       (d) Compute the smallest eigenvalue and corresponding normalized eigen-
           vector of $B_{\alpha_+}$ to get the new iterates $\lambda_+$ and $x_+$;
     End
```

Two iterations of this type will be developed. One is based on just the previous iterate and the other on the previous two iterates. The first is linearly convergent and the second will prove to be superlinearly convergent. The initialization of $\alpha \leftarrow 0$ assures that the smallest eigenvalue of B_α is nonpositive and thus will satisfy the optimality condition for the multiplier (cf. Lemma 2.1).

To construct the single point method, consider an interpolant of the form

$$\hat{\phi}(\lambda) = \frac{\gamma^2}{\delta - \lambda}.$$

Let x_1 and λ_1 denote the current iterates corresponding to α so that

$$\alpha - \lambda_1 = -g^T x_1 \quad \text{with} \quad (A - \lambda_1 I)x_1 = -g.$$

The interpolant must satisfy

$$\frac{\gamma^2}{\delta - \lambda_1} = -g^T x_1 \quad \text{and} \quad \frac{\gamma^2}{(\delta - \lambda_1)^2} = x_1^T x_1,$$

and from this it is straightforward to derive

$$\delta = \lambda_1 - \frac{g^T x_1}{x_1^T x_1} \quad \text{and} \quad \gamma^2 = \frac{(g^T x_1)^2}{x_1^T x_1}.$$

It is easy to show that $\delta = \frac{x_1^T A x_1}{x_1^T x_1}$, and this is a nice feature since it implies $\delta_1 \le \delta$ where δ_1 is the smallest eigenvalue of A. The formula for $\hat{\lambda}$ in step 2 of Algorithm 1 is given by

$$\hat{\lambda} = \delta + \frac{g^T x_1}{\|x_1\|\Delta},$$

and the updating formula to obtain α_+ at step 3 is shown to be

$$\alpha_+ = \hat{\lambda} + \frac{\gamma^2}{\delta - \hat{\lambda}} = \alpha + \frac{(\alpha - \lambda_1)}{\|x_1\|}\left[\frac{\Delta - \|x_1\|}{\Delta}\right]\left[\Delta + \frac{1}{\|x_1\|}\right]$$

after a little algebraic manipulation. This method may be shown to be linearly convergent, but this convergence may be slow in some cases so it will not suffice to solve the entire problem. However, it may be used to obtain a second iterate from an initial guess to provide the starting values needed to initiate a method based upon interpolating two previous iterates at each step. Since this is the only role it will play in the algorithm, a convergence proof is not given here.

The two-point method is based upon an interpolant of the form

$$\hat{\phi}(\lambda) = \frac{\gamma^2}{\delta - \lambda} + \beta(\delta - \lambda) + \eta.$$

Let x_1 and λ_1 denote the current iterates and let x_2 and λ_2 denote the previous ones. The pole δ is defined by

$$\delta = \min\left(\delta_{\min}, \frac{x_1^T A x_1}{x_1^T x_1}\right) \quad \text{if} \quad \|x_1\| < \Delta \text{ or } \|x_2\| < \Delta$$

or

$$\delta = \max\left(\frac{x_1^T A x_1}{x_1^T x_1}, \frac{x_2^T A x_2}{x_2^T x_2}\right) \quad \text{if} \quad \|x_1\| > \Delta \text{ and } \|x_2\| > \Delta,$$

and then $\delta_{\min} \leftarrow \min(\delta_{\min}, \delta)$. Here, $\delta_{\min} \geq \delta_1$ is the current best estimate to δ_1, the smallest eigenvalue of A. Initially, $\delta_{\min}$ is set to $\frac{x_1^T A x_1}{x_1^T x_1}$, where x_1 is the first iterate obtained from the one point interpolation formula. These conditions have been designed to assure that the iterates obtained by this interpolation scheme will be well defined (see section 6).

The remaining three coefficients are determined to satisfy

$$\hat{\phi}(\lambda_1) = -g^T x_1, \quad \hat{\phi}'(\lambda_1) = x_1^T x_1, \quad \hat{\phi}'(\lambda_2) = x_2^T x_2.$$

Satisfying the derivative conditions requires

$$\frac{\gamma^2}{(\delta - \lambda_1)^2} - \beta = x_1^T x_1, \quad \frac{\gamma^2}{(\delta - \lambda_2)^2} - \beta = x_2^T x_2, \tag{4.1}$$

and it follows that

$$\gamma^2 = \frac{[x_2^T x_2 - x_1^T x_1][(\delta - \lambda_1)(\delta - \lambda_2)]^2}{(\lambda_2 - \lambda_1)[2\delta - (\lambda_1 + \lambda_2)]}, \tag{4.2}$$

$$\beta = \frac{\gamma^2}{(\delta - \lambda_1)^2} - x_1^T x_1 = \frac{x_2^T x_2(\delta - \lambda_2)^2 - x_1^T x_1(\delta - \lambda_1)^2}{(\lambda_2 - \lambda_1)[2\delta - (\lambda_1 + \lambda_2)]}, \tag{4.3}$$

and

$$\eta = -g^T x_1 - \beta(\delta - \lambda_1) - \frac{\gamma^2}{(\delta - \lambda_1)}.$$

The formula for $\hat{\lambda}$ in step 2 is derived from the condition

$$\frac{\gamma^2}{(\delta-\hat{\lambda})^2}-\beta=\Delta^2$$

and yields

$$\hat{\lambda}=\delta-\sqrt{\frac{\gamma^2}{\Delta^2+\beta}}. \tag{4.4}$$

Finally, the formula for α_+ is

$$\alpha_+=\hat{\lambda}+\eta+\beta(\delta-\hat{\lambda})+\frac{\gamma^2}{\delta-\hat{\lambda}}. \tag{4.5}$$

The formula (4.5) is, unfortunately, plagued with numerical cancellation problems, and computational experience has shown that this will prevent superlinear convergence when the quantity $\left|\frac{\|x\|-\Delta}{\Delta}\right|$ falls below the square root of working precision (i.e., below 10^{-8} when working in double precision on a SUN workstation). After considerable manipulation one may arrive at a mathematically equivalent update formula that does achieve superlinear convergence to the level of working precision. This formula is

$$\alpha_+=\alpha+\frac{(\delta-\lambda_1)\omega}{(1+\sqrt{1+\omega})\sqrt{1+\omega}}\left[\frac{\Delta^2-x_1^Tx_1}{1+\sqrt{1+\omega}}+x_1^Tx_1+1\right], \tag{4.6}$$

where

$$\omega=\left[\frac{\Delta^2-x_1^Tx_1}{x_2^Tx_2-x_1^Tx_1}\right]\left[\left(\frac{\delta-\lambda_1}{\delta-\lambda_2}\right)^2-1\right].$$

Of course, the formula (4.6) is only used in place of formula (4.5) when the quantity $1+\omega\geq 0$ and this is eventually satisfied as $\lambda_j\to\lambda_*$.

Considering the branch of the function ϕ that is supposed to be approximated by these formulas, it is desirable that the formula (4.2) yields a positive number and that the number $\beta+\Delta^2$ appearing under the square root sign in (4.4) is also positive so that the iteration will be well defined. These conditions are indeed satisfied and this will be established in section 6. In section 6 it will also be established that the iteration based upon the two point formula is locally and superlinearly convergent. However, both iterations can break down when faced with the so-called *hard case*.

5. The hard case.

There is one particularly difficult situation that may arise in trust-region problems. This is referred to in [10] as the hard case. It can only occur when the vector g is orthogonal to the eigenspace $\mathcal{S}_1\equiv\{q : Aq=q\delta_1\}$ corresponding to the smallest eigenvalue δ_1 of A. The precise statement is as follows.

LEMMA 5.1. *Let* $p=-(A-\delta_1 I)^\dagger g$. *If* $\delta_1\leq 0$ *and* $\|p\|<\Delta$ *then the solutions to* (2.1) *consist of the set*

$$\mathcal{S}_o\equiv\{x : x=p+z,\quad z\in\mathcal{S}_1,\quad \|x\|=\Delta\}.$$

In the statement of Lemma 5.1 the symbol † denotes the Moore–Penrose generalized inverse. This lemma is proved in [14] and its computational implications are discussed in [10]. The following lemma is a restatement of a result given in [10] that is useful in dealing with the hard case.

LEMMA 5.2. *Let* $0 < \sigma < 1$ *be given and suppose*

$$(A - \lambda I)p = -g, \quad \lambda \leq 0,$$

with $(A - \lambda I)$ *positive semidefinite. If*

$$\|p + z\| = \Delta \;\; \textit{and} \;\; z^T(A - \lambda I)z \leq -\sigma(g^T p + \lambda\Delta^2)$$

then

$$\psi_* \leq \psi(p + z) \leq \tfrac{1}{2}(1 - \sigma)(g^T p + \lambda\Delta^2) \leq (1 - \sigma)\psi_*,$$

where $\psi_* \leq 0$ *is the optimal value of* (2.1).

Moré and Sorensen used this lemma to detect near hard case behavior and terminate the iterative solution to (2.1) early. In that setting, explicit eigeninformation was not available and deemed too expensive to obtain. Instead, a suitable point z was obtained from the LINPACK condition estimator [3] applied to the Cholesky factor of $(A - \lambda I)$. In the present setting, the Cholesky factor is not computed but the necessary eigeninformation will be readily available.

The reformulation leading to the key relation (2.3) depends upon the ability to normalize the selected eigenvector of the bordered matrix to have its first component set to one. This is of course impossible when the first component of this eigenvector vanishes. Interestingly enough, the hard case occurs precisely when this happens.

LEMMA 5.3. *Every vector of the form* $(0, q^T)^T$ *with* $q \in \mathcal{S}_1$ *is an eigenvector of the bordered matrix*

$$\begin{pmatrix} \alpha & g^T \\ g & A \end{pmatrix}$$

if and only if g *is orthogonal to* $\mathcal{S}_1$.

The proof of this lemma is straightforward and will be omitted.

Generally, a near hard case condition is painfully obvious in practice. If the search for the optimal α discussed in section 4 is initiated with $\alpha = 0$ then the first iterate or its successor given by the one point interpolation formula typically will have an extremely small first component in the eigenvector corresponding to the smallest eigenvalue of the bordered matrix B_α. If the vector $(\nu, q^T)^T$ is an eigenvector of length one for the bordered matrix corresponding to the smallest eigenvalue λ, then satisfying a test of the form

$$\sqrt{1 - \nu^2} > \kappa\Delta|\nu|$$

with $\kappa >> 1$ detects the hard case. Moreover, since $(A - \lambda I)q = -g\nu$ it follows that

$$\frac{\|(A - \lambda I)q\|}{\|q\|} = \frac{\|g\||\nu|}{\sqrt{1 - \nu^2}} \leq \frac{\|g\|}{\kappa\Delta}, \tag{5.1}$$

and choosing $\kappa = \frac{\|g\|}{\epsilon\Delta}$ assures $\frac{\|(A - \lambda I)q\|}{\|q\|} \leq \epsilon$ and hence that λ , q are an approximate eigenpair for A.

If a hard case condition has been detected, set $\lambda_U = \lambda$ (note λ_U is an upper bound on the optimal λ_*) and $z = q/\|q\|$. Put

$$\rho \equiv z^T A z = \lambda_U - \nu(g^T q)/(q^T q)$$

and enter the following iteration with x_1, λ_1 as the most recent iterates obtained before detection of the hard case.

```
ALGORITHM 2.
Let $\theta \in (0,1)$ and $\lambda_1 < \lambda_* < \lambda_U$.
Repeat:
  1. $\alpha \leftarrow (1-\theta)\lambda_U + \theta\lambda_1 - g^T x_1 + (1-\theta)(\lambda_U - \lambda_1)(x_1^T x_1)$;
  2. Compute $\lambda$ and $(\nu, q^T)^T$ the smallest eigenvalue and corresponding vector of $B_\alpha$;
  3. Put $x_2 \leftarrow q/\nu$ , $\lambda_2 \leftarrow \lambda$ and let $\tau$ satisfy $\|x_2 + z\tau\| = \Delta$;
  4. If $\tau^2(\rho - \lambda_2) < -\sigma(g^T x_2 + \lambda_2\Delta^2)$ then stop with $x \leftarrow x_2 + z\tau$;
  5. If $\|x_2\| > \Delta$ then $\lambda_U \leftarrow \lambda_2$ , $\alpha \leftarrow \min(2 * \lambda_U, \alpha - |\alpha|)$ else $x_1 \leftarrow x_2$, $\lambda_1 \leftarrow \lambda_2$;
End
```

In the computational tests that follow, $\theta = .0001$, $\sigma = .000001$, and $\epsilon = .00001$ in the formula for κ as defined above.

Note that on entering this hard case iteration λ_U will be a good under estimate to δ_1, the smallest eigenvalue of A. Moreover, λ_U will be the first iterate to be greater than the optimal λ_*. If a previous iterate to the right of λ_* did not pass the hard case test then no subsequent iterates will either, due to safeguarding. This assures that the bracketing condition at the beginning of Algorithm 2 is valid. Finally, the positive number σ in step 4 is the σ of Lemma 5.2.

The update at step 1 is derived from linear interpolation of ϕ and its first derivative at λ_1 and then solving for the α that would produce a new $\hat{\lambda} = (1-\theta)\lambda_U + \theta\lambda$ if ϕ were linear. In other words, α satisfies

$$\alpha - \hat{\lambda} = \phi(\lambda_1) + \phi'(\lambda_1)(\hat{\lambda} - \lambda_1) = -g^T x_1 + x_1^T x_1(\hat{\lambda} - \lambda_1),$$

with $\hat{\lambda} = \theta\lambda_1 + (1-\theta)\lambda_U$.

Since ϕ is convex on the interval $(-\infty, \delta_1)$, the new λ_2 obtained by solving the bordered problem with this α will satisfy $\lambda_1 < \lambda_2 < \hat{\lambda}$. Moreover, the length of the interval (λ_1, λ_U) will always shrink.

LEMMA 5.4. *Assume $\theta < \frac{1}{4}$. Let λ_1^+ and λ_U^+ be the updated values of λ_1 and λ_U obtained from one pass through the hard case iteration. Then*

$$|\lambda_U^+ - \lambda_1^+| \leq (1-\theta)|\lambda_U - \lambda_1|.$$

Proof. By its construction, λ_2 will satisfy $\phi(\lambda_2) = \alpha - \lambda_2$. Substituting the defined value of α gives

$$\phi(\lambda_2) = (1-\theta)\lambda_U + \theta\lambda_1 + \phi(\lambda_1) + (1-\theta)(\lambda_U - \lambda_1)(x_1^T x_1) - \lambda_2.$$

Rearranging terms will give

$$\phi(\lambda_2) - \phi(\lambda_1) = \lambda_1 - \lambda_2 + (1-\theta)[1 + x_1^T x_1](\lambda_U - \lambda_1). \tag{5.2}$$

It is straightforward to show

$$\phi(\lambda_2) - \phi(\lambda_1) = (\lambda_2 - \lambda_1)x_2^T x_1,$$

and substituting this into (5.2) and rearranging terms will give

$$(\lambda_2 - \lambda_1)(1 + x_2^T x_1) = (1-\theta)(1 + x_1^T x_1)(\lambda_U - \lambda_1).$$

If $\|x_2\| < \Delta$ then $\lambda_1^+ = \lambda_2$ and $\lambda_U^+ = \lambda_U$. Hence,

$$\begin{aligned}
\lambda_U^+ - \lambda_1^+ &= \lambda_U - \lambda_2 \\
&= \lambda_U - \lambda_1 - (\lambda_2 - \lambda_1) \\
&= \left[1 - (1-\theta)\frac{(1 + x_1^T x_1)}{(1 + x_2^T x_1)}\right](\lambda_U - \lambda_1) \\
&= \left[\frac{(x_2 - x_1)^T x_1}{(1 + x_2^T x_1)} + \theta\frac{(1 + x_1^T x_1)}{(1 + x_2^T x_1)}\right](\lambda_U - \lambda_1). \qquad (5.3)
\end{aligned}$$

Now, if $\lambda_2 - \lambda_1 < \frac{1}{4}(\lambda_U - \lambda_1)$ then

$$\begin{aligned}
\frac{(x_2 - x_1)^T x_1}{(1 + x_2^T x_1)} &= \frac{(\lambda_2 - \lambda_1) x_1^T A_2^{-1} x_1}{(1 + x_2^T x_1)} \\
&\leq \frac{(\lambda_2 - \lambda_1)}{(\delta_1 - \lambda_2)} \frac{x_1^T x_1}{(1 + x_2^T x_1)} \\
&\leq \left[\frac{\frac{1}{4}(\lambda_U - \lambda_1)}{(\delta_1 - \lambda_1) - (\lambda_2 - \lambda_1)}\right] \frac{x_1^T x_1}{(1 + x_1^T x_1)} \\
&\leq \frac{\frac{1}{4}(\lambda_U - \lambda_1)}{\frac{3}{4}(\lambda_U - \lambda_1)} \\
&= \tfrac{1}{3},
\end{aligned}$$

where $A_2 \equiv A - \lambda_2 I$. Thus

$$\lambda_U^+ - \lambda_1^+ \leq (\tfrac{1}{3} + \theta)(\lambda_U - \lambda_1) < \tfrac{3}{4}(\lambda_U - \lambda_1)$$

follows from (5.3). If $\lambda_2 - \lambda_1 \geq \frac{1}{4}(\lambda_U - \lambda_1)$ then

$$\lambda_U^+ - \lambda_1^+ = \lambda_U - \lambda_2 = (\lambda_U - \lambda_1) - (\lambda_2 - \lambda_1) \leq \tfrac{3}{4}(\lambda_U - \lambda_1),$$

and in both cases the desired result holds since $\frac{1}{4} < (1 - \theta)$. Now suppose $\|x_2\| \geq \Delta$. Then $\lambda_U^+ = \lambda_2$ and $\lambda_1^+ = \lambda_1$ and it follows that

$$\lambda_U^+ - \lambda_1^+ = (1-\theta)\frac{(1 + x_1^T x_1)}{(1 + x_2^T x_1)}(\lambda_U - \lambda_1) < (1-\theta)(\lambda_U - \lambda_1).$$

This establishes the result. □

This result establishes convergence but is far from indicative of what will occur in practice. A value $\theta = .001$ works well in practice even though this lemma would indicate a potentially slow rate of convergence with this value. This is because the point λ_2 almost always satisfies $\|x_2\| < \Delta$.

Satisfaction of the stopping rule at step 4 assures that the conditions of Lemma 5.2 are satisfied so the accepted point x_1 satisfies

$$\psi(x_*) \leq \psi(x_1) \leq (1 - \sigma)\psi(x_*).$$

In many applications, including the two mentioned previously, a value of $\sigma = .01$ is used and this is generally satisfied very rapidly indeed.

6. Convergence. In this section, the issues of forcing convergence and determining the rate of local convergence will be discussed. It will be shown that the iterates based upon the two point rational interpolation formulas are well defined and are locally convergent at a superlinear rate. This may be of considerable interest computationally, since evaluating the function ϕ and its derivative requires the computation of the smallest eigenvalue and corresponding eigenvector of the bordered matrix B_α, and this is potentially very expensive. Note, however, in practice one is often interested in just a few digits of accuracy and then superlinear convergence is of little consequence. Nevertheless, it is reassuring to know this rapid convergence can be expected when higher accuracy is needed.

There is very little to say about safeguarding. Perhaps in the future with more computational experience this will become an important issue. In the computational results presented here, a fairly standard simple safeguard was used to obtain an *interval of uncertainty* and then to assure that this interval is updated on each iteration and required to decrease. This safeguard rarely forced a modification of the step given by the two point formula in Algorithm 1.

The main purpose of this section is to establish the local superlinear convergence of the iteration defined by Algorithm 1. It must be established that in the standard case where Algorithm 1 applies, the iterates are well defined in a neighborhood and converge to the solution at a superlinear rate. This is formally stated in the following.

THEOREM 6.1. *Suppose the solution x_* to* (2.1) *is on the boundary of the trust region and that $\{\lambda_k\}$ is a sequence of iterates produced by Algorithm* 1 *using the two point scheme (with λ_k, λ_{k+1} corresponding to λ, λ_+, respectively). Then there is a neighborhood $\mathcal{N}$ of λ_* such that $\lambda_1, \lambda_2 \in (\mathcal{N})$ implies the sequence $\{\lambda_k\}$ will be well defined, remain in $\mathcal{N}$, and converge superlinearly to λ_* with the corresponding iterates x_k converging superlinearly to x_*.*

The proof of this theorem is through a sequence of three lemmas. Lemma 6.2 shows that the iterates are well defined by establishing the validity of (4.4) regardless of whether or not they are close to the solution. Lemma 6.3 is a technicality used to establish the basic asymptotic results. These asymptotic results established in Lemma 6.4 give the final result.

In order to present this local convergence result as simply as possible, it shall be useful to introduce some notation. The subscript 1 shall indicate the most recent iterate, and the subscript 2 shall denote the previous iterate. Thus λ_1 and λ_2 are the current and previous approximations to the optimal λ_*, and λ_1 is the smallest eigenvalue of the bordered matrix B_α. The updated λ_+ is the smallest eigenvalue of the updated B_{α_+}, and α_* will denote the value of α that gives the optimal parameter λ_* and corresponding solution vector x_*. The notation $A_j \equiv A - \lambda_j I$ for $j = 1, 2$ and $A_* \equiv A - \lambda_* I$ will be used. Thus $x_j = -A_j^{-1} g$ for $j = 1, 2$ and $x_* = -A_*^{-1} g$. At a general point λ the notation $A_\lambda \equiv A - \lambda I$ and $x_\lambda \equiv -A_\lambda^{-1} g$ will be used. Finally, the notation $\mathcal{O}((\lambda_* - \lambda_1)^j)$ will be used to denote a quantity whose absolute value is bounded by a fixed positive constant times the quantity $|\lambda_* - \lambda_1|^j$ for any value of λ_1 in a sufficiently small neighborhood of λ_* $(j = 0, 1, 2)$.

First, the fact that the iterates are well defined shall be established. In this development it is useful to note

$$\begin{aligned} x_2^T x_2 - x_1^T x_1 &= g^T (A_2^{-2} - A_1^{-2}) g \\ &= g^T A_2^{-2} (A_1 - A_2)(A_1 + A_2) A_1^{-2} g \\ &= (\lambda_2 - \lambda_1)[x_2^T A_1^{-1} x_2 + x_1^T A_2^{-1} x_1]. \end{aligned} \tag{6.1}$$

From this it follows that

$$\begin{aligned}\gamma^2 &= \frac{[x_2^T x_2 - x_1^T x_1][(\delta-\lambda_1)(\delta-\lambda_2)]^2}{(\lambda_2-\lambda_1)[2\delta-(\lambda_1+\lambda_2)]}\\ &= \frac{[x_2^T A_1^{-1} x_2 + x_1^T A_2^{-1} x_1][(\delta-\lambda_1)(\delta-\lambda_2)]^2}{2\delta-(\lambda_1+\lambda_2)}. \qquad (6.2)\end{aligned}$$

Now, with the exception of the hard case, the smallest eigenvalue of the bordered matrix B_α is always less than the smallest eigenvalue δ_1 of A and $\delta > \delta_1$. Hence, $x_2^T A_1^{-1} x_2 > 0$, $x_1^T A_2^{-1} x_1 > 0$, and $2\delta - (\lambda_1+\lambda_2) > 0$. Therefore, the formula (4.2) for γ^2 does indeed yield a positive number.

Moreover, the number $\Delta^2 + \beta$ appearing under the square root sign in (4.4) is always nonnegative.

LEMMA 6.2. *The quantity* $\Delta^2+\beta$ *in* (4.4) *is always nonnegative.*

Proof. If either $x_1^T x_1 \le \Delta^2$ or $x_2^T x_2 \le \Delta^2$ then $\Delta^2+\beta \ge 0$, since

$$\begin{aligned}\Delta^2+\beta &= \frac{\gamma^2}{(\delta-\lambda_1)^2} + (\Delta^2 - x_1^T x_1)\\ &= \frac{\gamma^2}{(\delta-\lambda_2)^2} + (\Delta^2 - x_2^T x_2)\end{aligned}$$

is implied by (4.1). Otherwise, it may be assumed without loss of generality that $x_*^T x_* \equiv \Delta^2 < x_1^T x_1 < x_2^T x_2$ and hence that $\lambda_* < \lambda_1 < \lambda_2$. In this case the pole δ satisfies $\delta = \max(\frac{x_1^T A x_1}{x_1^T x_1}, \frac{x_2^T A x_2}{x_2^T x_2})$. Observe that the function

$$\rho(\lambda) \equiv \frac{x_\lambda^T A x_\lambda}{x_\lambda^T x_\lambda}$$

is decreasing on the interval (λ_1, δ_1) since the Cauchy–Schwarz inequality implies

$$(x_\lambda^T A_\lambda x_\lambda)(x_\lambda^T A_\lambda^{-1} x_\lambda) \ge (x_\lambda^T A_\lambda^{1/2} A_\lambda^{-1/2} x_\lambda)^2 = (x_\lambda^T x_\lambda)^2, \qquad (6.3)$$

and hence

$$\rho^{'}(\lambda) = 2\left[1 - \frac{(x_\lambda^T A_\lambda x_\lambda)(x_\lambda^T A_\lambda^{-1} x_\lambda)}{(x_\lambda^T x_\lambda)^2}\right] \le 0$$

for all $\lambda \in (\lambda_1, \delta_1)$. It follows that

$$\delta - \lambda \ge \rho(\lambda_1) - \lambda \ge \rho(\lambda) - \lambda = \frac{(x_\lambda^T A_\lambda x_\lambda)}{x_\lambda^T x_\lambda} > 0$$

for all $\lambda \in (\lambda_1, \delta_1)$. From (4.3) it may be found that

$$\Delta^2+\beta = \frac{(x_2^T x_2 - x_*^T x_*)(\delta-\lambda_2)^2 - (x_1^T x_1 - x_*^T x_*)(\delta-\lambda_1)^2}{(\delta-\lambda_1)^2 - (\delta-\lambda_2)^2}. \qquad (6.4)$$

Now, $\lambda_* < \lambda_1 < \lambda_2 < \delta$ implies $\delta - \lambda_* > \delta - \lambda_1 > \delta - \lambda_2 > 0$ so the denominator in (6.4) is positive and the result will be established if it is shown that the function

$$\sigma(\lambda) \equiv (x_\lambda^T x_\lambda - x_*^T x_*)(\delta-\lambda)^2$$

is strictly increasing on the interval (λ_*, δ). Differentiating σ with respect to λ gives

$$\begin{aligned} \sigma^{'}(\lambda) &= 2(\delta - \lambda)[x_\lambda^T A_\lambda^{-1} x_\lambda(\delta - \lambda) - (x_\lambda^T x_\lambda - x_*^T x_*)] \\ &\geq 2(\delta - \lambda)(x_\lambda^T x_\lambda)\left[\frac{(x_\lambda^T A_\lambda x_\lambda)(x_\lambda^T A_\lambda^{-1} x_\lambda)}{(x_\lambda^T x_\lambda)^2} - 1 + \frac{x_*^T x_*}{x_\lambda^T x_\lambda}\right] \\ &> 0, \end{aligned} \tag{6.5}$$

which again follows from (6.3). This implies $\sigma(\lambda)$ is increasing on the interval $\lambda \in (\lambda_1, \delta_1)$, and since

$$\Delta^2 + \beta = \frac{\sigma(\lambda_2) - \sigma(\lambda_1)}{(\delta - \lambda_1)^2 - (\delta - \lambda_2)^2},$$

it follows that $\Delta^2 + \beta > 0$ when $\lambda_* < \lambda_1 < \lambda_2 < \delta$ and the result is established. □

It has just been demonstrated that the iterates are well defined and it is now necessary to establish the local rate of convergence. To this end it is useful to establish a technical lemma that will facilitate the proof of the final desired result.

LEMMA 6.3. *The intermediate point* $\hat{\lambda}$ *given by* (4.4) *satisfies*

$$\hat{\lambda} - \lambda_1 = \left(\frac{\delta - \lambda_1}{2}\right)\left(\frac{\Delta^2 - x_1^T x_1}{\beta + x_1^T x_1}\right) + \mathcal{O}((\lambda_1 - \lambda_*)^2). \tag{6.6}$$

Proof. The result is established using a Taylor expansion of the square root function near 1. The formulas of Algorithm 1 give

$$\begin{aligned} \hat{\lambda} &= \delta - \sqrt{\frac{\gamma^2}{\Delta^2 + \beta}} \\ &= \delta - (\delta - \lambda_1)\sqrt{\frac{\frac{\gamma^2}{(\delta - \lambda_1)^2}}{\frac{\gamma^2}{(\delta - \lambda_1)^2} + (\Delta^2 - x_1^T x_1)}} \\ &= \delta - (\delta - \lambda_1)\sqrt{\frac{1}{1 + \frac{\Delta^2 - x_1^T x_1}{\beta + x_1^T x_1}}} \\ &= \delta - (\delta - \lambda_1)\left[1 - \tfrac{1}{2}\left(\frac{\Delta^2 - x_1^T x_1}{\beta + x_1^T x_1}\right)\right] + \mathcal{O}((\lambda_1 - \lambda_*)^2). \end{aligned}$$

Simplifying this last term yields the desired formula (6.6). □

The updating formula for α will now be used to establish a result to relate $\lambda_+ - \lambda_*$ to $\lambda_1 - \lambda_*$.

LEMMA 6.4. *There is a neighborhood* $\mathcal{N}$ *of* λ_* *such that the iterate* λ_+ *produced at steps* 3 *and* 4 *of Algorithm* 1 *using formula* (4.5) *to compute* α_+ *based upon points* $\lambda_2, \lambda_1 \in (\mathcal{N})$ *will satisfy*

$$(\lambda_+ - \lambda_*) = (\lambda_1 - \lambda_*)\mu(\lambda_1, \lambda_2)\mathcal{O}(1) + \mathcal{O}((\lambda_1 - \lambda_*)^2), \tag{6.7}$$

where

$$\mu(\lambda_1, \lambda_2) \to 0 \quad \textit{as} \quad \lambda_1, \lambda_2 \to \lambda_*.$$

Proof. The proof begins with the formula

$$\alpha_+ = \hat{\lambda} + \eta + \beta(\delta - \hat{\lambda}) + \frac{\gamma^2}{\delta - \hat{\lambda}}.$$

Using the definition

$$\eta = -g^T x_1 - \beta(\delta - \lambda_1) - \frac{\gamma^2}{(\delta - \lambda_1)}$$

and the fact that

$$\beta(\delta - \hat{\lambda}) + \frac{\gamma^2}{\delta - \hat{\lambda}} = 2\beta(\delta - \hat{\lambda}) + (\delta - \hat{\lambda})\Delta^2,$$

substituting into the above formula gives

$$\alpha_+ = \hat{\lambda} - g^T x_1 + 2\beta(\lambda_1 - \hat{\lambda}) + (\delta - \hat{\lambda})\Delta^2 - (\delta - \lambda_1)x_1^T x_1.$$

Since $-g^T x_1 = \alpha - \lambda_1$, it follows after substitution and simplification that

$$\alpha_+ = \alpha + (\lambda_1 - \hat{\lambda})[2\beta + \Delta^2 - 1] + (\delta - \lambda_1)(\Delta^2 - x_1^T x_1). \tag{6.8}$$

Now utilize the relation $\alpha_+ = \lambda_+ - g^T x_+$ and $\alpha_* = \lambda_* - g^T x_*$ to see that

$$\begin{aligned} \alpha_+ - \alpha_* &= \lambda_+ - \lambda_* - g^T(x_+ - x_*) \\ &= \lambda_+ - \lambda_* + g^T(A_+^{-1} - A_*^{-1})g \\ &= \lambda_+ - \lambda_* + g^T A_+^{-1}(A_* - A_+)A_*^{-1} g \\ &= (\lambda_+ - \lambda_*)(1 + x_+^T x_*). \end{aligned} \tag{6.9}$$

Similarly,

$$\alpha - \alpha_* = (\lambda_1 - \lambda_*)(1 + x_1^T x_*).$$

Subtracting α_* from both sides of (6.8) above and substituting for $\alpha_+ - \alpha_*$ using (6.9) and Lemma 6.2 gives

$$\begin{aligned} &(\lambda_+ - \lambda_*)(1 + x_+^T x_*) \\ &\quad = (\lambda_1 - \lambda_*)(1 + x_1^T x_*) - (\delta - \lambda_1)(\Delta^2 - x_1^T x_1)\left[\frac{2\beta + \Delta^2 - 1}{2\beta + 2x_1^T x_1} - 1\right] + \mathcal{O}((\lambda_1 - \lambda_*)^2) \\ &\quad = (\lambda_1 - \lambda_*)(1 + x_1^T x_*) - (\Delta^2 - x_1^T x_1)(\delta - \lambda_1)\left[\frac{(\Delta^2 - x_1^T x_1) - (1 + x_1^T x_1)}{2\beta + 2x_1^T x_1}\right] \\ &\qquad\qquad + \mathcal{O}((\lambda_1 - \lambda_*)^2). \end{aligned}$$

Since

$$\begin{aligned} \Delta^2 - x_1^T x_1 &= x_*^T x_* - x_1^T x_1 \\ &= (\lambda_* - \lambda_1)[x_*^T A_1^{-1} x_* + x_1^T A_*^{-1} x_1] \end{aligned}$$

and since

$$\begin{aligned}2(\beta + x_1^T x_1) &= 2\frac{\gamma^2}{(\delta - \lambda_1)^2} \\ &= \frac{[x_2^T A_1^{-1} x_2 + x_1^T A_2^{-1} x_1][(\delta - \lambda_2)]^2}{\delta - \frac{1}{2}(\lambda_1 + \lambda_2)} \\ &= \mathcal{O}(1),\end{aligned}$$

it follows that

$$\begin{aligned}&(\lambda_+ - \lambda_*)(1 + x_+^T x_*) \\ &\quad = (\lambda_1 - \lambda_*)(1 + x_1^T x_*) \\ &\qquad - (\lambda_1 - \lambda_*)(1 + x_1^T x_1)\left[\frac{x_*^T A_1^{-1} x_* + x_1^T A_*^{-1} x_1}{x_2^T A_1^{-1} x_2 + x_1^T A_2^{-1} x_1}\right]\left[\frac{(\delta - \frac{1}{2}(\lambda_1 + \lambda_2))(\delta - \lambda_1)}{(\delta - \lambda_2)(\delta - \lambda_2)}\right] \\ &\qquad\qquad + \mathcal{O}((\lambda_1 - \lambda_*)^2) \\ &\quad = (\lambda_1 - \lambda_*)(x_1^T x_* - x_1^T x_1) \\ &\qquad - (\lambda_1 - \lambda_*)(1 + x_1^T x_1)\left(\left[\frac{x_*^T A_1^{-1} x_* + x_1^T A_*^{-1} x_1}{x_2^T A_1^{-1} x_2 + x_1^T A_2^{-1} x_1}\right]\left[\frac{(\delta - \frac{1}{2}(\lambda_1 + \lambda_2))(\delta - \lambda_1)}{(\delta - \lambda_2)(\delta - \lambda_2)}\right] - 1\right) \\ &\qquad\qquad + \mathcal{O}((\lambda_1 - \lambda_*)^2).\end{aligned} \tag{6.10}$$

Noting that

$$(\lambda_1 - \lambda_*)(x_1^T x_* - x_1^T x_1) = -(\lambda_1 - \lambda_*)^2 x_1^T A_*^{-1} x_1$$

and that

$$\frac{(1 + x_1^T x_1)}{(1 + x_+^T x_*)} = \mathcal{O}(1),$$

substituting into (6.10) establishes

$$(\lambda_+ - \lambda_*) = (\lambda_1 - \lambda_*)\mu(\lambda_1, \lambda_2)\mathcal{O}(1) + \mathcal{O}((\lambda_1 - \lambda_*)^2), \tag{6.11}$$

where

$$\mu(\lambda_1, \lambda_2) \equiv \left[\frac{x_*^T A_1^{-1} x_* + x_1^T A_*^{-1} x_1}{x_2^T A_1^{-1} x_2 + x_1^T A_2^{-1} x_1}\right]\left[\frac{(\delta - \frac{1}{2}(\lambda_1 + \lambda_2))(\delta - \lambda_1)}{(\delta - \lambda_2)(\delta - \lambda_2)}\right] - 1.$$

Since

$$\mu(\lambda_1, \lambda_2) \to 0 \quad \text{as} \quad \lambda_1, \lambda_2 \to \lambda_*,$$

the proof is complete. □

The previous discussion together with Lemmas 6.2–6.4 establishes the proof of Theorem 6.1.

Table 7.1
Average behavior for different tolerances.

	Trust mvecs	iters	Cg mvecs	Ratio
tol = .0001	59.3	4.2	44.4	1.34
tol = .000001	98.1	8.4	58.0	1.69
tol = .00000001	132.8	12.3	72.2	1.84

7. Computational results and conclusions.

In this final section, a limited set of computational results shall be presented to illustrate the viability of the approach presented here. These results are not meant to be exhaustive. They should be regarded as preliminary results intended to illustrate selected aspects of the behavior of this approach. They have been selected to illustrate behavior associated with different problem conditions. A comparison with the corresponding cost of solving the requisite linear systems via conjugate gradients to various accuracy levels is given. Behavior relative to widely different values of Δ are also demonstrated. Superlinear convergence is verified and hard case behavior is illustrated in the following examples.

The methods described in sections 3–5 were implemented in MATLAB, version 4.1. All experiments were carried out on a SUN SPARC station IPX. The floating point arithmetic is IEEE standard double precision with machine precision of $\epsilon_M \equiv 2^{-52} \approx 2.2204 \cdot 10^{-16}$. In all cases the IRLM described in section 3 was used to solve the eigenproblems. The number of Lanczos basis vectors was limited to nine. Six shifts (i.e., six matrix vector products) were applied on each implicit restart. The iteration was halted as soon as the smallest Ritz value had a Ritz estimate (3.2) below the specified tolerance.

The first experiment presents the performance on the problem (2.1) with the matrix $A = L - 5 * I$, where L is set to the standard 2-dimensional discrete Laplacian on the unit square based upon a 5-point stencil with equally-spaced mesh points. The shift of -5 was introduced to make the matrix indefinite. A sequence of 20 related problems were solved. The order of A was $n = 1024$ in all cases. The trust-region radius was fixed at $\Delta = 100$ for all of the problems. For each problem, a random vector g was constructed with entries uniformly distributed on (0, 1) and the problem was solved three times with a tolerance of 10^{-4}, 10^{-6}, and 10^{-8}. In Table 7.1 the average number of trust-region iterations and average number of matrix vector products $w \leftarrow Av$ per trust-region iteration are reported. In addition, the average number of matrix-vector products required to solve the system $(A - \lambda I)x = -g$ using the conjugate-gradient method is given. These tests indicate that a trust-region solution requires fewer than twice as many matrix-vector products on average than the number needed to solve a *single* linear system to the same accuracy using the conjugate-gradient method. The accuracy requirement of the eigenvalue solution computed by the IRLM at each step was relaxed and made proportional to the relative accuracy of the computed solution. More specifically, $\|B_\alpha q - q\lambda\| < \tau_1/1000$, where $\tau_1 = \min(10^{-6}, \left|\frac{\|x\|-\Delta}{\Delta}\right|)$. In addition to this, the inner IRLM iteration was initialized with the solution from the previous outer trust-region iteration.

The second experiment illustrates how the size of the trust-region parameter Δ may affect the solution process. In these problems the matrices were distributed in the form $A = UDU^T$ with D being a diagonal matrix with diagonal elements selected randomly from a uniform distribution on $(-.5, .5)$. The matrix $U = I - 2uu^T$ with the vector u and the vector g constructed with randomly distributed elements and then normalized to have unit length. The matrix A was of order $n = 1000$. The

TABLE 7.2
Behavior for different trust-region radii.

Δ	100	10	1	.1	.01	.001	.0001
Trust iters	13	8	4	4	4	4	4
Matvecs	579	240	36	36	36	36	36
CG-matvecs	1307	384	51	39	30	26	24
$\|g+(A-\lambda I)x\|$	(-4)	(-6)	(-12)	(-15)	(-15)	(-15)	(-15)
$\left\|\frac{\Delta-\|x\|}{\Delta}\right\|$	(-7)	(-7)	(-10)	(-9)	(-11)	(-13)	(-14)

trust-region radius Δ was varied by a factor of 10 through the values $100, 10, ..., .0001$ and each problem was solved to the level $\left|\frac{\Delta-\|x\|}{\Delta}\right| < 10^{-6}$. By way of comparison, the conjugate-gradient method was used to solve the same linear systems $(A-\lambda_j I)x = -g$ using the parameter λ_j provided by the eigensolution of B_{α_j} at the jth step of the trust-region iteration. Each system was solved by conjugate gradients to the same level of accuracy as the solution provided from the eigenvalue solution. The total number of matrix-vector products required by the eigenvalue method is to be compared to the number required by the conjugate-gradient method. These results are presented in Table 7.2.

The entries in parentheses in Table 7.2 represent powers of 10 (i.e., (-4) represents 10^{-4}). The row labeled *Trust iters* gives the number of iterations required in Algorithm 1. The row labeled *Matvecs* gives the number of matrix-vector products required to solve the resulting eigenvalue problems, and the row labeled *CG-matvecs* gives the number of matrix-vector products required by the conjugate-gradient iteration to solve the same linear systems. Note that for small trust-region radii there is not a significant difference in the required number of matrix-vector products but the conjugate-gradient method has a much easier time for smaller values of Δ than for larger values. This is because the matrix $A - \lambda I$ will have a very large value of λ and hence will act as though there are essentially two distinct eigenvalues when the value of Δ is small. Just the opposite situation occurs when the value of the trust-region radius gets larger. The eigenvalue problems do get more difficult to solve but the conjugate-gradient method has more trouble with these systems than the eigenvalue method. This phenomena is partially explained in [11]. When the spectrum is not clustered, it is often more difficult to solve the linear system by conjugate gradients than it is to find an extreme eigenvalue.

The next results verify the superlinear rate of convergence for the two point iteration. In this case the matrix A is again set to $A = L - 5I$ with L being the 2-dimensional discrete Laplacian on the unit square, but the order of A was $n = 256$ in this case. The trust-region radius was set at $\Delta = 10$ for all of the problems. Again, a random vector g was constructed with entries uniformly distributed on $(-.5, .5)$ and the problem was solved with a tolerance of 10^{-11}. In Table 7.3 the progressive decrease in the magnitude of $\frac{\Delta-\|x\|}{\Delta}$ is charted as the iteration proceeds. The required number of iterations was 6 and it took 144 matrix-vector products to solve the associated eigenvalue problems. Each eigenproblem was solved to the accuracy level $\|B_\alpha v - v\lambda\| \leq 10^{-9}$.

To study the behavior of the algorithm in the hard case, the same matrix $A = L - 5I$ of order 256 was used. In order to generate the hard case the vector g was randomly generated as before and then the operation $g \leftarrow g - q(q^T g)$ was performed to orthogonalize g to the eigenvector q corresponding to the smallest eigenvalue of

TABLE 7.3
Verification of superlinear convergence.

Iter	$\frac{\Delta - \|x\|}{\Delta}$
1	0.8730
2	-0.1028
3	0.0063
4	7.1389e-05
5	-4.8522e-08
6	1.2491e-12

TABLE 7.4
The hard case.

Iter	$\left\| \frac{z^T(A-\lambda I)z}{(g^Tp+\lambda\Delta^2)} \right\|$
1	0.2916
2	0.1448
3	0.0631
4	0.0221
5	0.0049
6	3.8942e-04
7	2.9174e-06
8	8.5908e-09

A. Then a "noise" vector of norm 10^{-8} was added to g. In this test the trust-region radius was $\Delta = 100$. A number of different problems were solved and the following behavior of one of the problems was typical of all of them. In every problem the hard case was detected on the second regular iteration of Algorithm 1 and then the iteration of Algorithm 2 was entered. Table 7.4 displays the ratio

$$\left| \frac{z^T(A-\lambda I)z}{(g^Tp+\lambda\Delta^2)} \right|,$$

and the iteration was halted when this ratio was less than $\sigma = .000001$, where σ is the tolerance introduced in Lemma 5.2 (see step 4 of Algorithm 2).

The final solution was on the trust-region boundary to within working precision and it required a total of 11 eigenvalue problems which required 291 matrix-vector products. One of these was done between the transition from Algorithm 1 to Algorithm 2 in order to assure that a lower bound on α had been obtained. This step is not reported in Table 7.4. The behavior of this iteration seemed to be more sensitive to the level of accuracy required by the eigensolution than in the standard case. A rational approximation was tried instead of the linear interpolation and this performed poorly. However, more testing is needed and perhaps a modification of the scheme for the hard case will lead to improvements. No testing was done on large matrices since it was desirable to have complete control over which eigenvectors the vector g would be orthogonalized against. Moreover, no testing was done with higher dimensional eigenspaces corresponding to the smallest eigenvalue. Finally, special consideration may be called for in the case of least squares problems arising from the discretization of ill-posed continuous problems. These problems will be of the form $\min\{\|Mx - b\| \ : \ \|x\| \leq \Delta\}$, and for ill-posed problems, the matrix $A = M^TM$ will be singular or nearly singular and the vector $g = M^Tb$ will be orthogonal or nearly orthogonal to the corresponding null space of A. The method described by Golub and von Matt [6] may be better suited to this situation, and this comparison should be made.

Although a direct comparison to the secant method has not been made here, the results that have been compiled with respect to the performance of the conjugate-gradient iteration may be used to draw some conclusions. Two possibilities for a secant iteration come to mind. The first would be to apply the secant method directly to the problem of adjusting λ to obtain

$$\frac{1}{\Delta} - \frac{1}{\|x_\lambda\|} = 0$$

using the conjugate-gradient method to solve the resulting linear systems of the form $(A - \lambda I)x = -g$. An immediate problem with this approach is to discover the range of λ for which $(A - \lambda I)$ is positive definite. Moreover, the systems that would have to be solved would be as computationally demanding for the conjugate-gradient iteration as the ones arising within the iteration presented here. The computational results indicate this approach would be inferior to the eigenvalue approach for modest to large trust-region radii and roughly comparable for small radii.

A second possibility would be to use the eigenvalue formulation (2.2) to obtain points $x_{\lambda(\alpha)}$ but to apply the secant method to the problem

$$\frac{1}{\Delta} - \frac{1}{\|x_{\lambda(\alpha)}\|} = 0$$

in order to adjust the parameter α instead of using the specialized iteration derived in section 4. This method was coded and computational tests showed it to be inferior to the method presented here. It took many more iterations in general than the specialized iteration based upon rational interpolation.

These results indicate promise for this approach to solving the large-scale trust-region subproblem. The examples given here were solved to tolerances which are unlikely to arise in most applications. This was done to get some indication of the asymptotic behavior and to verify the convergence results presented in section 6. While these preliminary tests are very encouraging, further experience with testing and with actual application will be necessary.

REFERENCES

[1] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, *A trust region algorithm for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 24 (1987), pp. 1152–1170.

[2] M. Celis, J. E. Dennis, and R. A. Tapia, *A trust region strategy for nonlinear equality constrained optimization*, in Numerical Optimization 1984, SIAM, Philadelphia, PA, 1985, pp. 71–82.

[3] A. K. Cline, C. B. Moler, G. W. Stewart, and J. H. Wilkinson, *An estimate for the condition number of a matrix*, SIAM J. Numer. Anal., 16 (1979), pp. 368–375.

[4] M. El-Alem, *A global convergence theory for the Celis–Dennis–Tapia trust region algorithm for constrained optimization*, SIAM J. Numer. Anal., 28 (1991), pp. 266–290.

[5] W. Gander, *Least squares with a quadratic constraint*, Numer. Math., 36 (1981), pp. 291–307.

[6] G. Golub and U. von Matt, *Quadratically constrained least squares and quadratic problems*, Numer. Math., 59 (1991), pp. 561–580.

[7] R. B. Lehoucq, D. C. Sorensen, P. Vu, and C. Yang, *ARPACK: An implementation of the implicitly re-started Arnoldi iteration that computes some of the eigenvalues and eigenvectors of a large sparse matrix*, 1995. Available from ftp.caam.rice.edu under the directory pub/software/ARPACK.

[8] W. Menke, *Geophysical Data Analysis: Discrete Inverse Theory*, Academic Press, San Diego, CA 1989.

[9] J. Moré, *Recent developments in algorithms and software for trust region methods*, in Mathematical Programming: The State of the Art, A. Bachem, M. Grotschel, and B. Korte, eds., Springer-Verlag, Berlin, New York, 1983, pp. 258–287.

[10] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[11] C. C. Paige, B. N. Parlett, and H. A. V. der Vorst, *Approximate solutions and eigenvalue bounds from Krylov subspaces*, Numer. Linear Algebra Appl., 2 (1995), pp. 115–134.

[12] B. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[13] M. J. D. Powell and Y. Yuan, *A trust region algorithm for equality constrained optimization*, Math. Programming, 49 (1991), pp. 189–211.

[14] D. C. Sorensen, *Newton's method with a model trust region modification*, SIAM J. Numer. Anal., 19 (1982), pp. 409–426.

[15] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.

[16] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[17] A. Tarantola, *Inverse Problem Theory*, Elsevier, Amsterdam, 1987.

[18] P. L. Toint, *Towards an efficient sparsity exploiting newton method for minimization*, in Sparse Matrices and Their Uses, I. S. Duff, ed., Academic Press, New York, 1981, pp. 7–87.

SIAM J. OPTIM.
Vol. 7, No. 1, pp. 162–176, February 1997

NEWTON METHODS FOR LARGE-SCALE LINEAR INEQUALITY-CONSTRAINED MINIMIZATION*

ANDERS FORSGREN[†] AND WALTER MURRAY[‡]

Abstract. Newton methods of the linesearch type for large-scale minimization subject to linear inequality constraints are discussed. The purpose of the paper is twofold: (i) to give an active–set-type method with the ability to delete multiple constraints simultaneously and (ii) to give a relatively short general convergence proof for such a method. It is also discussed how multiple constraints can be added simultaneously. The approach is an extension of a previous work by the same authors for equality-constrained problems. It is shown how the search directions can be computed without the need to compute the reduced Hessian of the objective function. The convergence analysis states that every limit point of a sequence of iterates satisfies the *second-order* necessary optimality conditions.

Key words. linear inequality-constrained minimization, negative curvature, modified Newton method, symmetric indefinite factorization, large-scale minimization, linesearch method

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

PII. S1052623494279122

1. Introduction.

We consider a method for finding a local minimizer of the problem

$$\begin{array}{lll} (1.1) & \underset{x\in I\!\!R^n}{\text{minimize}} & f(x) \\ & \text{subject to} & Ax \geq b, \end{array}$$

where A is an $m \times n$ matrix and $f \in C^2$. We are interested in the case when n and possibly m are large and when second derivatives of f are available. The method is a Newton method of the linesearch type using an active-set strategy to identify the constraints that are active at the solution, where the active set at each iteration may change significantly. No assumptions are made about the number of constraints active at the solution or in the problem. In the approach advocated, it is not necessary to make any initial transformation of the problem such as transforming it into canonical form. The method proposed builds on a method we proposed recently for the *equality*-constrained problem [11] and requires only a single matrix factorization per iteration.

Linearly constrained optimization has been studied quite extensively over the years; see, e.g., Gill, Murray, and Wright [17, Chapter 5] and Fletcher [10, Chapter 11]. As mentioned above, our interest is in linesearch methods of the active-set type, i.e., methods that solve a sequence of equality-constrained subproblems. Methods of this type, designed to give limit points that satisfy the *first-order* optimality conditions, have been given by, e.g., Rosen [28], Goldfarb [18], Ritter [25, 26, 27], and Byrd and Shultz [6]. Similarly, linesearch methods designed to give limit points that satisfy the

* Received by the editors December 21, 1994; accepted for publication (in revised form) October 18, 1995.

http://www.siam.org/journals/siopt/7-1/27912.html

[†] Optimization and Systems Theory, Department of Mathematics, Royal Institute of Technology, S-100 44 Stockholm, Sweden (andersf@math.kth.se). This author was partially supported by the Royal Swedish Academy of Sciences (Magnuson's fund, KVA) and the Swedish Natural Science Research Council (NFR).

[‡] Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA 94305-4022 (walter@sol-walter.stanford.edu). This author was supported by Department of Energy contract DE-FG03-92ER25117, National Science Foundation grant DDM-9204208, and Office of Naval Research grant N00014-90-J-1242.

second-order necessary optimality conditions have been given by, e.g., McCormick [20] and Gill and Murray [13]. Methods for large-scale linearly constrained problems are given by, e.g., Buckley [1] and Murtagh and Saunders [23]. The motivation for our work is to give a method for large-scale problems together with a concise and comprehensive convergence analysis. The method proposed here gives limit points that satisfy the second-order necessary optimality conditions and it is based on a single matrix factorization per iteration. Although only linesearch methods are considered in this paper, *trust-region* methods with similar convergence properties have been proposed; see, e.g., Gay [12].

2. Notation and assumptions.

The method proposed generates a sequence $\{x_k\}_{k=0}^{\infty}$ of iterates of the form

$$x_{k+1} = x_k + \alpha_k p_k,$$

where p_k is a search direction and α_k is determined by a linesearch along p_k. It is assumed that $f_k \equiv f(x_k)$, the gradient $g_k \equiv \nabla f(x_k)$, and the Hessian $H_k \equiv \nabla^2 f(x_k)$ can be evaluated. The definition of p_k is given in section 3 and the conditions on α_k are discussed in section 3.5. We denote by a_i^T the ith row of A and by b_i the ith component of b. At a point x_k, a constraint $a_i^T x \geq b_i$ is said to be *active* if $a_i^T x_k = b_i$, *inactive* if $a_i^T x_k > b_i$, and *violated* if $a_i^T x_k < b_i$. We denote by A_k a matrix comprising a subset of the rows of A that correspond to constraints active at x_k. Similarly, b_k is the vector of the corresponding elements of b. We denote by $\mathcal{W}_k \subseteq \{1, 2, \ldots, m\}$ the indices of the rows of A in A_k and refer to $\mathcal{W}_k$ as the *working set* at iteration k. The notation $\mathcal{W}_{k+1} \backslash \mathcal{W}_k$ is used for the set of indices that belong to $\mathcal{W}_{k+1}$ but not to $\mathcal{W}_k$. (Note that $\mathcal{W}_{k+1} \backslash \mathcal{W}_k$ is defined also when $\mathcal{W}_k \not\subseteq \mathcal{W}_{k+1}$.) The matrix Z_k denotes an orthonormal matrix whose columns form a basis for the null space of A_k. Note that Z_k need not be known; our use of this matrix is for theoretical purposes only. We shall assume that A_0 has full row rank. Then, the rules we give in section 3.6 for updating A_k ensure that A_k has full rank for all k. In section 5.1 it is shown how A_0 may be obtained without making any assumptions about A, and in section 5.6 it is shown how A_k may be updated while maintaining the full row rank. For a symmetric matrix M, we use the notation $\lambda_{\min}(M) \geq 0$ for M positive semidefinite and $\lambda_{\min}(M) > 0$ for M positive definite but this is just for notational purposes, and the eigenvalues are not computed. For a sequence $I \subseteq \{0, 1, \ldots\}$, the abbreviated notation $\lim_{k \in I}$ is used for $\lim_{k \to \infty, k \in I}$.

Throughout, the following assumptions are made:

A1. The objective function f is twice continuously differentiable.

A2. The initial feasible point x_0 is known, and the level set $\{x : Ax \geq b, f(x) \leq f(x_0)\}$ is compact.

A3. The constraint matrix associated with the active constraints has full row rank at all points that satisfy the *second-order* necessary optimality conditions if these constraints are regarded as equalities. Formally, let $\bar{x}$ denote a feasible point of (1.1), let A_A denote the matrix associated with the active constraints at $\bar{x}$, and let Z_A denote a matrix whose columns form an orthonormal basis for the null space of A_A. If it holds that

$$Z_A^T \nabla f(\bar{x}) = 0 \quad \text{and} \quad \lambda_{\min}(Z_A^T \nabla^2 f(\bar{x}) Z_A) \geq 0,$$

then A_A has full row rank.

Assumption A3 states that the problem does not have *primal degenerate* second-order constrained stationary points (dual degeneracy may occur). Any algorithm for general problems we are familiar with, for which primal nondegeneracy does not need to be assumed, requires an iteration that in itself has a subiteration. Our purpose here is to devise algorithms that do not require such subiterations since our primary concern is to solve large problems. Nonetheless, degeneracy (or near degeneracy) is possible and needs to be dealt with in any practical implementation. In practice degeneracy may be dealt with by techniques that allow the standard iteration to be used; see, e.g., Gill et al. [14]. Such a technique is used within the MINOS code, see Murtagh and Saunders [24], which has been used to solve thousands of practical problems. The consequence of using this approach to degeneracy is that the solution obtained may be infeasible. However, the degree of infeasibility may be set at a level similar to that which arises due to finite precision. Indeed, even if degeneracy was not present such techniques are necessary in an endeavor to make the matrix of active constraints well conditioned. Discussions on theoretical aspects of degeneracy are given in Burke and Moré [3, 4], Burke [2], and Burke, Moré, and Toraldo [5].

3. Definition of the algorithm.

The search direction p_k is a sum of three directions. More specifically,

$$p_k = s_k + d_k + q_k,$$

where a nonzero s_k is a descent direction of bounded norm in the null space of A_k, a nonzero d_k is a direction of negative curvature with bounded norm in the null space of A_k, and a nonzero q_k is a descent direction of bounded norm such that $A_k q_k \geq 0$ and $a_j^T q_k > 0$ for some $j \in \mathcal{W}_k$. At each iteration, a set of Lagrange multiplier estimates π_k, associated with A_k, is required. In this section, the required properties of s_k, d_k, π_k, and q_k are given, and in section 5 an appropriate way of computing the directions for large-scale problems is discussed.

3.1. Properties of $\boldsymbol{s_k}$.

A nonzero s_k has to have bounded norm and be a descent direction in the null space of A_k, i.e., satisfy $g_k^T s_k < 0$ and $A_k s_k = 0$. We also require that s_k be a *sufficient* descent direction in the following sense:

$$\lim_{k \in I} g_k^T s_k = 0 \quad \Rightarrow \quad \lim_{k \in I} Z_k^T g_k = 0 \quad \text{and} \quad \lim_{k \in I} s_k = 0, \tag{3.1}$$

where I is any subsequence.

3.2. Properties of $\boldsymbol{d_k}$.

We require a nonzero d_k to be a nonascent direction of negative curvature in the null space of the A_k, i.e., $g_k^T d_k \leq 0$, $d_k^T H_k d_k < 0$, and $A_k d_k = 0$. Furthermore, the norm of d_k has to be bounded and the curvature has to be *sufficient* in the sense that

$$\lim_{k \in I} d_k^T H_k d_k = 0 \quad \Rightarrow \quad \liminf_{k \in I} \lambda_{\min}(Z_k^T H_k Z_k) \geq 0 \quad \text{and} \quad \lim_{k \in I} d_k = 0, \tag{3.2}$$

where I is any subsequence.

3.3. Properties of $\boldsymbol{\pi_k}$.

At each iteration, a vector of Lagrange multiplier estimates, π_k, is required. The vector π_k must satisfy

$$\lim_{k \in I} \|Z_k^T g_k\| = 0 \quad \Rightarrow \quad \lim_{k \in I} \|g_k - A_k^T \pi_k\| = 0, \tag{3.3}$$

where I is any subsequence. We define $\pi_{\min,k} = \min_i (\pi_k)_i$ and use this notation throughout.

3.4. Properties of q_k. If $\pi_{\min,k} \geq 0$ or $\mathcal{W}_k \not\subseteq \mathcal{W}_{k-1}$, we set $q_k = 0$. This is to say that we take at least one step towards minimality for a given A_k before considering deleting constraints. When $q_k \neq 0$ we require it to be a descent direction that moves off at least one constraint in the working set and remains feasible with respect to the others, i.e., $g_k^T q_k < 0$ and $0 \neq A_k q_k \geq 0$. Furthermore, the norm of q_k has to be bounded and it is also required that the q_k's are such that

$$\lim_{k \in I} g_k^T q_k = 0 \Rightarrow \liminf_{k \in I} \pi_{\min,k} \geq 0 \quad \text{and} \quad \lim_{k \in I} q_k = 0, \tag{3.4a}$$

$$a_i^T q_k > 0 \Rightarrow (\pi_k)_i \leq \nu \pi_{\min,k} \quad \text{for} \quad k \in I,\ i \in \mathcal{W}_k, \tag{3.4b}$$

where I is any subsequence such that $\mathcal{W}_k \subseteq \mathcal{W}_{k-1}$ for all $k \in I$ and ν is a preassigned tolerance, $(0 < \nu \leq 1)$.

3.5. Definition of the iterates. We follow Moré and Sorensen [21] and Forsgren and Murray [11] in the linesearch and adapt it to cope with inequality constraints. For the sake of completion, the linesearch is reviewed here, and the properties that are subsequently required for the linear inequality-constrained case are given in Lemmas 4.1, 4.2, and 4.3 below.

Iteration k takes the following form. The search direction is obtained as $p_k = s_k + d_k + q_k$, where s_k, d_k, and q_k satisfy the conditions of sections 3.1–3.4. Define $\phi_k(\alpha) = f(x_k + \alpha p_k)$. Sections 3.1–3.4 give $p_k = 0$ if and only if $\phi_k'(0) = 0$ and $\phi_k''(0) \geq 0$. The linesearch is designed to give $\lim_{k\to\infty} \phi_k'(0) = 0$ and $\liminf_{k\to\infty} \phi_k''(0) \geq 0$. An upper bound on the steplength is computed as

$$\bar{\alpha}_k = \min\left\{\alpha_{\max}, \min_{i: a_i^T p_k < 0} \frac{a_i^T x_k - b_i}{-a_i^T p_k}\right\},$$

where $\alpha_{\max}$, $(\alpha_{\max} \geq 1)$ is a fixed upper bound on the maximum steplength. If $\bar{\alpha}_k = 0$, then $\alpha_k = 0$. Otherwise, the steplength α_k is determined such that $\alpha_k \in (0, \bar{\alpha}_k]$ satisfies

$$\phi_k(\alpha_k) \leq \phi_k(0) + \mu(\phi_k'(0)\alpha_k + \tfrac{1}{2}\min\{\phi_k''(0), 0\}\alpha_k^2) \tag{3.5}$$

and at least one of

$$|\phi_k'(\alpha_k)| \leq \eta |\phi_k'(0) + \min\{\phi_k''(0), 0\}\alpha_k| \quad \text{or} \tag{3.6a}$$

$$\alpha_k = \bar{\alpha}_k, \tag{3.6b}$$

where $0 < \mu < 0.5$ and $\mu \leq \eta < 1$. Finally, $x_{k+1} = x_k + \alpha_k p_k$. The conditions of sections 3.1–3.4 give $\phi_k'(0) \leq 0$ for all k, and $\phi_k'(0) = 0$ if and only if $p_k = d_k$. It follows from Moré and Sorensen [21, Lemma 5.2] that α_k is well defined.

We refer to a step α_k as *restricted* if

$$\alpha_k = \min_{i: a_i^T p_k < 0} \frac{a_i^T x_k - b_i}{-a_i^T p_k},$$

i.e., a constraint is encountered in the linesearch at iteration k. Otherwise, the step is referred to as *unrestricted*. Hence, a restricted step always satisfies (3.6b) whereas an unrestricted step satisfies at least one of $\alpha_k = \alpha_{\max}$ or (3.6a).

3.6. Properties of $\boldsymbol{A_k}$. The initial working-set matrix A_0 is required to have full row rank and contain constraints active at x_0. To give the rule for updating $\mathcal{W}_k$, define

$$\mathcal{W}_k^0 = \{i \in \mathcal{W}_k : a_i^T p_k = 0\}.$$

Let $\mathcal{P}_k^a$ denote the index set of constraints that are encountered in the linesearch at iteration k, i.e.,

$$\mathcal{P}_k^a = \{i \notin \mathcal{W}_k : a_i^T p_k < 0,\ a_i^T x_{k+1} = b_i\}.$$

Note that either of $\mathcal{W}_k^0$ and $\mathcal{P}_k^a$ may be the empty set. We then define $\mathcal{W}_{k+1} = \mathcal{W}_k^0 \cup \mathcal{W}_k^a$, where $\mathcal{W}_k^a \subseteq \mathcal{P}_k^a$ and the associated A_{k+1} are required to satisfy

$$\mathcal{P}_k^a \neq \emptyset \Rightarrow \mathcal{W}_k^a \neq \emptyset \quad \text{and} \tag{3.7a}$$

$$A_{k+1} \text{ has full row rank.} \tag{3.7b}$$

The implication of (3.7a) is that if new constraints are encountered in the linesearch, at least one of them has to be added. If A_k has full row rank, (3.7b) will trivially hold if $\mathcal{W}_k^a = \emptyset$. Otherwise, care has to be taken to ensure that A_{k+1} has full row rank. This is further discussed in section 5.6.

Note that an implication of the above conditions is that a step α_k is restricted if and only if $\mathcal{W}_{k+1} \backslash \mathcal{W}_k \neq \emptyset$.

4. Convergence results for linear inequality constraints.

Lemmas 4.1, 4.2, and 4.3 below review results from unconstrained optimization originally proposed by Moré and Sorensen [21]. These give results for unrestricted steps. The remainder of this section then establishes the convergence results for linear inequality-constrained problems.

The following lemma gives some properties of the iterates for a sequence generated by the above linesearch conditions.

Lemma 4.1. *Given assumptions* A1–A3*, assume that a sequence* $\{x_k\}_{k=0}^{\infty}$ *is generated as outlined in section* 3*. Then*

(i) $\lim_{k\to\infty} \alpha_k \phi_k'(0) = 0$;

(ii) $\lim_{k\to\infty} \alpha_k^2 \min\{\phi_k''(0), 0\} = 0$;

(iii) $\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0$.

Proof. Rearrangement of (3.5) gives

$$\phi_k(0) - \phi_k(\alpha_k) \geq -\mu(\phi_k'(0)\alpha_k + \tfrac{1}{2}\min\{\phi_k''(0), 0\}\alpha_k^2).$$

Since $\mu > 0$, $\phi_k'(0) \leq 0$, and the objective function is bounded from below on the feasible region, (i) and (ii) follow.

To show (iii), we write $x_{k+1} - x_k = \alpha_k p_k$ and show that $\lim_{k\to\infty} \|\alpha_k p_k\| = 0$. Since α_k and $\|p_k\|$ are bounded, if $\lim_{k\to\infty} \|\alpha_k p_k\| \neq 0$, there must exist a subsequence I and $\epsilon_1 > 0$ and $\epsilon_2 > 0$ such that $\alpha_k \geq \epsilon_1$ and $\|p_k\| \geq \epsilon_2$ for $k \in I$. From the existence of ϵ_1, (i) implies $\lim_{k\in I} \phi_k'(0) = 0$ and (ii) implies $\liminf_{k\in I} \phi_k''(0) \geq 0$. Since $\phi_k'(0) = g_k^T p_k = g_k^T(s_k + d_k + q_k)$ and it holds that $g_k^T s_k \leq 0$, $g_k^T d_k \leq 0$, and $g_k^T q_k \leq 0$, (3.1) implies $\lim_{k\in I} s_k = 0$ and (3.4) implies $\lim_{k\in I} q_k = 0$. Hence, since $\phi_k''(0) = p_k^T H_k p_k$ and $\lim_{k\in I} \|p_k - d_k\| = 0$, (3.2) implies $\lim_{k\in I} d_k = 0$. Thus, $\lim_{k\in I} \|p_k\| = 0$. This contradicts the existence of ϵ_2, thus establishing (iii). □

The following lemma relates α_k to $\phi_k'(0)$ for an unrestricted step. The implication is that α_k is bounded away from zero if $\phi_k'(0)$ is bounded away from zero.

LEMMA 4.2. *Given assumptions* A1–A3*, assume that a sequence* $\{x_k\}_{k=0}^{\infty}$ *is generated as outlined in section* 3*. If, at iteration* k*, an unrestricted step is taken, then either* $\alpha_k = \alpha_{\max}$ *or there exists a* θ_k*,* $(0 < \theta_k < \alpha_k)$ *such that*

$$\alpha_k(\phi_k''(\theta_k) + \eta \max\{-\phi_k''(0), 0\}) \geq -(1-\eta)\phi_k'(0). \tag{4.1}$$

Proof. Since $\phi_k'(0) \leq 0$, it follows from (3.6) that if α_k is unrestricted and $\alpha_k < \alpha_{\max}$, it satisfies

$$-\phi_k'(\alpha_k) \leq -\eta\phi_k'(0) + \eta \max\{-\phi_k''(0), 0\}\alpha_k. \tag{4.2}$$

Further, since ϕ_k' is a continuously differentiable univariate function, the mean-value theorem ensures the existence of a $\theta_k \in (0, \alpha_k)$ such that

$$\phi_k'(\alpha_k) = \phi_k'(0) + \alpha_k\phi_k''(\theta_k). \tag{4.3}$$

A combination of (4.2) and (4.3) now gives (4.1), as required. □

Finally, the following lemma gives some properties of subsequences of unrestricted iterates for a sequence generated by the above linesearch conditions.

LEMMA 4.3. *Given assumptions* A1–A3*, assume that a sequence* $\{x_k\}_{k=0}^{\infty}$ *is generated as outlined in section* 3*. Let* I *denote a subsequence of iterations where unrestricted steps are taken*; *then*

(i) $\lim_{k\in I} \phi_k'(0) = 0$;
(ii) $\liminf_{k\in I} \phi_k''(0) \geq 0$;
(iii) $\lim_{k\in I} Z_k^T g_k = 0$ *and* $\liminf_{k\in I} \lambda_{\min}(Z_k^T H_k Z_k) \geq 0$.

Proof. To show (i), assume by contradiction there is a subsequence $I' \subseteq I$ such that $\phi_k'(0) \leq -\epsilon_1 < 0$ for $k \in I'$. Lemma 4.2 in conjunction with assumptions A1 and A2 then implies that $\limsup_{k\in I'} \alpha_k \neq 0$, contradicting Lemma 4.1. Hence, the assumed existence of I' is false, and we conclude that (i) holds.

Similarly, to show (ii), assume by contradiction that there is a subsequence $I'' \subseteq I$ such that $\phi_k''(0) \leq -\epsilon_2 < 0$ for $k \in I''$. Since $\alpha_k > 0$ and $\phi_k'(0) \leq 0$, Lemma 4.2 implies that for $k \in I''$ there exists $\theta_k \in (0, \alpha_k)$ such that

$$\phi_k''(\theta_k) - \eta\phi_k''(0) \geq 0. \tag{4.4}$$

Lemma 4.1 gives $\lim_{k\in I''} \alpha_k = 0$, and thus (4.4) cannot hold for k sufficiently large. Consequently, the assumed existence of I'' is false, and (ii) holds.

Finally, we show that (i) and (ii) imply (iii). Since $\phi_k'(0) = g_k^T p_k = g_k^T(s_k + d_k + q_k)$ and it holds that $g_k^T s_k \leq 0$, $g_k^T d_k \leq 0$, and $g_k^T q_k \leq 0$, (i) and (3.1) imply $\lim_{k\in I} Z_k^T g_k = 0$ and $\lim_{k\in I} s_k = 0$. Further, (i) and (3.4a) imply $\lim_{k\in I} q_k = 0$. Hence, since $\phi_k''(0) = p_k^T H_k p_k$ and $\lim_{k\in I} \|p_k - d_k\| = 0$, (ii) and (3.2) imply $\liminf_{k\in I} \lambda_{\min}(Z_k^T H_k Z_k) \geq 0$ and, thus, (iii) holds. □

We now extend these results to the case of linear inequality constraints. The first lemma shows that if there exists a subsequence of iterates at which a constraint is deleted with the smallest multiplier negative and bounded away from zero and for which no constraints were deleted at the previous iteration, then eventually a constraint will be added.

LEMMA 4.4. *Given assumptions* A1–A3*, assume that a sequence* $\{x_k\}_{k=0}^{\infty}$ *is generated as outlined in section* 3*. If there is a subsequence* I *and an* $\epsilon > 0$ *such that* $q_{k-1} = 0$*,* $q_k \neq 0$*, and* $\pi_{\min,k} < -\epsilon$ *for* $k \in I$*, then there is an integer* K *such that* $\mathcal{W}_{k+1} \backslash \mathcal{W}_k \neq \emptyset$ *for all* $k \in I$ *and* $k \geq K$*.*

Proof. Suppose that there is a subsequence I and an $\epsilon > 0$ such that $q_{k-1} = 0$, $q_k \neq 0$, and $\pi_{\min,k} < -\epsilon$ for $k \in I$. Now assume that there is a subsequence $I' \subseteq I$ such that an unrestricted step is taken for $k \in I'$. Lemma 4.3 implies that $\lim_{k \in I'} \phi'_k(0) = 0$. On the other hand, (3.4a) ensures the existence of a subsequence $I'' \subseteq I'$ and a positive constant ϵ_2 such that $g_k^T q_k \leq -\epsilon_2$ for all $k \in I''$. However, since $g_k^T s_k \leq 0$ and $g_k^T d_k \leq 0$, this implies that $\phi'_k(0) \leq -\epsilon_2$ for all $k \in I'$, which is a contradiction. Hence, the assumed existence of the subsequence I' is false, and there must exist a K such that for $k \in I$ and $k \geq K$ a restricted step is taken, i.e., $\mathcal{W}_{k+1} \backslash \mathcal{W}_k \neq \emptyset$ for all $k \in I$ and $k \geq K$. □

Assumption A3 can now be used to show that for a subsequence of iterates where constraints are deleted, but no constraints were deleted at the previous iteration, the smallest multiplier is nonnegative in the limit.

LEMMA 4.5. *Given assumptions* A1–A3*, assume that a sequence* $\{x_k\}_{k=0}^{\infty}$ *is generated as outlined in section* 3*. If there is a subsequence I such that* $q_{k-1} = 0$ *and* $q_k \neq 0$ *for $k \in I$, then* $\liminf_{k \in I} \pi_{\min,k} \geq 0$*.*

Proof. Assume that there exists a subsequence I and an $\epsilon > 0$ such that $q_{k-1} = 0$, $q_k \neq 0$, and $\pi_{\min,k} < -\epsilon$ for $k \in I$. For each $k \in I$, let l_k denote the following iteration with least index such that $\mathcal{W}_{l_k} = \mathcal{W}_{l_k - 1}$; i.e., an unrestricted step is taken at iteration $l_k - 1$ and $q_{l_k - 1} = 0$. Lemma 4.4 implies that there is an integer K such that $\mathcal{W}_{k+1} \backslash \mathcal{W}_k \neq \emptyset$ for all $k \in I$ and $k \geq K$. The properties of q_k from section 3.4 imply that $q_{k+1} = 0$ for $k \in I$, $k \geq K$. Consequently, for $k \geq K$, l_k is the iteration with least index following k where no constraint is added in the linesearch. Since there can be at most $\min\{m, n\}$ consecutive iterations where a constraint is added, it follows from (iii) of Lemma 4.1 that $\lim_{k \in I} \|x_k - x_{l_k}\| = 0$. Consequently, there must exist a point $\bar{x}$, which is a common limit point to $\{x_k\}_{k \in I}$ and $\{x_{l_k}\}_{k \in I}$. By taking appropriate subsequences, there exists a subsequence $I' \subseteq I$ such that $\lim_{k \in I'} x_k = \bar{x}$ and $\lim_{k \in I'} x_{l_k} = \bar{x}$. Again, by taking appropriate subsequences, there must exist a subsequence $I'' \subseteq I'$ such that $\mathcal{W}_k$ is identical for every $k \in I''$ and $\mathcal{W}_{l_k}$ is identical for every $l_k \in J$, where J denotes the subsequence $\{l_k\}_{k \in I''}$. Define $\mathcal{W}^I \equiv \mathcal{W}_k$ for any $k \in I''$ and $\mathcal{W}^J \equiv \mathcal{W}_{l_k}$ for any $l_k \in J$.

Since all constraints corresponding to $\mathcal{W}^I$ are active at $\bar{x}$ and an infinite number of unrestricted steps are taken where the working set is constant, it follows from assumptions A1 and A2 in conjunction with (iii) of Lemma 4.1 and (iii) of Lemma 4.3 that $\lim_{k \in I''} Z_I^T g_k = 0$ and $\liminf_{k \in I''} \lambda_{\min}(Z_I^T H_k Z_I) \geq 0$, where Z_I denotes a matrix whose columns form an orthonormal basis for the null space of A_I, the constraint matrix associated with $\mathcal{W}^I$. Consequently, (3.3) and the full row rank of A_I imply that $\lim_{k \in I''} \pi_k = \pi^I$, where π^I satisfies

$$\nabla f(\bar{x}) = A_I^T \pi^I = \sum_{i \in \mathcal{W}^I} a_i \pi_i^I. \tag{4.5}$$

By a similar reasoning and notation for Z_J and A_J we have $\lim_{k \in I''} Z_J^T g_{l_k} = 0$, $\liminf_{k \in I''} \lambda_{\min}(Z_J^T H_{l_k} Z_J) \geq 0$, and $\lim_{k \in I''} \pi_{l_k} = \pi^J$, where π^J satisfies

$$\nabla f(\bar{x}) = A_J^T \pi^J = \sum_{i \in \mathcal{W}^J} a_i \pi_i^J. \tag{4.6}$$

Combining (4.5) and (4.6), we obtain

$$\sum_{i \in \mathcal{W}^I \backslash \mathcal{W}^J} a_i \pi_i^I + \sum_{i \in \mathcal{W}^I \cap \mathcal{W}^J} a_i \left(\pi_i^I - \pi_i^J \right) - \sum_{i \in \mathcal{W}^J \backslash \mathcal{W}^I} a_i \pi_i^J = 0. \tag{4.7}$$

By assumption A3, the vectors a_i, $i \in \mathcal{W}^I \cup \mathcal{W}^J$ are linearly independent. Hence, it follows from (4.7) that

$$\pi_i^I = 0 \quad \text{for } i \in \mathcal{W}^I \backslash \mathcal{W}^J, \tag{4.8a}$$

$$\pi_i^I = \pi_i^J \quad \text{for } i \in \mathcal{W}^I \cap \mathcal{W}^J, \tag{4.8b}$$

$$\pi_i^J = 0 \quad \text{for } i \in \mathcal{W}^J \backslash \mathcal{W}^I. \tag{4.8c}$$

Since Lemma 4.4 implies that there is an integer K such that $\mathcal{W}_{k+1} \backslash \mathcal{W}_k \neq \emptyset$ for all $k \in I$ and $k \geq K$, we conclude that $\mathcal{W}^J \backslash \mathcal{W}^I \neq \emptyset$. Since no constraints have been deleted between iterations k and l_k for $k \in I''$, any constraints whose index is in the set $\mathcal{W}^I \backslash W^J$ must have been deleted in an iteration $k \in I''$. Since $I'' \subseteq I$, it follows that $\pi_{\min,k} \leq -\epsilon$ for $k \in I''$. From the rule for moving off a constraint, (3.4b), we can deduce that $(\pi_k)_i \leq -\nu\epsilon$ for $k \in I''$ and $i \in \mathcal{W}^I \backslash \mathcal{W}^J$, where $\nu \in (0,1)$. Since $\lim_{k \in I''} \pi_k = \pi^I$, we conclude that $\pi_i^I \leq -\nu\epsilon$ for $i \in \mathcal{W}^I \backslash \mathcal{W}^J$. Hence, (4.8a) implies that $\mathcal{W}^I \backslash \mathcal{W}^J = \emptyset$. Consequently, it must hold that $|\mathcal{W}^J| \geq |\mathcal{W}^I| + 1$ and, by (4.8c), π^J has at least one component zero.

We can conclude from (4.8b) that $\pi_{\min,l_k} < -0.5\epsilon$ for $k \in I''$ and k sufficiently large. The rules for computing q_k, (3.4a), ensure that there is a subsequence $I''' \subseteq I''$ such that $q_{l_k} \neq 0$ for all $k \in I'''$. From the definition of l_k, it holds that $q_{l_k - 1} = 0$ for all $k \in I'''$. Therefore, if $J' = \{l_k : k \in I'''\}$, we may replace I by J' and repeat the argument. Since $|\mathcal{W}^J| \geq |\mathcal{W}^I| + 1$ and $|\mathcal{W}_k| \leq \min\{m, n\}$ for any k, after having repeated the argument at most $\min\{m, n\}$ times we have a contradiction to assumption A3, implying that the assumed existence of a subsequence I such that $q_{k-1} = 0$ and $q_k \neq 0$ and $\pi_{\min,k} < -\epsilon$ for $k \in I$ is false. Thus, the result of the lemma follows. □

We are now in the position to give the main convergence result. In addition to the global convergence established here, we also add a well-known rate-of-convergence result from Moré and Sorensen [22].

Theorem 4.6. *Given assumptions* A1–A3, *assume that a sequence* $\{x_k\}_{k=0}^{\infty}$ *is generated as outlined in section* 3. *Then, any limit point* x^* *satisfies the second-order necessary optimality conditions*; *i.e., if the constraint matrix associated with the active constraints at* x^* *is denoted by* A_A, *there is a vector* π_A *such that*

$$\nabla f(x^*) = A_A^T \pi_A, \quad \pi_A \geq 0,$$

and it holds that

$$\lambda_{\min}(Z_A^T \nabla^2 f(x^*) Z_A) \geq 0,$$

where Z_A *denotes a matrix whose columns form an orthonormal basis for the null space of* A_A.

If, in addition, $\lambda_{\min}(Z_A^T \nabla^2 f(x^*) Z_A) > 0$ *and* $\pi_A > 0$ *hold, then* $\lim_{k \to \infty} x_k = x^*$. *Further, for* k *sufficiently large, it follows that if* $s_k = -Z_A(Z_A^T H_k Z_A)^{-1} Z_A^T g_k$ *then* s_k *is sufficient in the sense of* (3.1), $p_k = s_k$, *and* $\alpha_k = 1$ *satisfies* (3.5) *and* (3.6). *Moreover, for this choice of* s_k *and* α_k, *the rate of convergence is at least q-quadratic, provided the second-derivative matrix is Lipschitz continuous in a neighborhood of* x^*.

Proof. Let x^* denote a limit point of a generated sequence of iterates. By assumption A2, there is a subsequence I such that $\lim_{k \in I} x_k = x^*$. We claim that this implies the existence of a subsequence I' such that $\lim_{k \in I'} x_k = x^*$, $q_{k-1} = 0$ and $A_{k-1} = A_k = \hat{A}$ for each $k \in I'$, where $\hat{A}$ denotes a matrix which is identical for each $k \in I'$. For $k \in I$, an iterate l_k is defined as follows. If $q_k \neq 0$, let l_k be the iteration with largest index that does not exceed k for which $q_{l_k-1} = 0$. Since no constraints are deleted immediately upon adding constraints, we obtain $q_{l_k-1} = 0$, $q_{l_k} \neq 0$, $\mathcal{W}_{l_k-1} = \mathcal{W}_{l_k}$, and $k - m \leq l_k \leq k$. If $q_k = 0$, let l_k denote the following iteration with least index such that $\mathcal{W}_{l_k} = \mathcal{W}_{l_k-1}$. If $q_{l_k-1} \neq 0$, the properties of q_{l_k-1} and the rules for updating the working set give $\mathcal{W}_{l_k} \neq \mathcal{W}_{l_k-1}$. Hence, for this case, we must have $q_{l_k-1} = 0$. Since no constraints are deleted immediately upon adding constraints, it follows that l_k is the following iteration with least index when no constraint is added. For this case, we obtain $q_{l_k-1} = 0$, $\mathcal{W}_{l_k-1} = \mathcal{W}_{l_k}$, and $k+1 \leq l_k \leq k+m$. It follows from (iii) of Lemma 4.1 that $\lim_{k\in I} \|x_k - x_{l_k}\| \to 0$, and hence $\lim_{k \in I} x_{l_k} = x^*$. With $\{l_k\}_{k\in I}$ defined this way, since there is only a finite number of different active-set matrices, the required subsequence I' can be obtained as a subsequence of $\{l_k\}_{k\in I}$.

Since, for each $k \in I'$, an unrestricted step is taken at iteration $k-1$, assumptions A1 and A2 in conjunction with property (iii) of Lemma 4.3 give

$$\hat{Z}^T \nabla f(x^*) = 0 \quad \text{and} \quad \lambda_{\min}(\hat{Z}^T \nabla^2 f(x^*) \hat{Z}) \geq 0, \tag{4.9}$$

where $\hat{Z}$ denotes an orthonormal matrix whose columns form a basis for the null space of $\hat{A}$. Since $\lim_{k\in I'} \hat{Z}^T g_k = 0$ and $\hat{A}$ has full row rank, it follows from (3.3) and (4.9) that

$$\nabla f(x^*) = \hat{A}^T \hat{\pi} \quad \text{for} \quad \hat{\pi} = \lim_{k \in I'} \pi_k. \tag{4.10}$$

It remains to show that $\min_i \hat{\pi}_i \geq 0$. Assume that there is a subsequence $I'' \subseteq I'$ and an $\epsilon > 0$ such that $\pi_{\min,k} < -\epsilon$ for $k \in I''$. Lemma 4.5 shows that there exists a K such that $q_k = 0$ for $k \in I''$ and $k \geq K$. But this contradicts (3.4a), and since $\hat{\pi} = \lim_{k\in I'} \pi_k$, we conclude that

$$\min_i \hat{\pi}_i \geq 0. \tag{4.11}$$

A combination of (4.9), (4.10), and (4.11) now ensures that x^* satisfies the second-order necessary optimality conditions. If there are constraints in A_A that are not in $\hat{A}$, the associated Lagrange multipliers are zero, i.e., π_A equals $\hat{\pi}$ possibly extended by zeros. Also, in this situation, the range space of Z_A is contained in the range space of $\hat{Z}$. Hence, $\lambda_{\min}(\hat{Z}^T \nabla^2 f(x^*) \hat{Z}) \geq 0$ implies $\lambda_{\min}(Z_A^T \nabla^2 f(x^*) Z_A) \geq 0$.

To show the second half of the theorem, note that if $\pi_A > 0$, then we must have $\hat{\pi} = \pi_A$, and it follows from (4.10) that there cannot exist a subsequence $\tilde{I}' \subseteq I'$ such that $\pi_{\min,k} < 0$ for $k \in \tilde{I}'$. This implies that there is an iteration $\tilde{K}$ such that $A_k = \hat{A}$ and $q_k = 0$ for $k \geq \tilde{K}$. Then the problem may be written as an equality-constrained problem in the null space of $\hat{A}$, namely

$$\begin{array}{ll} \underset{x\in \mathbb{R}^n}{\text{minimize}} & f(x) \\ \text{subject to} & \hat{A}x = \hat{b}, \end{array} \tag{4.12}$$

where $\hat{b}$ denotes the corresponding subvector of b. If $\hat{Z}^T \nabla^2 f(x^*) \hat{Z}$ is now positive definite, then (iii) of Lemma 4.1 and (3.5) ensure that the limit point is unique, i.e., $\lim_{k\to\infty} x_k = x^*$. From the continuity of f, it follows that $\hat{Z}^T H_k \hat{Z}$ is positive definite for k sufficiently large. Hence, it must hold that $d_k = 0$ and $p_k = s_k$ for k sufficiently large. If $s_k = -Z_A(Z_A^T H_k Z_A)^{-1} Z_A^T g_k$, then s_k is sufficient in the sense of

(3.1) provided that $k \geq K$ and x_k is sufficiently close to x^*. Also, this choice of s_k is the Newton step for solving (4.12), and it follows from Moré and Sorensen [22, p. 53] that $\alpha_k = 1$ eventually satisfies (3.5) and (3.6). Moreover, Moré and Sorensen [22, Theorem 2.3] show that under these assumptions $\lim_{k\to\infty} x_k = x^*$ and the rate of convergence is *q*-quadratic provided the second-derivative matrix is Lipschitz continuous in a neighborhood of x^* [22, Theorem 2.8]. □

5. Computation of the search direction for large-scale problems.

We now show how to compute s_k, d_k, π_k, and q_k that satisfy the properties of sections 3.1, 3.2, 3.3, and 3.4, respectively. A way of updating A_k to satisfy the properties of section 3.6 is also given. Our particular interest is large-scale problems for which no prior assumptions are made about the number of constraints in the problem or the number of constraints active at the solution. This precludes the use of the reduced Hessian.

Forsgren and Murray [11] describe how suitable search directions can be computed for large-scale linear *equality-constrained* problems without the need to form the reduced Hessian. The technique they describe can be utilized also in the current context for computing a suitable descent direction s_k, a suitable Lagrange multiplier vector π_k, and a suitable direction of negative curvature d_k. We briefly review this approach here. The key procedure is an indefinite symmetric factorization of the Karush–Kuhn–Tucker (KKT) matrix K_k, where

$$K_k = \begin{pmatrix} H_k & A_k^T \\ A_k & 0 \end{pmatrix}. \tag{5.1}$$

The factorization is an LBL^T *factorization*, i.e., a factorization of the form

$$\Pi_k^T K_k \Pi_k = L_k B_k L_k^T,$$

where Π_k is a permutation matrix, L_k is a unit lower-triangular matrix, and B_k is a symmetric block-diagonal matrix whose diagonal blocks are of size 1×1 or 2×2. For a general LBL^T factorization, the permutations are performed in order to obtain a matrix L_k that is sparse and well conditioned; see, e.g., Duff and Reid [8], [9]. It is shown in Forsgren and Murray [11] that by potentially requiring additional permutations, suitable s_k and d_k can be computed from one single factorization of K_k. We demonstrate below that the additional quantities π_k and q_k can also be computed from the same factors. In the discussion below, the inertia of $Z_k^T H_k Z_k$ is required. Note that this inertia can be deduced from the inertia of K_k; see Gould [19, Lemma 3.4]. First we show how to choose A_0.

5.1. Finding an A_0 with full row rank.

It is required that A_0 has full row rank. Let $\bar{A}_0$ denote the matrix composed of all the rows of A corresponding to the active set at x_0. A straightforward way to determine A_0 is to form an LU-factorization of $\bar{A}_0^T$. An alternative approach, which fits well with the discussion of section 5, is to form the symmetric factorization of K_0 described in Forsgren and Murray [11], with $A_0 = \bar{A}_0$. In forming the factorization a redundant constraint is identified if its associated pivot is zero. The factorization may then be terminated prematurely when only redundant rows are left.

5.2. Computation of $\boldsymbol{s}_k$ and $\boldsymbol{\pi}_k$. The computation of s_k and π_k is identical to the computation of s_k in Forsgren and Murray [11]. We solve

$$\begin{pmatrix} \bar{H}_k & A_k^T \\ A_k & 0 \end{pmatrix} \begin{pmatrix} s_k \\ -\pi_k \end{pmatrix} = \begin{pmatrix} -g_k \\ 0 \end{pmatrix}, \tag{5.2}$$

and $\bar{H}_k = H_k$ when $Z_k^T H_k Z_k$ is sufficiently positive definite; otherwise, $\bar{H}_k$ is a modification of H_k such that $Z_k^T \bar{H}_k Z_k$ is sufficiently positive definite and has bounded norm. It is shown in Forsgren and Murray [11] how the factors of $\bar{K}_k$ may be obtained directly from those of K_k, where $\bar{K}_k$ denotes the modified matrix of (5.2) and K_k is given by (5.1). The matrix $\bar{K}_k$ is bounded away from a singular matrix, $\bar{H}_k$ is bounded, and $Z_k^T \bar{H}_k Z_k$ is positive definite with bounded condition number and smallest eigenvalue bounded away from zero. It is straightforward to verify that s_k from (5.2) can be written as

$$s_k = -Z_k (Z_k^T \bar{H}_k Z_k)^{-1} Z_k^T g_k, \tag{5.3}$$

and it follows that s_k is sufficient in the sense of (3.1). Moreover, assumptions A1 and A2 ensure that s_k has bounded norm if evaluated in the set $\{x : Ax \geq b, f(x) \leq f(x_0)\}$.

A combination of (5.2) and (5.3) gives

$$g_k - A_k^T \pi_k = \bar{H}_k Z_k (Z_k^T \bar{H}_k Z_k)^{-1} Z_k^T g_k,$$

and it follows that π_k satisfies (3.3).

5.3. Computation of $\boldsymbol{d}_k$. The computation of d_k is identical to the computation of d_k in Forsgren and Murray [11]. If $Z_k^T H_k Z_k$ is positive definite then $d_k = 0$; otherwise, we may define a suitable d_k as the solution of a system of the form

$$\begin{pmatrix} H_k & A_k^T \\ A_k & 0 \end{pmatrix} \begin{pmatrix} d_k \\ -\mu_k \end{pmatrix} = \begin{pmatrix} u_k \\ 0 \end{pmatrix}$$

for some suitable vector u_k. Forsgren and Murray [11] show how to compute d_k by a single solve with the triangular factor L_k without the need to form u_k explicitly. They also show that d_k is sufficient in the sense of (3.2) and that it has bounded norm.

5.4. Computation of $\boldsymbol{q}_k$. We may compute a suitable q_k using the matrix $\bar{K}_k$ and the vector π_k from (5.2). As was mentioned when describing the computation of s_k and π_k, the factors of $\bar{K}_k$ may be obtained directly from those of K_k. For a positive tolerance ν, $(0 < \nu \leq 1)$, we first compute a vector v_k such that $(v_k)_i = -(\pi_k)_i$ if $(\pi_k)_i \leq \nu \pi_{\min,k}$ and $(v_k)_i = 0$ if $(\pi_k)_i > \nu \pi_{\min,k}$. The direction q_k is then obtained from the system

$$\begin{pmatrix} \bar{H}_k & A_k^T \\ A_k & 0 \end{pmatrix} \begin{pmatrix} q_k \\ -\eta_k \end{pmatrix} = \begin{pmatrix} 0 \\ v_k \end{pmatrix}. \tag{5.4}$$

The following lemma shows that a nonzero q_k is a descent direction such that $A_k q_k \geq 0$.

LEMMA 5.1. *Let s_k and π_k be defined from* (5.2). *If $\pi_{\min,k} < 0$ and q_k and η_k are defined from* (5.4), *then $q_k^T g_k = \pi_k^T v_k \leq -\pi_{\min,k}^2$ and $A_k q_k = v_k \geq 0$.*

Proof. Premultiplication of both sides of (5.4) by the vector $(s_k^T \ -\pi_k^T)$ from (5.2) yields

$$\begin{pmatrix} s_k^T & -\pi_k^T \end{pmatrix} \begin{pmatrix} \bar{H}_k & A_k^T \\ A_k & 0 \end{pmatrix} \begin{pmatrix} q_k \\ -\eta_k \end{pmatrix} = \begin{pmatrix} s_k^T & -\pi_k^T \end{pmatrix} \begin{pmatrix} 0 \\ v_k \end{pmatrix}. \tag{5.5}$$

Utilization of (5.2) and the symmetry of $\bar{H}_k$ in the left-hand side of (5.5) yields

$$\begin{pmatrix} -g_k^T & 0 \end{pmatrix} \begin{pmatrix} q_k \\ -\eta_k \end{pmatrix} = \begin{pmatrix} s_k^T & -\pi_k^T \end{pmatrix} \begin{pmatrix} 0 \\ v_k \end{pmatrix}. \tag{5.6}$$

Simplification of (5.6) gives $q_k^T g_k = \pi_k^T v_k$. The definition of v_k yields

$$\pi_k^T v_k = - \sum_{i:(\pi_k)_i \le \nu \pi_{\min,k}} (\pi_k)_i^2 \le -\pi_{\min,k}^2.$$

Moreover, it follows from the definition of v_k that $v_k \ge 0$, and (5.4) implies $A_k q_k = v_k \ge 0$, as required. □

The norm of π_k is bounded because of the properties of $\bar{K}_k$ and assumptions A1 and A2. Hence, since $Z_k^T \bar{H}_k Z_k$ is positive definite and has bounded norm, we conclude that q_k computed from (5.4) has bounded norm. It follows from (5.4) that $a_i^T q_k = 0$ if $(\pi_k)_i > \nu \pi_{\min,k}$ for $i \in \mathcal{W}_k$, and hence (3.4b) holds. Lemma 5.1 implies that

$$\lim_{k \in I} g_k^T q_k = 0 \quad \Rightarrow \quad \liminf_{k \in I} \pi_{\min,k} \ge 0, \tag{5.7}$$

where I is any subsequence such that q_k is computed from (5.4) for $k \in I$ and hence (3.4a) holds.

5.5. Combination of the search direction. It is not specified in sections 3.1–3.4 exactly how to choose s_k, d_k, π_k, and q_k. Sections 5.2–5.4 give suitable ways of computing these quantities. In certain situations these components are necessarily zero; if $Z_k^T g_k = 0$ then $s_k = 0$, if $Z_k^T H_k Z_k$ is positive semidefinite then $d_k = 0$, and if $\pi_{\min,k} \ge 0$ or $\mathcal{W}_k \not\subseteq \mathcal{W}_{k-1}$ then $q_k = 0$. However, it may be desirable occasionally to let some components be zero even when it is not necessary. For example, having a nonzero q_k whenever possible may not be the most efficient strategy. If the current reduced Hessian has many negative eigenvalues this suggests more constraints should be active rather than less. It is possible to impose a rule that only considers deleting constraints when to do so significantly impacts p_k. The property (3.4a) required of q_k suggests having an additional condition saying that $q_k = 0$ if

$$\pi_{\min,k} \ge \beta \left(g_k^T s_k + d_k^T H_k d_k \right),$$

where β is a positive constant. Since Lemma 4.3 implies that $\lim_{k\to\infty} g_k^T s_k = 0$ and $\lim_{k\to\infty} d_k^T H_k d_k = 0$ for unrestricted steps, such a condition does not impact on (3.4a), and hence it does not alter the convergence analysis. Similar conditions can be imposed to set $s_k = 0$ or $d_k = 0$ at certain iterations.

5.6. The update of $\boldsymbol{A_k}$. The working-set matrix A_k is required to have full row rank. A straightforward way to ensure this property is to add at most one constraint at every iteration, as the following lemma shows.

LEMMA 5.2. *Given assumptions* A1–A3*, assume that a sequence* $\{x_k\}_{k=0}^{\infty}$ *is generated as outlined in section* 3*. If* A_0 *has full row rank,* $|\mathcal{W}_{k+1}| \leq |\mathcal{W}_k^0| + 1$*,* $a_i^T p_k < 0$ *for all* $k \geq 0$*, and* $i \in \mathcal{W}_{k+1} \backslash \mathcal{W}_k$*, then each* A_k *has full row rank.*

Proof. See, e.g., Gill et al. [16, Lemma 2.1]. □

Although the computed search directions described in sections 5.2–5.4 are not designed specifically to add more than one constraint per iteration, the convergence analysis presented gives room for defining algorithms that add any number of active constraints, as long as the working-set matrix has full row rank. The issue would be twofold: (i) to modify the definitions of the search directions, so as to make more than one new constraint become active in the linesearch, while still maintaining the required properties of these directions, and (ii) to maintain the full rank of the working-set matrix. This approach may be advantageous for certain problems, e.g., problems where all constraints are simple bounds. In this situation, it is known a priori that any working-set matrix will have full row rank. Techniques similar to gradient projection, see, e.g., Calamai and Moré [7], might prove useful for altering the search direction.

6. Primal degeneracy.

Assumptions A1 and A2 ensure that the objective function is sufficiently smooth and the iterates remain in a feasible region. Assumption A3 implies that no primal degenerate second-order constrained stationary points exist. Although for nonlinear problems degeneracy is not as common in practice as it is for linear programming problems, there are problems for which A3 does not hold. Consequently, in a practical implementation of our algorithm some technique to handle degeneracy is necessary. The nature of degeneracy is different for nonlinear problems. In linear programming the main concern is degenerate vertices. In effect the iterate is at the degenerate stationary point. In a nonlinear problem we may never be at the stationary point. Moreover it is likely not to be a vertex. What we are likely to encounter is rank deficient active-set matrices for which the number of rows is less than n, and we are not at a constrained stationary point. We need only be concerned if we plan to delete constraints. In exact arithmetic we could define a subiteration to search for a suitable active set. A method of implementing this strategy that makes use of the known factorization of the KKT matrix is described in Gill et al. [15]. Such an approach is an improvement over algorithms based on sequential quadratic programming where a subiteration may be necessary at each iteration of the quadratic programming subproblem. The difficulty with this strategy is the need to define the active set. In inexact arithmetic precisely what is the active set is not clear. We prefer therefore to rely on the approach adopted by Gill et al. [14]. This technique allows infeasibility tolerances on the constraints that are altered at each iteration. The impact on the algorithm is that a zero step is never taken. The consequences of allowing infeasibility tolerances is that the solution obtained may be infeasible. However, the maximum degree of infeasibility may be specified. In practice the maximum infeasibility allowed when solving nonlinear problems is unlikely to be attained and is in any event consistent with the infeasibility that results from the impact of finite precision operations. An advantage of this approach is that it is equally useful for handling near degeneracy. This is likely to be common on problems where the linearly constrained problem being solved is an approximation to a nonlinearly constrained problem whose Jacobian is rank deficient at the solution. The use of a procedure similar to that in [14] is in any event essential in practice for the purpose of trying to introduce a choice in the definition of A_k in an attempt to ensure that the condition number of A_k is not too large. For example, if infeasibilities are allowed then nearly dependent active constraints need not be included in the working set. The search direction will not be exactly orthogonal to the constraint normals of the constraints ignored but it will be close, hence the next iterate will not be too infeasible.

7. Discussion. A convergence analysis for an algorithm to solve linear inequality-constrained optimization problems has been presented. The algorithm is described in broad terms by assuming the availability at each iteration of three directions with certain properties. It has also been shown how to compute all the required search directions from a single symmetric indefinite factorization of the KKT matrix. Such an algorithm is well suited to solving large-scale problems. Unlike some alternatives the efficiency of the method is not dependent on either the active set or the null space of the active set being small.

For convenience of notation, the problem is stated in all-inequality form (1.1), but we emphasize that the analysis can be modified in a straightforward manner to cover the case with a mixture of inequality and equality constraints. A particularly attractive feature of the algorithm described is that the problem does not have to be transformed into a specific form.

Acknowledgments. We thank the referees for their helpful comments and criticism, which significantly improved the presentation.

REFERENCES

[1] A. Buckley, *An alternative implementation of Goldfarb's minimization algorithm*, Math. Prog., 8 (1975), pp. 207–231.

[2] J. Burke, *On the identification of active constraints* II: *The nonconvex case*, SIAM J. Numer. Anal., 27 (1990), pp. 1081–1102.

[3] J. V. Burke and J. J. Moré, *On the identification of active constraints*, SIAM J. Numer. Anal., 25 (1988), pp. 1197–1211.

[4] J. V. Burke and J. J. Moré, *Exposing constraints*, SIAM J. Optim., 4 (1994), pp. 573–595.

[5] J. V. Burke, J. J. Moré, and G. Toraldo, *Convergence properties of trust region methods for linear and convex constraints*, Math. Prog., 47 (1990), pp. 305–336.

[6] R. H. Byrd and G. A. Shultz, *A Practical Class of Globally Convergent Active Set Strategies for Linearly Constrained Optimization*, Tech. rep. CU-CS-238-82, Department of Computer Science, University of Colorado at Boulder, Boulder, CO, 1982.

[7] P. H. Calamai and J. J. Moré, *Projected gradient methods for linearly constrained problems*, Math. Prog., 39 (1987), pp. 93–116.

[8] I. S. Duff and J. K. Reid, *MA*27*: A Set of Fortran Subroutines for Solving Sparse Symmetric Sets of Linear Equations*, Tech. rep. R-10533, Computer Science and Systems Division, AERE Harwell, Oxford, England, 1982.

[9] I. S. Duff and J. K. Reid, *The multifrontal solution of indefinite sparse symmetric linear equations*, ACM Trans. Math. Software, 9 (1983), pp. 302–325.

[10] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., John Wiley Sons, Chichester, New York, Brisbane, Toronto, Singapore, 1987.

[11] A. Forsgren and W. Murray, *Newton methods for large-scale linear equality-constrained minimization*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 560–587.

[12] D. M. Gay, *A trust-region approach to linearly constrained optimization*, in Numerical Analysis, D. F. Griffiths, ed., Lecture Notes in Mathematics 1066, Springer-Verlag, Berlin, Heidelberg, New York, 1984, pp. 72–105.

[13] P. E. Gill and W. Murray, *Newton-type methods for unconstrained and linearly constrained optimization*, Math. Prog., 7 (1974), pp. 311–350.

[14] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *A practical anti-cycling procedure for linearly constrained optimization*, Math. Prog., 45 (1989), pp. 437–474.

[15] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *A Schur-complement method for sparse quadratic programming*, in Reliable Numerical Computation, M. G. Cox and S. J. Hammarling, eds., Oxford University Press, London, England, 1990, pp. 113–138.

[16] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *Inertia-controlling methods for general quadratic programming*, SIAM Rev., 33 (1991), pp. 1–36.

[17] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London, New York, 1981.

[18] D. Goldfarb, *Extensions of Davidon's variable metric method to maximization under linear inequality and equality constraints*, SIAM J. Appl. Math., 17 (1969), pp. 739–764.

[19] N. I. M. Gould, *On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem*, Math. Prog., 32 (1985), pp. 90–99.

[20] G. P. McCormick, *A second order method for the linearly constrained nonlinear programming problem*, in Nonlinear Programming, J. B. Rosen, O. L. Mangasarian, and K. Ritter, eds., Academic Press, New York, London, 1970, pp. 207–243.

[21] J. J. Moré and D. C. Sorensen, *On the use of directions of negative curvature in a modified Newton method*, Math. Prog., 16 (1979), pp. 1–20.

[22] J. J. Moré and D. C. Sorensen, *Newton's method*, in Studies in Numerical Analysis, Studies in Mathematics, Vol. 24, G. H. Golub, ed., The Mathematical Association of America, 1984, pp. 29–82.

[23] B. A. Murtagh and M. A. Saunders, *Large-scale linearly constrained optimization*, Math. Prog., 14 (1978), pp. 41–72.

[24] B. A. Murtagh and M. A. Saunders, *MINOS* 5.4 *User's Guide*, Tech. rep. SOL 83-20R, Department of Operations Research, Stanford University, Stanford, CA, 1993.

[25] K. Ritter, *A superlinearly convergent method for minimization problems with linear inequality constraints*, Math. Prog., 4 (1973), pp. 44–71.

[26] K. Ritter, *A method of conjugate direction for linearly constrained nonlinear programming problems*, SIAM J. Numer. Anal., 12 (1975), pp. 274–303.

[27] K. Ritter, *Convergence and superlinear convergence of algorithms for linearly constrained minimization problems*, in Nonlinear Optimization: Theory and Algorithms, Part II, L. C. W. Dixon, E. Spedicato, and G. P. Szegö, eds., Birkhäuser, Boston, MA, 1980, pp. 221–251.

[28] J. B. Rosen, *The gradient projection method for nonlinear programming. Part* I. *Linear constraints*, SIAM J. Appl. Math., 8 (1960), pp. 181–217.

SIAM J. OPTIM.
Vol. 7, No. 1, pp. 177–207, February 1997

A GLOBAL CONVERGENCE THEORY FOR GENERAL TRUST-REGION-BASED ALGORITHMS FOR EQUALITY CONSTRAINED OPTIMIZATION*

J. E. DENNIS, JR.[†], MAHMOUD EL-ALEM[‡], AND MARIA C. MACIEL[§]

Abstract. This work presents a global convergence theory for a broad class of trust-region algorithms for the smooth nonlinear programming problem with equality constraints. The main result generalizes Powell's 1975 result for unconstrained trust-region algorithms.

The trial step is characterized by very mild conditions on its normal and tangential components. The normal component need not be computed accurately. The theory requires a quasi-normal component to satisfy a fraction of Cauchy decrease condition on the quadratic model of the linearized constraints. The tangential component then must satisfy a fraction of Cauchy decrease condition on a quadratic model of the Lagrangian function in the translated tangent space of the constraints determined by the quasi-normal component. Estimates of the Lagrange multipliers and the Hessians are assumed only to be bounded.

The other main characteristic of this class of algorithms is that the step is evaluated by using the augmented Lagrangian as a merit function with the penalty parameter updated using the El-Alem scheme. The properties of the step and the way that the penalty parameter is chosen are sufficient to establish global convergence.

As an example, an algorithm is presented that can be viewed as a generalization of the Steihaug–Toint dogleg algorithm for the unconstrained case. It is based on a quadratic programming algorithm that uses a step in a quasi-normal direction to the tangent space of the constraints and then takes feasible conjugate reduced-gradient steps to solve the reduced quadratic program. This algorithm should cope quite well with large problems for which effective preconditioners are known.

Key words. constrained optimization, global convergence, trust regions, equality constrained, nonlinear programming, conjugate gradient, inexact Newton Method

AMS subject classifications. 65K05, 49D37

PII. S1052623492238881

1. Introduction.

This work is concerned with the development of a global convergence theory for a broad class of algorithms for the equality constrained minimization problem:

$$(\text{EQC}) \equiv \begin{cases} \text{minimize} & f(x) \\ \text{subject to} & C(x) = 0. \end{cases}$$

The functions $f : \Re^n \to \Re$ and $C : \Re^n \to \Re^m$ are at least twice continuously differentiable where $C(x) = (c_1(x), \ldots, c_m(x))^T$ and $m < n$.

Our purpose is to generalize to constrained problems a powerful theorem given in 1975 by Powell for unconstrained problems.

The global convergence theory that we establish in this work holds for a class of nonlinear programming algorithms for (EQC) that is characterized by the following features:

[*] Received by the editors October 14, 1992; accepted for publication (in revised form) October 6, 1995. This research was supported by DOE DE-FG005-86ER25017, CRPC CCR-9120008, AFOSR-F49620-95-1-0210, and the REDI Foundation.

http://www.siam.org/journals/siopt/7-1/23888.html

[†] Department of Computational and Applied Mathematics & Center for Research on Parallel Computation, Rice University, P. O. Box 1892, Houston TX 77251 (dennis@ariel.rice.edu).

[‡] Department of Mathematics, Faculty of Science, Alexandria University, Alexandria, Egypt (elalem@alex.eun.eg).

[§] Departamento de Matematica, Universidad Nacional del Sur, Avenida Alem 1253, 8000 Bahia Blanca, Argentina.

1. The algorithms of the family use the *trust-region approach* as a globalization strategy.
2. All these algorithms generate steps that satisfy our mild conditions on the trial step's normal and tangential components. It is important to note that the condition on our "quasi-normal" step s_c^n, that $\|s_c^n\|_2 \leq K_1\|C(x_c)\|_2$ for some independent constant K_1, is always satisfied under our hypotheses if a truly normal component is used for the trial step. The other conditions are that the quasi-normal component satisfies a *fraction of Cauchy decrease* condition on the quadratic model of the linearized constraints and that the tangential component (as measured from the quasi-normal component) satisfies a *fraction of Cauchy decrease* on the quadratic model of the reduced Lagrangian function associated with (EQC).
3. The estimates of the Lagrange multiplier vector and the Hessian matrix are assumed only to be bounded uniformly across all iterations.
4. The step is evaluated for acceptance by using the augmented Lagrangian function with penalty parameter updated by the El-Alem scheme [9]. A key point here is that the step is computed before the penalty parameter, which will be used to evaluate the step, is updated.

Points 1 and 3 are satisfied by the algorithms of Byrd, Schnabel, and Shultz [2]; Omojokun [21]; Celis, Dennis, and Tapia [4]; and Powell and Yuan [23]. The first two papers require a normal, rather than just a quasi-normal, s_c^n in point 2.

We use the following notation: the sequence of points generated by an algorithm is denoted by $\{x_k\}$. This work also uses subscripts -, c, and $+$ to denote the previous, the current, and the next iterates, respectively. However, when we need to work with a whole sequence we will use the index k. The matrix H_c denotes the Hessian of the Lagrangian at the current iterate or an approximation to it. Subscripted functions mean the function value at a particular point; for example, $f_c = f(x_c)$, $\ell_c = \ell(x_c, \lambda_c)$, and so on. Finally, unless otherwise specified, all the norms are ℓ_2-norms, and we use the same symbol 0 to denote the real number zero and the zero vector.

The rest of the paper is organized as follows: in section 2, we review the concept of fraction of Cauchy decrease. In section 3, we review the SQP algorithm. In section 4, we survey existing trust-region algorithms for solving problem (EQC). In section 5, we present a general trust-region algorithm with the conditions that the trial step must satisfy. In section 6 we state the algorithm. Sections 7 and 8 present the global convergence theory that we have developed. In section 7.1, we state the assumptions under which global convergence is established. In section 7.2, we discuss some properties of the trial steps. In section 7.3, we study the behavior of the penalty parameter. Section 8 presents our main global convergence result. In section 9, we present, as an example, an algorithm that solves problem (EQC), and we prove that it fits the assumptions of the paper. This algorithm was one we had in mind as motivation for the convergence theory. It can be viewed as a generalization to the constrained problem of the Steihaug–Toint dogleg algorithm for the unconstrained case. This algorithm has worked quite well for some large problems. Finally, we make some concluding remarks in section 10.

2. Fraction of Cauchy decrease condition.

Consider the following unconstrained minimization problem

$$\text{(UCMIN)} \equiv \begin{cases} \text{minimize} & f(x) \\ \text{subject to} & x \in \Re^n, \end{cases}$$

where $f : \Re^n \to \Re$ is a continuously differentiable function. A trust-region algorithm for solving the above problem is an iterative procedure that computes *a trial step* as an approximate solution to the following *trust-region subproblem:*

$$\text{(TRS)} \equiv \begin{cases} \text{minimize} & m_c(s) = f_c + \nabla f_c^T s + \frac{1}{2} s^T G_c s \\ \text{subject to} & \|s\| \leq \delta_c, \end{cases}$$

where G_c is the Hessian matrix $\nabla^2 f_c$ or an approximation to it and $\delta_c > 0$ is a given trust-region radius. For a complete survey see Moré [18] and the book by Dennis and Schnabel [7].

To assure global convergence, the step is required only to satisfy a *fraction of Cauchy decrease* condition. This means that s_c must predict via the quadratic model function m_c at least as much as a fraction of the decrease given by the Cauchy step on m_c; that is, there exists a constant $\sigma > 0$ fixed across all iterations, such that

$$m_c(0) - m_c(s_c) \geq \sigma[m_c(0) - m_c(s_c^{\text{cp}})], \tag{2.1}$$

where $s_c^{\text{cp}} = -t_c^{\text{cp}} \nabla f_c$ and its step length

$$t_c^{\text{cp}} = \begin{cases} \frac{\|\nabla f_c\|^2}{\nabla f_c^T G_c \nabla f_c} & \text{if } \frac{\|\nabla f_c\|^3}{\nabla f_c^T G_c \nabla f_c} \leq \delta_c \text{ and } \nabla f_c^T G_c \nabla f_c > 0, \\ \frac{\delta_c}{\|\nabla f_c\|} & \text{otherwise.} \end{cases}$$

Thus, s_c^{cp} is the steepest descent step for m_c inside the trust region.

The form of (2.1) we use to prove convergence is given in the following technical lemma. More details about the role of this lemma in the convergence theory of trust-region algorithms can be found in Carter [3], Moré [18], Powell [22], and Shultz, Schnabel, and Byrd [25].

LEMMA 2.1. *If the trial step s_c satisfies a fraction of Cauchy decrease condition, then*

$$m_c(0) - m_c(s_c) \geq \frac{\sigma}{2} \|\nabla f_c\| \min\left\{ \frac{\|\nabla f_c\|}{\|G_c\|}, \delta_c \right\}. \tag{2.2}$$

Proof. See Powell [22] for the proof. □

We end this section by stating Powell's powerful theorem for unconstrained trust-region algorithms. The proof can be found in Powell [22]. More details about the convergence theory for trust-region algorithms for unconstrained optimization can be found in Fletcher [14], Moré [18], Moré and Sorensen [19], and Sorensen [26].

THEOREM 2.2. *Let $f : \Re^n \to \Re$ be continuously differentiable and bounded below on the level set $\{x \in \Re^n : f(x) \leq f(x_0)\}$. Assume that the sequence $\{G_k\}$ is uniformly bounded. If $\{x_k\}$ is the sequence generated by any trust-region algorithm that satisfies* (2.1) *or* (2.2), *then*

$$\liminf_{k \to \infty} \|\nabla f_k\| = 0.$$

Notice that this theorem does not prove convergence to a solution of the unconstrained problem; rather, it proves a "weak" first-order convergence. However, we do not see that as the point of this theorem, nor is it surprising given the weak assumptions on the sequence of local models. In other words, this theorem is not about convergence conditions on a quasi-Newton method. Such a theorem would be expected to be based on analyzing some way of estimating the Hessian, and we all know how important the method for estimating the Hessian is in the practical performance of a trust-region algorithm. In the unconstrained case, one version of Powell's theorem, which says that the sequence of gradients converges to zero, requires the additional hypothesis that the gradient is uniformly continuous. The algorithms here would probably require a uniformly continuous reduced gradient, a strengthening of the assumptions used here. The related algorithms mentioned earlier also prove weak first-order stationary convergence, as do we.

The point of this line of research is not to give a convergence proof for a specific SQP approach using a specific Lagrange multiplier estimation technique and perhaps an exact merit function. Instead, the point is to give an analysis of the local quadratic-model/trust-region paradigm for unconstrained optimization. In that context, this theorem says that the power of using a trust-region globalization is that if the first-order information is correct, then little is required of the second-order information. Specifically, the sequence of model Hessians need only be bounded.

Our theory is analogous for problem (EQC). In this case, the local model of the problem is generally taken to be a linear model of the constraints and a quadratic model of the Lagrangian. The information in the local model depends on the Lagrange multiplier estimates as well as second-order information. In this paper, we identify a way to extend the unconstrained paradigm to problem (EQC) for which the only requirement is boundedness of the sequence of model Lagrange multipliers and Hessians.

3. The SQP algorithm. The Lagrangian function $\ell : \Re^n \times \Re^m \to \Re$ associated with problem (EQC) is the function

$$\ell(x, \lambda) = f(x) + \lambda^T C(x),$$

where $\lambda = (\lambda_1, \ldots, \lambda_m)^T$ is a Lagrange multiplier vector estimate.

A common algorithm for solving problem (EQC) is the successive quadratic programming algorithm. It is an iterative procedure. At each iteration, a step s^{QP} and associated Lagrange multiplier $\Delta\lambda^{QP}$ are obtained by solving the following quadratic program:

$$(\mathrm{QP}) \equiv \begin{cases} \text{minimize} & q_c(s) = \frac{1}{2} s^T H_c s + \nabla_x \ell_c^T s + \ell_c \\ \text{subject to} & \nabla C_c^T s + C_c = 0, \end{cases}$$

where the matrix H_c is the Hessian of the Lagrangian at (x_c, λ_c) or an approximation to it.

Unfortunately, the SQP algorithm cannot be guaranteed to work without modification. There is a fundamental difficulty in the definition of the SQP step because the second-order sufficiency condition need not hold at each iteration. By this we mean that the matrix H_c need not be positive definite on the null space of ∇C_c^T; hence the QP subproblem may have no solution or many solutions. This difficulty will not arise near a solution of problem (EQC) if the standard assumptions for Newton's method hold at the solution. For this reason, the SQP algorithm usually performs very well locally. See Tapia [28] for more details.

An effective modification that deals with the lack of positive definiteness on the null space is the use of a trust-region globalization strategy. This takes us to the following section.

4. Existing trust-region algorithms for (EQC). A straightforward way to extend the trust-region idea to problem (EQC) is to add a trust-region constraint to the (QP) subproblem to restrict the size of the step. So, at each iteration, we solve the following trust-region subproblem:

$$\begin{cases} \text{minimize} & q_c(s) = \frac{1}{2} s^T H_c s + \nabla_x \ell_c^T s + \ell_c \\ \text{subject to} & \nabla C_c^T s + C_c = 0, \\ & \|s\| \leq \delta_c. \end{cases}$$

However, in this straightforward approach, observe that the trust-region constraint and the linearized constraints may be inconsistent, so that the model subproblem does not have a solution. To overcome this difficulty, two main approaches have been introduced for dealing with the case when $\{s : \nabla C_c^T s + C_c = 0\} \cap \{s : \|s\| \leq \delta_c\} = \emptyset$. They are the tangent-space approach and the full-space approach. We describe them briefly in the next section. More details can be found in Maciel [17]. See also Byrd, Schnabel, and Shultz [2]; Celis, Dennis, and Tapia [4]; Omojokun [21]; Powell and Yuan [23]; and Vardi [31], [32].

4.1. The tangent-space approach. In this approach the trial step is determined as $s_c = s_c^n + s_c^t$, where s_c^n is the quasi-normal component and s_c^t is the tangential component with respect to the null space of the constraint Jacobian. The component s_c^n is in the trust region, and ideally it truly is normal to the null space $\mathcal{N}(\nabla C_c^T)$. The substep s_c^t is the component of the step in the tangent space of the constraints given by $s_c^t = W_c \bar{s}_c^t$, with $\bar{s}_c^t \in \Re^{n-m}$ and W_c is an $n \times (n-m)$ matrix whose columns form a basis for $\mathcal{N}(\nabla C_c^T)$.

This raises two questions to be answered. We must say how to determine s_c^n, and given s_c^n we must say how to determine s_c^t. We proceed in reverse order. Given s_c^n we determine s_c^t by considering the transformed subproblem

$$\begin{cases} \text{minimize} & q_c(s^t + s_c^n) \\ \text{subject to} & \nabla C_c^T s^t = 0, \\ & \|s^t\| \leq \bar{\delta}_c, \end{cases}$$

where $\bar{\delta}_c = \sqrt{\delta_c^2 - \|s_c^n\|^2}$. We choose s_c^t by using one of the standard unconstrained trust-region trial-step selection methods on this reduced problem.

These algorithms have the trust region capability of dealing quite well with zero or negative curvature in the tangent space of the constraints. Thus, nonexistence of an SQP step at the current iterate is readily handled.

To choose s_c^n, Byrd, Schnabel, and Shultz [2] and Vardi [31],[32] suggest relaxing the linearized constraints by replacing C_c by αC_c, where $\alpha \in (0, 1]$ is chosen to ensure that the above trust-region subproblem is feasible. Thus, $s_c^n = -\alpha \nabla C_c (\nabla C_c^T \nabla C_c)^{-1} C_c$. Observe that if $\alpha = 0$ then $\nabla C_c^T s + \alpha C_c = 0$ contains $s = 0$ and hence for any $\sigma \in (0, 1]$ there is some $\alpha_\sigma \in (0, 1)$ for which $\{s : \nabla C_c^T s + \alpha_\sigma C_c = 0\} \cap \{s : \|s\| \leq \sigma \delta_c\} \neq \emptyset$.

The drawback of the above approach is that the step depends on the parameter α; it is not clear how to choose it.

Omojokun [21], used this approach to compute a trial step that does not depend on α by choosing s_c^n to be the step that solves the following problem:

$$\begin{cases} \text{minimize} & \frac{1}{2} \|\nabla C_c^T s + C_c\|^2 \\ \text{subject to} & \|s\| \leq \sigma \delta_c \end{cases}$$

for $0 < \sigma < 1$.

It might appear that Omojokun has traded the choice of α for the choice of σ, but in fact σ is easy to choose. Some nominal value like $\sigma = 0.8$ is used throughout, and the particular value of σ at a given iteration is allowed to be in some uniformly bounded strict subinterval like $(0.7, 0.9)$. This subinterval corresponds to stopping criteria on a trust-region algorithm to solve for s_c^n. See Moré [18], Moré and Sorensen [19], or Dennis and Schnabel [7].

4.2. The full-space approach. The other approach to overcoming the problem of inconsistency is the full-space approach. Algorithms based on this approach compute s_c at once in the whole $\Re^n$ space instead of considering the decomposition of the trial step. This has the advantage of avoiding the computation of a Moore–Penrose pseudoinverse solution.

The first example we know of this category of trust-region subproblems is the CDT subproblem proposed by Celis, Dennis, and Tapia [4]. Instead of considering the linearized constraint $\nabla C_c^T s + C_c = 0$, they replace it by a particular inequality: $\|\nabla C_c^T s + C_c\| \leq \theta_c$, where $\theta_c \in \Re$. The *CDT subproblem* can be written as follows:

$$\left\{\begin{array}{ll} \text{minimize} & q_c(s) \\ \text{subject to} & \|\nabla C_c^T s + C_c\| \leq \theta_c, \\ & \|s\| \leq \delta_c. \end{array}\right.$$

The key to the CDT subproblem (and its variants) is the choice of θ_c. For more details, see Williamson [33]. Celis, Dennis, and Tapia [4] choose θ_c based on a *fraction of Cauchy decrease* condition on $\|\nabla C_c^T s + C_c\|^2$. They ask the step to satisfy for some $r_1 \in (0, 1]$,

$$\|C_c\|^2 - \|C_c + \nabla C_c^T s\|^2 \geq r_1\{\|C_c\|^2 - \|\nabla C_c^T s_c^{\text{cp}} + C_c\|^2\}.$$

This can be done by choosing

$$\theta_c^2 = (\theta_c^{\text{fcd}})^2 \equiv r_1 \|\nabla C_c^T s_c^{\text{cp}} + C_c\|^2 + (1 - r_1)\|C_c\|^2, \tag{4.1}$$

where s_c^{cp} solves the problem

$$\left\{\begin{array}{lll} \text{minimize} & \frac{1}{2}\|\nabla C_c^T s + C_c\|^2 & \\ \text{subject to} & \|s\| \leq r\delta_c, & \\ & s = -t\nabla C_c C_c, & t \geq 0. \end{array}\right.$$

Note that in this case, the CDT subproblem minimizes the quadratic model of ℓ over a set of steps inside the trust region. Specifically, the set is of those steps that give at least r_1 times as much decrease in the ℓ_2-norm of the residual of the linearized constraints as does the Cauchy step.

In order to prevent the possibility of having only a single feasible point for the subproblem and so not to have a meaningful trust-region subproblem, it is suggested that $r < 1$; for instance $r = 0.8$.

5. A general trust-region algorithm. In this section we describe a very inclusive class of trust-region algorithms.

The typical form of trust-region algorithms for solving (EQC) is basically as follows: at the current point x_c with associated multiplier estimate λ_c, a step s_c is computed by solving some trust-region subproblems, and a Lagrange multiplier estimate λ_+ is obtained by using a scheme. The point x_+, where $x_+ = x_c + s_c$, is tested using some merit function to decide whether it is a better approximation to a solution $x_\star$. Such merit functions often involve a penalty parameter, which is updated using a scheme. The trust-region radius is then adjusted and a new quadratic model is formed.

In our requirements on the trust-region algorithm, the way of computing the trial steps is replaced by some conditions the steps must satisfy, and the estimates of the Lagrange multiplier vectors and the Hessian matrices need only be uniformly bounded. This allows the inclusion of a wide variety of trust-region algorithms and it is exactly in the spirit of Powell's Theorem 2.2 for unconstrained trust-region methods. In section 9, we present an example algorithm that satisfies these mild conditions.

5.1. Computing the trial steps. We first write the trial step as $s_c = s_c^t + s_c^n$, where s_c^t and s_c^n are, respectively, the tangential and a quasi-normal component. We do not require that s_c^n be normal to the tangent space.

We require that the components s_c^n and s_c^t satisfy a fraction of Cauchy decrease condition on appropriate model functions. At the current iterate, if $C_c \neq 0$, then we require that the quasi-normal component give at least as much decrease as $s_c^{\rm cp} = -{\rm n}_c^{\rm cp} \nabla C_c C_c$ on the quadratic model of the linearized constraints in a trust region of radius $r\delta_c$, where the step length ${\rm n}_c^{\rm cp}$ is given by

$$
{\rm n}_c^{\rm cp} = \begin{cases} \frac{\|\nabla C_c C_c\|^2}{\|\nabla C_c^T \nabla C_c C_c\|^2} & \text{if } \frac{\|\nabla C_c C_c\|^3}{\|\nabla C_c^T \nabla C_c C_c\|^2} \leq \hat{\delta}_c, \\ \frac{\hat{\delta}_c}{\|\nabla C_c C_c\|} & \text{otherwise,} \end{cases}
$$

where $\hat{\delta}_c = r\delta_c$ and $0 < r < 1$. In words, the step s_c^n is chosen from the set of steps that satisfy a fraction of Cauchy decrease condition on the quadratic model of the linearized constraints inside $\|s\| \leq \hat{\delta}_c$. Equivalently, s_c^n lies in the set

$$
S_c = \{s : \|s\| \leq \hat{\delta}_c\} \cap \{s : \|\nabla C_c^T s + C_c\|^2 \leq (\theta_c^{\rm fcd})^2\},
$$

where $(\theta_c^{\rm fcd})^2$ is given by (4.1). Because the quasi-normal component s_c^n is not required to be normal to the tangent space, a condition on the step is needed to ensure global convergence. In particular, the following condition is required:

$$
\|s_c^n\| \leq K_1 \|C_c\|, \tag{5.1}
$$

where K_1 is some positive constant independent of the iteration.

If s_c^n is normal to the tangent space, this condition holds (see Lemma 7.1) as long as K_1 is greater than a uniform bound on the norm of the right inverse for $\nabla C(x)^T$. When s_c^n is not normal to the tangent space, we do not suggest choosing K_1 and enforcing (5.1). Rather, we suggest (as in section 9) that (5.1) is enforced naturally by any reasonable algorithm for computing a linearly feasible point.

We deal with the quasi-normal components of the trial steps, assuming that they satisfy (5.1). We are indebted to Robert Michael Lewis for informing us of the effectiveness of this feature in the algorithm which he has implemented to solve a PDE inverse problem [6]. Specifically, this allows special linear algebra developed for simulation constraints to be used in place of prohibitively large least-squares solutions.

Now we use the quasi-normal component to pick a linear manifold $\mathcal{M}_c$, parallel to the null space of the constraints. We select the tangential component in $\mathcal{M}_c$. Let $\mathcal{M}_c = \{s : \nabla C_c^T s = \nabla C_c^T s_c^n\}$. Thus, $\mathcal{M}_c \cap \{s = s^t + s_c^n : \|s\| \leq \delta_c\} \neq \emptyset$.

Observe that in the set S_c we are taking a fraction of δ_c in order to forestall the case that $\mathcal{M}_c$ lies too close to the boundary of the trust region of radius δ_c.

On the manifold $\mathcal{M}_c$, we consider a quadratic model $q_c(s)$ of the Lagrangian function associated with problem (EQC). Then when $W_c^T \nabla q_c(s_c^n) \neq 0$, we ask the tangential component to satisfy a fraction of Cauchy decrease condition from s_c^n on $q_c(s)$ reduced to $\mathcal{M}_c$. That is $s_c = s_c^t + s_c^n \in \mathcal{G}_c \cap \mathcal{M}_c$, where

$$\mathcal{G}_c = \{s = s^t + s_c^n : \|s\| \le \delta_c, q_c(s) - q_c(s_c^n) \le \sigma[q_c(s_c^n - \mathrm{t}_c^{\mathrm{cp}} W_c W_c^T \nabla q_c(s_c^n)) - q_c(s_c^n)]\}$$

for some $\sigma > 0$ and

$$\text{(5.2)} \quad \mathrm{t}_c^{\mathrm{cp}} = \begin{cases} \frac{\|W_c^T \nabla q_c(s_c^n)\|^2}{\nabla q_c(s_c^n)^T W_c \bar{H}_c W_c^T \nabla q_c(s_c^n)} & \text{if } \frac{\|W_c^T \nabla q_c(s_c^n)\|^2 \|W_c W_c^T \nabla q_c(s_c^n)\|}{\nabla q_c(s_c^n)^T W_c \bar{H}_c W_c^T \nabla q_c(s_c^n)} \le \bar{\delta_c} \\ & \text{and } \nabla q_c(s_c^n)^T W_c \bar{H}_c W_c^T \nabla q_c(s_c^n) > 0, \\ \frac{\bar{\delta}_c}{\|W_c W_c^T \nabla q_c(s_c^n)\|} & \text{otherwise,} \end{cases}$$

where $\bar{H}_c = W_c^T H_c W_c$ is the reduced Hessian matrix and $\bar{\delta_c}$ is the maximum length of the step allowed inside the set $\mathcal{M}_c \cap \{s = s^t + s_c^n : \|s\| \le \delta_c\}$ in the negative reduced gradient direction $-W_c^T \nabla q_c(s_c^n)$.

It is easy to see that $\bar{\delta_c}$ satisfies

$$\text{(5.3)} \qquad (1 + r)\delta_c \; > \; \bar{\delta_c} \; > \; (1 - r)\delta_c.$$

We have intentionally not stated the computation of the tangential component as a trust-region subproblem. Condition 5.2 is a lopsided condition in the sense that $\bar{\delta_c}$ is direction dependent because the quasi-normal step is not the center of the natural trust region for the reduced quadratic. A better step might come from minimizing the reduced quadratic in $\mathcal{M}_c \cap \{s = s^t + s_c^n : \|s\| \le \bar{\delta_c}\}$, and an ideal step would probably come from minimizing the reduced quadratic in $\mathcal{M}_c \cap \{s = s^t + s_c^n : \|s\| \le \delta_c\}$. In any case, both result in steps that satisfy our conditions.

We have defined the tangent space Cauchy step along $-W_c^T \nabla q_c(s_c^n)$, which is the steepest descent direction for $q_c(s_c^n + W_c \bar{s}^t)$ in the ℓ_2-norm. The steepest descent direction in the $\|W_c \cdot \|$-norm would be $-[W_c W_c^T]^{-1} W_c^T \nabla q_c(s_c^n)$. Of course, as long as $[W_c W_c^T]^{-1}$ is uniformly bounded, which seems a reasonable assumption, then either step satisfies a fraction of Cauchy decrease condition with respect to the other, and our theory holds for either. We do not need this boundedness assumption for our choice of Cauchy step. For a particular application, the choice of variables may be determined by which form of the reduced problem is easiest to precondition. See the discussion after Algorithm 9.2. For the problems of interest to us $-[W_c W_c^T]^{-1} W_c^T \nabla q_c(s_c^n)$ would be an extremely expensive—or impossible—direction to compute.

5.2. Updating the model Lagrange multiplier and the model Hessian. The method for estimating the multiplier λ_c is left unspecified. We only require that the sequence of estimates $\{\lambda_k\}$ be bounded. Any approximation to the Lagrange multiplier vector that produces a bounded sequence can be used. For example, setting λ_k to a fixed vector (or even the zero vector) for all k is valid. Similarly, we require only boundedness of the sequence $\{H_k\}$ of approximate Hessians. Thus, all $H_k = 0$ is allowed. Note that, here, we are not addressing the question of the choices of the Lagrange multiplier and Hessian estimates that produce an efficient algorithm. We are addressing some weak assumptions on those estimates $\{\lambda_k\}$ and $\{H_k\}$ that produce a globally convergent algorithm. For example, our theory applies to a form of successive linear programming with an elliptical move limit.

5.3. The choice of the merit function. Let x_c be the current iterate. We need to decide if a trial step chosen to satisfy $s_c^n \in S_c$ and $s_c = s_c^n + s_c^t \in \mathcal{G}_c \cap \mathcal{M}_c$ is a *good* step; that is, we decide if the step s_c gives a new iterate x_+ that is a better approximation than x_c to a solution $x_\star$ of (EQC). In constrained optimization, the meaning of better approximation should consider improvement not only in f but also in the constraint violation $\|C\|$. The evaluation of the trial step requires the choice of a merit function, which usually involves the objective function and the constraint violations.

Here, we use the augmented Lagrangian as a merit function

$$\mathcal{L}(x, \lambda; \rho) = f(x) + \lambda^T C(x) + \rho C(x)^T C(x), \qquad \rho > 0. \tag{5.4}$$

This function has been used as a merit function in trust-region algorithms also by Celis, Dennis, and Tapia [4]; El-Alem [9], [10]; and Powell and Yuan [23].

El-Alem [10] and Powell and Yuan [23] use the formula $\lambda(x) = -(\nabla C(x)^T \nabla C(x))^{-1} \nabla C(x)^T \nabla f(x)$ for updating the Lagrange multiplier. For this particular choice of the multiplier, λ is a function of x and (5.4) is an exact penalty function. This means that if ρ is sufficiently large, then the solution to problem (EQC) is an unconstrained minimizer of the penalty function. See Fletcher [12], [13].

Celis, Dennis, and Tapia [4] and El-Alem [9], on the other hand, with a particular choice of the multiplier, have treated the multiplier as an independent parameter that really only enters in the merit function for accepting the step and updating the other parameters in the algorithm. In other words, one never explicitly uses the merit function in computing the optimization step; it is used only for evaluating the steps. The effect on the trial step computation of the multiplier estimates is in the tangential component through the estimate of the Hessian of the Lagrangian. This is a major difference between merit function roles in trust region algorithms and in line-search algorithms.

In the context of a line-search globalization strategy, Gill, Murray, Saunders, and Wright [15] and Schittkowski [24] have considered the augmented Lagrangian as a merit function but also as an objective function for choosing the step along the direction of search. They have treated the multiplier as an independent variable and proved global convergence for their algorithms.

In summary, we believe that having an exact penalty function as a merit function is, of course, a desirable property, especially in line-search algorithms. On the other hand, in practice, one never really knows whether the penalty constant has been chosen so that the exactness property holds. In [8], [9] global convergence for a particular trust-region method is shown with no assumption of exactness.

In this work, the choice of the multiplier estimate is left open and $\lambda = 0$ is allowed, in which case one is using the ℓ_2 penalty function as a merit function.

5.4. Evaluating the trial step. Let s_c be a trial step chosen to satisfy the conditions of section 5.1. We accept it if it produces sufficient improvement in the merit function. To measure this improvement, we compare the *actual reduction* and *predicted reduction* in the merit function from the current iterate x_c to the new one $x_+ = x_c + s_c$. The *actual reduction* is defined by

$$\begin{aligned} Ared_c(s_c; \rho_c) &= \mathcal{L}(x_c, \lambda_c; \rho_c) - \mathcal{L}(x_+, \lambda_+; \rho_c) \\ &= \ell(x_c, \lambda_c) - \ell(x_+, \lambda_+) + \rho_c(\|C_c\|^2 - \|C_+\|^2), \end{aligned} \tag{5.5}$$

and the *predicted reduction* is defined to be

$$Pred_c(s_c; \rho_c) = \mathcal{L}(x_c, \lambda_c; \rho_c) - \mathcal{Q}(s_c, \Delta\lambda_c; \rho_c), \tag{5.6}$$

where $\mathcal{Q}(s_c, \Delta\lambda_c; \rho_c) = \ell(x_c, \lambda_c) + \nabla_x \ell(x_c, \lambda_c)^T s_c + \frac{1}{2} s_c^T H_c s_c + (\Delta\lambda_c)^T (C_c + \nabla C_c^T s_c) + \rho_c(\|C_c + \nabla C_c^T s_c\|^2)$.

We accept the step and set $x_+ = x_c + s_c$ if $\frac{Ared_c}{Pred_c} \geq \eta_1$, where $\eta_1 \in (0, 1)$ is a fixed constant. A typical value for η_1 is 10^{-4}.

5.5. Updating the trust-region radius. The strategy that we follow for updating the trust-region radius is based on the standard rules for the unconstrained case. More details can be found in Dennis and Schnabel [7] or Fletcher [14]. However, for our global convergence theory we use a modification due to Zhang, Kim, and Lasdon [34] (see also El Hallabi and Tapia [11]). This modification is of no importance in practice; it is merely an analytic formality. At the beginning we set constants $\delta_{\max} \geq \delta_{\min}$ and each time we find an acceptable step, we start the next iteration with a value of $\delta_+ \geq \delta_{\min}$. In short, δ_c can be reduced below $\delta_{\min}$ while seeking an acceptable step, but $\delta_+ \geq \delta_{\min}$ must hold at the beginning of the next iteration after finding an acceptable step. The following is the scheme for evaluating the step and updating the trust-region radius.

ALGORITHM 5.1. **Evaluating the step and updating the trust-region radius**

Given the constants: $0 < \alpha_1 < 1$, $\alpha_2 > 1$ *and* $0 < \eta_1 < \eta_2 < 1$ *and* $\delta_{\max} \geq \delta_c \geq \delta_{\min} > 0$.

```
While $\frac{Ared_c}{Pred_c} < \eta_1$    (* e.g., $\eta_1 = 10^{-4}$ *)
    Do not accept the step.
    Reduce the trust-region radius: $\delta_c \leftarrow \alpha_1 \|s_c\|$    (* e.g., $\alpha_1 = 0.5$ *), and
    compute a new trial step $s_c$.
End while
If $\eta_1 \leq \frac{Ared_c}{Pred_c} < \eta_2$    (* e.g., $\eta_2 = 0.5$ *)    then
    Accept the step: $x_+ = x_c + s_c$.
    Set the trust-region radius: $\delta_+ = \max\{\delta_c, \delta_{\min}\}$.
End if
If $\frac{Ared_c}{Pred_c} \geq \eta_2$ then
    Accept the step: $x_+ = x_c + s_c$.
    Increase the trust-region radius:
(5.7)        $\delta_+ = \min\{\delta_{\max}, \max\{\delta_{\min}, \alpha_2 \delta_c\}\}$
    (* e.g., $\alpha_2 = 2$ *).
End if
```

It is worth noting that in practice one might have another branch in which some $\eta_{\frac{3}{2}} \in (\eta_1, \eta_2)$ is used to reduce the trust-region radius if $\eta_1 \leq \frac{Ared_c}{Pred_c} \leq \eta_{\frac{3}{2}}$. A typical value for $\eta_{\frac{3}{2}}$ is .1, and the motivation is to try to avoid the expense of a next unacceptable trial step. Another modification sometimes used in practice is to allow internal doubling. This can be viewed loosely as letting α_2 in (5.7) depend on $\frac{Ared_c}{Pred_c}$. See Dennis and Schnabel, [7, p. 144]. The present analysis would allow these niceties, but to avoid further complication, we do not include them here. Observe that in (5.5) and (5.6) we have expressed the quantities *Ared* and *Pred* as functions of ρ. Thus, although ρ_c does not effect the choice of the trial step s_c, we need to determine ρ_c *before* deciding the acceptance of the step s_c. The right choice of the penalty parameter is one of the most important issues for algorithms that use the augmented Lagrangian as a merit function. This takes us to the following section.

5.6. The penalty parameter. Numerical experience with nonlinear programming algorithms that use the augmented Lagrangian as a merit function has shown that good performance of the algorithm depends on keeping the penalty parameter as small as possible. See Gill, Murray, and Wright [16]. On the other hand, global convergence theories developed by El-Alem [8], [9] and Powell and Yuan [23] require that the sequence $\{\rho_k\}$ be nondecreasing. El-Alem [8] requires that ρ be chosen so that the predicted decrease in the merit function is at least as much as the decrease in $\|\nabla C_c^T s + C_c\|^2$.

We consider as an update formula for the penalty parameter El-Alem's scheme given in [9], since it ensures that the merit function is predicted to decrease at each iteration by at least a fraction of Cauchy decrease in the quadratic model of the constraints. This indicates compatibility with the fraction of Cauchy decrease conditions imposed on the trial steps. In addition, good performance was reported when implementing this scheme. See Williamson [33]. It can be stated as follows:

Algorithm 5.2. **Updating the penalty parameter**

```
1. Initialization
   Set $\rho_{-1} = 1$ and choose a small constant $\beta > 0$.
2. At the current iterate $x_c$, after $s_c$ has been chosen:
   Compute
   $Pred_c(s_c;\rho_-) = q_c(0) - q_c(s_c) - \Delta\lambda_c^T(C_c + \nabla C_c^T s_c) + \rho_-[\|C_c\|^2 - \|\nabla C_c^T s_c + C_c\|^2]$.
      If $Pred_c(s_c;\rho_-) \ge \frac{\rho_-}{2}[\|C_c\|^2 - \|\nabla C_c^T s_c + C_c\|^2]$,
         then set $\rho_c = \rho_-$,
      else set $\rho_c = \bar{\rho}_c + \beta$, where
         $\bar{\rho}_c = \frac{2[q_c(s_c) - q_c(0) + \Delta\lambda_c^T(C_c + \nabla C_c^T s_c)]}{\|C_c\|^2 - \|\nabla C_c^T s_c + C_c\|^2}$.
      End if
```

The initial choice of the penalty parameter ρ_{-1} is arbitrary. However, it should be chosen consistent with the scale of the problem. Here, we take $\rho_{-1} = 1$ for convenience.

An immediate consequence of the above algorithm is that at the current iteration we have

$$Pred_c(s_c;\rho_c) \ge \frac{\rho_c}{2}[\|C_c\|^2 - \|C_c + \nabla C_c^T s_c\|^2]. \tag{5.8}$$

5.7. Termination of the algorithm. We use first-order necessary conditions for problem (EQC) to terminate the algorithm. The algorithm is terminated if $\|W_c^T \nabla_x \ell_c\| + \|C_c\| \le \varepsilon_{tol}$, where $\varepsilon_{tol} > 0$ is a prespecified constant and W_c is a matrix with columns forming a basis for the null space. We require that $\{W_k\}$ be uniformly bounded in norm for all k.

6. Statement of the algorithm.

We present a formal description of our class of nonlinear programming algorithms.

Algorithm 6.1. **The NLP algorithm.**

```
step 0. (Initialization)
   Given $x_0$, $\lambda_0$, compute $W_0$.
   Choose $\delta_0$, $\delta_{\min}$, $\delta_{\max}$, and $\varepsilon_{tol} > 0$.
   Set $\rho_{-1} = 1$ and $\beta > 0$.
step 1. (Test for convergence)
      If $\|W_c^T \nabla_x \ell(x_c)\| + \|C(x_c)\| \le \varepsilon_{tol}$
```

```
            then terminate.
        End if
    step 2. (Compute a trial step)
        If $x_c$ is feasible then
            (a) find a step $s_c^t$ that satisfies a fraction of Cauchy decrease condition
                on the quadratic model $q_c(s)$ of the Lagrangian around $x_c$. (This
                might be done by solving a trust-region subproblem since $s_c^n = 0$ is
                available. See section 5.1)
            (b) Set $s_c = s_c^t$.
        else        (* $C(x_c) \neq 0$ *)
            (a) Compute a quasi-normal step $s_c^n$ that satisfies a fraction of Cauchy
                decrease condition on the square norm quadratic model of the lin-
                earized constraints. (See section 5.1)
            (b) If $W_c^T \nabla q(s_c^n) = 0$
                    then set $s_c^t = 0$
                else find $s_c^t$ that satisfies a fraction of Cauchy decrease condition
                    on the quadratic model $q_c(s_c^n + s)$ from $s_c^n$. (Perhaps not by
                    solving a specific trust-region subproblem. See section 5.1)
                End if
            (c) Set $s_c = s_c^n + s_c^t$.
        End if
    step 3. (Update $\lambda_c$)
            Choose an estimate $\lambda_+$ of the Lagrange multiplier vector.
            Set $\Delta\lambda_c = \lambda_+ - \lambda_c$.
    step 4. (Update the penalty parameter)
            Update $\rho_-$ to obtain $\rho_c$ by using Algorithm 5.2.
    step 5. (Evaluate the step)
            Compute

            $Ared_c(s_c;\rho_c) = \ell(x_c,\lambda_c) - \ell(x_+,\lambda_+) + \rho_c(\|C_c\|^2 - \|C_+\|^2).$

            Evaluate the step and update the trust-region radius by using Algorithm 5.1.
            If the step is accepted
                then update $H_c$ and go to step 1.
            else
                go to step 2.
            End if
```

The above represents a typical trust-region algorithm for solving problem (EQC). We leave the way of computing the trial steps undefined. This will allow the inclusion of a wide variety of trial step calculation techniques. For similar reasons we left the way of updating the Lagrange multiplier vector and the Hessian matrix undefined.

In the next two sections we prove global convergence of the above algorithm class.

7. The global convergence theory. Before beginning our global convergence theory, let us give an overview of the steps that comprise this theory.

The trial step is chosen to satisfy a sufficient predicted decrease condition, the fraction of Cauchy decrease. Note that in our algorithm, we assume that the tangential and the quasi-normal components of any trial step each satisfy this condition. In Lemma 7.2, we will express this in a technical form similar to inequality (2.2).

The definition of predicted reduction is shown to give an approximation to the actual reduction that is accurate to within the square of the trial step length times the penalty parameter. This is proved in Lemma 7.5. However, we emphasize again that the step is not chosen to maximize the predicted decrease.

We introduce some notation for the quantities computed during the trial steps. We have not introduced this notation up to now because it obscures the simplicity of the algorithm. However, in the analysis that follows we need to show some properties of every trial step, not just the successful steps $\{s_k\}$. Therefore, let δ_k^i, s_k^i, and ρ_k^i denote the quantities set by Algorithm 6.1 as it searches for an acceptable step. Thus, $\delta_k^0 = \delta_k$ at the first trial step of the kth iteration, s_k^0 is set by the first time through step 2, and ρ_k^0 is set using $\rho_k^{-1} = \rho_{k-1}$ the first time through step 4. If the trial step s_k^i is acceptable, then $s_k = s_k^i$, $\rho_k = \rho_k^i$, and δ_k^i is updated to become δ_{k+1}. In short, the algorithm is simpler to explain and code if one counts only successful steps. However, for the analysis, one needs a way to refer unambiguously to all the trial steps.

The model Lagrange multipliers also may depend on i. However, to keep the notation as simple as possible, we do not make this dependence explicit.

The penalty parameters ρ_k^i are shown to be bounded for $\epsilon_{tol} > 0$ as long as the algorithm does not terminate. The technique is to prove that at any iteration k in which the penalty parameter is increased we have that the product of the penalty parameter ρ_k^i and the trust-region radius δ_k^i are bounded by a constant that does not depend on k or i (this is done in Lemma 7.10), and the sequence of the trust-region radii δ_k^i is bounded away from zero (this is shown in Lemma 7.11). The proofs show the crucial role that is played by setting the trust region to be no smaller than $\delta_{\min}$ after every acceptable step. See section 5.5. Finally, under the assumption that the algorithm does not terminate, the penalty parameter ρ_k is shown to be bounded. The proof is given in Lemma 7.12.

The algorithm is shown to be well defined in the sense that at a given iterate it either terminates or it finds an acceptable step after finitely many trials. This result is proved in Theorem 8.1. Using the above results and Theorem 8.1, the trust-region radius is shown to be bounded away from zero. The proof is given in Lemma 8.2.

Finally, in Theorem 8.4, it is shown that for any $\varepsilon_{tol} > 0$, the algorithm always terminates, i.e., the termination condition of the algorithm is met after finitely many iterations.

7.1. The problem assumptions. We start by stating the assumptions under which global convergence is proved for Algorithm 6.1. Assumptions A1–A5 (see below) are used by Byrd, Schnabel, and Shultz [2]; El-Alem [8], [9], [10], and Powell and Yuan [23], and their particular choices of Lagrange multiplier vectors satisfy A6.

Let the sequence of iterates $\{x_k\}$ generated by the algorithm satisfy

A1. For all k, x_k and $x_k + s_k^i \in \Omega$, where Ω is a convex set of $\Re^n$.
A2. $f, C \in C^2(\Omega)$.
A3. $\operatorname{rank}(\nabla C(x)) = m$ for all $x \in \Omega$.
A4. $f(x), \nabla f(x)$, $\nabla^2 f(x)$, $C(x)$, $\nabla C(x)$, $(\nabla C(x)^T \nabla C(x))^{-1}$, $W(x)$, and $\nabla^2 c_i(x)$ for $i = 1, \ldots, m$ are all uniformly bounded in Ω.
A5. The matrices $H_k, k = 1, 2, \ldots$ are uniformly bounded.
A6. The vectors $\lambda_k, k = 1, 2, \ldots$ are uniformly bounded.

Assumption A4 means that for all $x \in \Omega$, there exist positive constants ν, ν_0, ν_1, ν_2, ν_3, ν_4, ν_5, and ν_6 such that $\|f(x)\| \leq \nu$, $\|\nabla f(x)\| \leq \nu_0$, $\|C(x)\| \leq \nu_1$, $\|\nabla C(x)\| \leq \nu_2$, $\|(\nabla C(x)^T \nabla C(x))^{-1}\| \leq \nu_3$, $\|\nabla^2 f(x)\| \leq \nu_4$, $\|\nabla^2 c_i(x)\| \leq \nu_5$ for all $i = 1, \ldots, m$, and $\|W(x)\| \leq \nu_6$.

An immediate consequence of assumptions A4 and A5 is the existence of a constant $\nu_7 > 0$ that does not depend on k such that $\|H_k\| \leq \nu_7$, $\|W_k^T H_k\| \leq \nu_7$, and

$\|W_k^T H_k W_k\| \leq \nu_7$.

Assumption A6 means that for all $x \in \Omega$ there exists a constant $\nu_8 > 0$ that does not depend on k, such that $\|\lambda_k\| \leq \nu_8$.

The following three subsections are devoted to presenting lemmas needed to prove global convergence.

7.2. Properties of the trial step. The following lemma shows that condition (5.1) holds for the normal component ${s_k^i}^n$ of s_k^i when it is truly normal to the tangent space.

LEMMA 7.1. *At the current iterate x_k, let the trial step component ${s_k^i}^n$ actually be normal to the tangent space. Under the problem assumptions, there exists a constant $K_1 > 0$ independent of the iterates such that*

$$\|{s_k^i}^n\| \leq K_1 \|C_k\|. \tag{7.1}$$

Proof. Because ${s_k^i}^n$ is actually normal to the tangent space, we have

$$\begin{aligned} \|{s_k^i}^n\| &= \|\nabla C_k (\nabla C_k^T \nabla C_k)^{-1} \nabla C_k^T s_k^i\| \\ &= \|\nabla C_k (\nabla C_k^T \nabla C_k)^{-1} (C_k + \nabla C_k^T s_k^i - C_k)\| \\ &\leq \|\nabla C_k (\nabla C_k^T \nabla C_k)^{-1}\| [\|C_k + \nabla C_k^T s_k^i\| + \|C_k\|]. \end{aligned}$$

Now, using the fact that $\|C_k + \nabla C_k^T s_k^i\| \leq \|C_k\|$ we have

$$\|{s_k^i}^n\| \leq 2 \cdot \|\nabla C_k (\nabla C_k^T \nabla C_k)^{-1}\| \cdot \|C_k\|.$$

The rest follows from the problem assumptions. □

The following lemma expresses in a workable form the pair of fraction of Cauchy decrease conditions imposed on the trial steps.

LEMMA 7.2. *If the trial steps satisfy the conditions given in step 2 of Algorithm 6.1, then under the problem assumptions there exist positive constants K_2, K_3, and K_4 independent of the iterates such that*

$$\|C_k\|^2 - \|C_k + \nabla C_k^T {s_k^i}^n\|^2 \geq K_2 \|C_k\| \min\{K_3 \|C_k\|, r\delta_k^i\} \tag{7.2}$$

and

$$\begin{aligned} & q_k({s_k^i}^n) - q_k(s_k^i) \\ & \qquad \geq \frac{\sigma}{2} \|W_k^T \nabla q_k({s_k^i}^n)\| \min\left\{ \frac{1-r}{\nu_6} \delta_k^i, K_4 \|W_k^T \nabla q_k({s_k^i}^n)\| \right\}. \end{aligned} \tag{7.3}$$

Proof. The proof is an application of Lemma 2.1 to the two subproblems, followed by a use of the problem assumptions and (5.3). □

Now we deal with the trial steps assuming that they satisfy inequalities (7.2) and (7.3). In what follows, we use implicitly the identity $\nabla C_k^T {s_k^i}^n = \nabla C_k^T s_k^i$.

LEMMA 7.3. *Under the problem assumptions, there exists a constant $K_5 > 0$ independent of the iterates such that*

$$q_k(0) - q_k({s_k^i}^n) - \Delta{\lambda_k}^T (C_k + \nabla C_k^T {s_k^i}^n) \geq -K_5 \|C_k\|. \tag{7.4}$$

Proof. Consider

$$\begin{aligned} q_k(0) - q_k({s_k^i}^n) &= -\nabla_x \ell_k^T {s_k^i}^n - \frac{1}{2} ({s_k^i}^n)^T H_k {s_k^i}^n \\ &\geq -\|\nabla_x \ell_k\| \, \|{s_k^i}^n\| - \frac{1}{2} \|H_k\| \, \|{s_k^i}^n\|^2 \\ &= -\left(\|\nabla_x \ell_k\| + \frac{1}{2} \|H_k\| \, \|{s_k^i}^n\| \right) \|{s_k^i}^n\|. \end{aligned}$$

Using (5.1), the fact that $\|{s_k^i}^n\| < \delta_{\max}$, λ_k and $\Delta\lambda_k$ are bounded, and $\|C_k + \nabla C_k^T s_k^i\| \leq \|C_k\|$, and the problem assumptions, we have

$$q_k(0) - q_k({s_k^i}^n) - {\Delta\lambda_k}^T(C_k + \nabla C_k^T s_k^i) \geq -K_5\|C_k\|,$$

and we obtain the desired result. □

The following lemma gives an upper bound on the difference between the actual reduction and the predicted reduction.

LEMMA 7.4. *Under the problem assumptions, there exist positive constants* K_6, K_7, *and* K_8, *independent of* k, *such that*

$$|Ared_k(s_k^i;\rho_k^i) - Pred_k(s_k^i;\rho_k^i)| \leq K_6\|s_k^i\|^2 + K_7\rho_k^i\|s_k^i\|^3 + K_8\rho_k^i\|s_k^i\|^2\|C_k\|. \tag{7.5}$$

Proof. The proof follows directly from El-Alem [9]. □

If the penalty parameter is uniformly bounded, the next lemma shows that the predicted reduction provides an approximation to the actual reduction that is accurate to within the square of the step length.

LEMMA 7.5. *Under the problem assumptions, there exists a constant* $K_9 > 0$ *that does not depend on* k *such that*

$$|Ared_k(s_k^i;\rho_k^i) - Pred_k(s_k^i;\rho_k^i)| \leq K_9\rho_k^i\|s_k^i\|^2. \tag{7.6}$$

Proof. The proof follows directly from the above lemma and the fact that $\|s_k^i\|$ and $\|C_k\|$ are bounded. □

7.3. The decrease in the model. This section deals with the predicted decrease in the merit function produced by the trial step. We start with a lemma.

LEMMA 7.6. *Let* s_k^i *be generated by Algorithm* 6.1. *Then under the problem assumptions for any positive* ρ *the predicted decrease in the merit function satisfies*

$$\begin{aligned} Pred_k(s_k^i;\rho) \geq{}& \frac{\sigma}{2}\|W_k^T\nabla q_k({s_k^i}^n)\| \min\left\{K_4\|W_k^T\nabla q_k({s_k^i}^n)\|\ ,\ \frac{1-r}{\nu_6}\delta_k^i\right\} \\ & - K_5\|C_k\| + \rho[\|C_k\|^2 - \|\nabla C_k^T s_k^i + C_k\|^2], \end{aligned} \tag{7.7}$$

where K_5 *is as in Lemma* 7.3.

Proof. We have

$$\begin{aligned} Pred_k(s_k^i;\rho) &= q_k(0) - q_k(s_k^i) - {\Delta\lambda_k}^T(C_k + \nabla C_k^T s_k^i) \\ &\quad +\rho[\|C_k\|^2 - \|\nabla C_k^T s_k^i + C_k\|^2] \\ &= (q_k({s_k^i}^n) - q_k(s_k^i)) \\ &\quad +(q_k(0) - q_k({s_k^i}^n)) - {\Delta\lambda_k}^T(C_k + \nabla C_k^T s_k^i) \\ &\quad + \rho[\|C_k\|^2 - \|\nabla C_k^T s_k^i + C_k\|^2]. \end{aligned}$$

From (7.3) and Lemma 7.3 we have

$$\begin{aligned} Pred_k(s_k^i;\rho) \geq{}& \frac{\sigma}{2}\|W_k^T\nabla q_k({s_k^i}^n)\| \min\left\{K_4\|W_k^T\nabla q_k({s_k^i}^n)\|\ ,\ \frac{1-r}{\nu_6}\delta_k^i\right\} \\ & - K_5\|C_k\| + \rho[\|C_k\|^2 - \|\nabla C_k^T s_k^i + C_k\|^2]. \end{aligned}$$

Hence the result is established. □

If x_k is feasible, then the predicted reduction does not depend on ρ_k, so we take ρ_k as the penalty parameter from the previous iteration. The question now is how near to feasibility must an iterate be in order that the penalty parameter need not be increased. The answer is given by the following lemma.

LEMMA 7.7. *Assume that the algorithm does not terminate at the current iterate. If $\|C_k\| \leq \alpha\delta_k^i$, where α satisfies*

$$\alpha \leq \min\left\{\frac{\varepsilon_{tol}}{3\delta_{\max}}\ ,\ \frac{\varepsilon_{tol}}{3\nu_7 K_1\delta_{\max}}\ ,\ \frac{\sigma\varepsilon_{tol}}{12K_5}\min\left\{\frac{K_4\varepsilon_{tol}}{3\delta_{\max}},\ \frac{1-r}{\nu_6}\right\}\right\}, \tag{7.8}$$

then for any positive ρ,

$$\begin{aligned} Pred_k(s_k^i;\rho) \geq \frac{\sigma}{4}\|W_k^T\nabla q_k({s_k^i}^n)\| \min\left\{K_4\|W_k^T\nabla q_k({s_k^i}^n)\|\ ,\ \frac{1-r}{\nu_6}\delta_k^i\right\} \\ + \rho[\|C_k\|^2 - \|\nabla C_k^T s_k^i + C_k\|^2]. \end{aligned} \tag{7.9}$$

Proof. If the algorithm does not terminate at x_k, then $\|W_k^T\nabla_x\ell_k\| + \|C_k\| > \varepsilon_{tol}$, and since $\|C_k\| \leq \alpha\delta_k^i$ with $\alpha \leq \frac{\varepsilon_{tol}}{3\delta_{\max}}$, therefore, $\|C_k\| \leq \frac{\varepsilon_{tol}}{3}$ and the reduced gradient satisfies $\|W_k^T\nabla_x\ell_k\| > \frac{2}{3}\varepsilon_{tol}$. Now,

$$\begin{aligned} \|W_k^T\nabla q_k({s_k^i}^n)\| &= \|W_k^T(\nabla_x\ell_k + H_k{s_k^i}^n)\| \\ &\geq \|W_k^T\nabla_x\ell_k\| - \|W_k^TH_k{s_k^i}^n\| \\ &\geq \frac{2}{3}\varepsilon_{tol} - \nu_7K_1\|C_k\| \geq \frac{2}{3}\varepsilon_{tol} - \nu_7K_1\alpha\delta_k^i. \end{aligned}$$

But since $\alpha \leq \frac{\varepsilon_{tol}}{3\nu_7K_1\delta_{\max}}$, it follows that

$$\|W_k^T\nabla q_k({s_k^i}^n)\| \geq \frac{1}{3}\varepsilon_{tol}.$$

From Lemma 7.6, we have

$$\begin{aligned} Pred_k(s_k^i;\rho) \geq \frac{\sigma}{2}\|W_k^T\nabla q_k({s_k^i}^n)\| \min\left\{\frac{1-r}{\nu_6}\delta_k^i\ ,\ K_4\|W_k^T\nabla q_k({s_k^i}^n)\|\right\} \\ - K_5\|C_k\| + \rho[\|C_k\|^2 - \|\nabla C_k^T s_k^i + C_k\|^2]. \end{aligned}$$

Since $\|W_k^T\nabla q({s_k^i}^n)\| > \frac{1}{3}\varepsilon_{tol}$, we have

$$\begin{aligned} Pred_k(s_k^i;\rho) \geq \frac{\sigma}{4}\|W_k^T\nabla q_k({s_k^i}^n)\| \min\left\{\frac{1-r}{\nu_6}\delta_k^i\ ,\ K_4\|W_k^T\nabla q_k({s_k^i}^n)\|\right\} \\ + \frac{\sigma}{12}\varepsilon_{tol}\min\left\{\frac{1-r}{\nu_6}\delta_k^i\ ,\ \frac{\varepsilon_{tol}K_4}{3}\right\} \\ - K_5\alpha\delta_k^i + \rho[\|C_k\|^2 - \|\nabla C_k^T s_k^i + C_k\|^2]. \end{aligned}$$

Thus

$$\begin{aligned} Pred_k(s_k^i;\rho) \geq \frac{\sigma}{4}\|W_k^T\nabla q_k({s_k^i}^n)\| \min\left\{\frac{1-r}{\nu_6}\delta_k^i\ ,\ K_4\|W_k^T\nabla q_k({s_k^i}^n)\|\right\} \\ + \frac{\sigma\varepsilon_{tol}\delta_k^i}{12}\min\left\{\frac{1-r}{\nu_6}\ ,\ \frac{\varepsilon_{tol}K_4}{3\delta_{\max}}\right\} \\ - K_5\alpha\delta_k^i + \rho[\|C_k\|^2 - \|\nabla C_k^T s_k^i + C_k\|^2], \end{aligned}$$

and since

$$\alpha \leq \frac{\sigma \varepsilon_{tol}}{12 K_5} \min\left\{ \frac{K_4 \varepsilon_{tol}}{3\delta_{\max}} \,,\, \frac{1-r}{\nu_6} \right\},$$

we have

$$\begin{aligned} Pred_k(s_k^i;\rho) \geq \frac{\sigma}{4} \| W_k^T \nabla q_k({s_k^i}^n) \| \min\left\{ K_4 \, \| W_k^T \nabla q_k({s_k^i}^n) \| \,,\, \frac{1-r}{\nu_6} \delta_k^i \right\} \\ + \rho[\|C_k\|^2 - \|\nabla C_k^T s_k^i + C_k\|^2]. \end{aligned}$$

This completes the proof. □

Inequality (7.9) with $\rho = \rho_k^{i-1}$ guarantees that if the algorithm does not terminate and if $\|C_k\| \leq \alpha \delta_k^i$, then the penalty parameter at the current trial step does not need to be increased in step 2 of Algorithm 6.1. This is equivalent to saying that any increases in the penalty parameter occur only when $\|C_k\| > \alpha \delta_k^i$.

Lemma 7.8. *Given $\varepsilon_{tol} > 0$, there exists $K_{10} > 0$, which depends on ε_{tol} but not on k or i such that at any trial step s_k^i of iteration k at which the algorithm does not terminate and $\|C_k\| \leq \alpha \delta_k^i$, where α is as in Lemma 7.7, the following inequality holds:*

$$Pred_k(s_k^i;\rho_k^i) \geq K_{10} \delta_k^i. \tag{7.10}$$

Proof. Since the algorithm does not terminate and $\|C_k\| \leq \alpha \delta_k^i$, where α is as in (7.8), then from (7.9) and using a similar argument as in Lemma 7.7, we can write

$$Pred_k(s_k^i;\rho_k^i) \geq \frac{\sigma \varepsilon_{tol}}{12} \min\left\{ \frac{1-r}{\nu_6} \delta_k^i, \, \frac{K_4 \varepsilon_{tol}}{3} \right\} \geq \frac{\sigma \varepsilon_{tol}}{12} \min\left\{ \frac{1-r}{\nu_6}, \, \frac{K_4 \varepsilon_{tol}}{3\delta_{\max}} \right\} \delta_k^i.$$

Defining

$$K_{10} = \frac{\sigma \varepsilon_{tol}}{12} \min\left\{ \frac{1-r}{\nu_6}, \, \frac{K_4 \varepsilon_{tol}}{3\delta_{\max}} \right\},$$

we have $Pred_k(s_k^i;\rho_k^i) \geq K_{10} \delta_k^i$ and this is the desired result. □

In the next section we will discuss the role of the penalty parameter in the global convergence of the nonlinear programming algorithm.

7.4. The behavior of the penalty parameter. In this section we discuss the behavior of the penalty parameter. The crucial result here is that the sequence $\{\delta_k^i\}$ of trust-region radii is bounded away from zero at those iterations for which the penalty parameter is increased at some trial step. This allows us to conclude under the nontermination hypothesis that the sequence $\{\rho_k^i\}$ of penalty parameters is bounded.

According to the rule for updating the penalty parameter, we use the penalty parameter from the previous trial step if the amount of predicted decrease with the old penalty parameter is at least a fraction of the decrease in the quadratic model of the linearized constraints; that is, if

$$Pred_k(s_k^i;\rho_k^{i-1}) \geq \frac{\rho_k^{i-1}}{2} [\|C_k\|^2 - \|C_k + \nabla C_k^T s_k^i\|^2], \tag{7.11}$$

then $\rho_k^i = \rho_k^{i-1}$. Otherwise, we use $\rho_k^i = \bar{\rho_k}^i + \beta$, which enforces (5.8). See section 5.6.

Lemma 7.9. *Let $\{\rho_k^i\}$ be the sequence of penalty parameters generated by the algorithm. Then*

1. *$\{\rho_k^i\}$ forms a nondecreasing sequence;*
2. *if the penalty parameter is increased, it will increase by at least β;*
3. *if the penalty parameter is not increased, then inequality* (7.11) *will hold.*

Proof. The proof is straightforward. □

LEMMA 7.10. *Let k, i be any pair of indices such that ρ_k^i is increased at the ith trial step of the kth iteration. If the algorithm does not terminate at x_k, then there exists $K_{11} > 0$ which depends on ε_{tol} but does not depend on k or i such that for every $j \geq i$,*

$$\rho_k^j \delta_k^j \leq K_{11}. \tag{7.12}$$

Proof. If ρ_k^i is increased at the ith trial step of the kth iteration, then it is updated by the rule

$$\rho_k^i = \frac{2[q_k(s_k^i) - q_k(0) + \Delta\lambda_k^T(C_k + \nabla C_k^T s_k^i)]}{\|C_k\|^2 - \|C_k + \nabla C_k^T s_k^i\|^2} + \beta.$$

Hence,

$$\begin{aligned}
\frac{\rho_k^i}{2}[\|C_k\|^2 - \|C_k + \nabla C_k^T {s_k^i}^n\|^2] &= [q_k(s_k^i) - q_k(0)] + {\Delta\lambda_k}^T(C_k + \nabla C_k^T {s_k^i}^n) \\
&\quad + \frac{\beta}{2}[\|C_k\|^2 - \|C_k + \nabla C_k^T {s_k^i}^n\|^2] \\
&= [q_k(s_k^i) - q_k({s_k^i}^n)] \\
&\quad + [q_k({s_k^i}^n) - q_k(0)] + \Delta\lambda_k^T(C_k + \nabla C_k^T {s_k^i}^n) \\
&\quad + \frac{\beta}{2}[-2(\nabla C_k C_k)^T {s_k^i}^n - \|\nabla C_k^T {s_k^i}^n\|^2].
\end{aligned}$$

Applying (7.2) to the left-hand side and (7.3) and Lemma 7.3 to the right-hand side, we can obtain the following:

$$\begin{aligned}
&\frac{\rho_k^i K_2}{2}\|C_k\| \min \{\, r\delta_k^i \,,\ K_3\|C_k\,\|\,\} \\
&\qquad \leq -\frac{\sigma}{2}\|W_k^T \nabla q_k({s_k^i}^n)\| \min \left\{ K_4\|W_k^T\nabla q_k({s_k^i}^n)\| \,,\ \frac{1-r}{\nu_6}\delta_k^i \right\} \\
&\qquad\quad + K_5\|C_k\| - \beta(\nabla C_k C_k)^T {s_k^i}^n - \frac{\beta}{2}\|\nabla C_k^T {s_k^i}^n\|^2 \\
&\qquad \leq K_5\|C_k\| - \beta(\nabla C_k C_k)^T {s_k^i}^n \\
&\qquad \leq K_5\|C_k\| + \beta\|\nabla C_k\|\,\|C_k\|\,\|{s_k^i}^n\| \\
&\qquad \leq (K_5 + \beta\|\nabla C_k\|\,\|{s_k^i}^n\|)\|C_k\|.
\end{aligned}$$

Then,

$$\rho_k^i \frac{K_2}{2} \min \{r\delta_k^i \,,\ K_3\|C_k\|\} \leq K_5 + \beta\nu_2\delta_{\max}.$$

Since at the current trial step the penalty parameter increases, from Lemma 7.7 we have $\|C_k\| > \alpha\delta_k^i$. Hence

$$\rho_k^i \frac{K_2}{2} \min \{r\delta_k^i \,,\ K_3\alpha\delta_k^i\} \leq K_5 + \beta\nu_2\delta_{\max}$$

and

$$\rho_k^i \delta_k^i \le \frac{2K_5 + 2\beta\nu_2\delta_{\max}}{K_2 \min\{r\ ,\ K_3\alpha\}}.$$

Now if $j \ge i$, then $\delta_k^j \le \delta_k^i$. Assume without loss of generality that $\rho_k^j = \rho_k^i$, i.e., that the ith trial step was the most recent increase with respect to j. Then $\rho_k^j \delta_k^j \le \rho_k^i \delta_k^i$, and defining

$$K_{11} = \frac{2K_5 + 2\beta\nu_2\delta_{\max}}{K_2 \min\{r\ ,\ K_3\alpha\}},$$

we obtain the desired result. □

The following lemma gives a lower bound for the sequence $\{\delta_k^i\}$ for those iterates at which the algorithm does not terminate and the penalty parameter is increased. In the next section, we do away with the assumption that the penalty parameter is increased.

LEMMA 7.11. *Let the penalty parameter be increased at the ith trial step of the kth iteration. Then under the problem assumptions, if the algorithm does not terminate, there exists $\tilde{\delta}$, which depends on ε_{tol} but does not depend on the iterates, such that*

$$\delta_k^i \ge \tilde{\delta}. \tag{7.13}$$

Proof. To begin, we note that if $i = 0$, i.e., we are at the first trial step of iteration k, then by Algorithm 5.1, δ_k cannot have become smaller than $\delta_{\min}$ during the course of the iteration. Thus, we can restrict our attention to the case where $i \ge 1$.

Our proof will consist of showing the existence of $\tilde{\delta}$ such that $\delta_k^i \ge \tilde{\delta}$ whether or not s_k^i is acceptable. Remember that for all the rejected trial steps we have $\delta_k^{j+1} = \alpha_1 \|s_k^j\|$.

We consider two cases:

(i) $\|C_k\| > \alpha\delta_k^j$ for all $j = 0, \ldots, i$.

(ii) $\|C_k\| > \alpha\delta_k^j$ does not hold for some j between 0 and i.

(i) Consider the case where the constraint violation $\|C_k\| > \alpha\delta_k^j$ for all $j = 0, \ldots, i$. We have from Lemma 7.5,

$$|Ared_k(s_k^j;\rho_k^j) - Pred_k(s_k^j;\rho_k^j)| \le K_9 \rho_k^j \|s_k^j\|^2.$$

Now since $\|C_k\| > \alpha\delta_k^j$, then from the way of updating ρ_k^j and using inequality (7.2), we have

$$\begin{aligned} Pred_k(s_k^j;\rho_k^j) &\ge \frac{\rho_k^j}{2}[\|C_k\|^2 - \|C_k + \nabla C_k^T s_k^j\|^2] \\ &\ge \frac{\rho_k^j}{2} K_2 \|C_k\| \min\{K_3\alpha,\ r\}\delta_k^j. \end{aligned}$$

Hence

$$\frac{|Ared_k(s_k^j;\rho_k^j) - Pred_k(s_k^j;\rho_k^j)|}{Pred_k(s_k^j;\rho_k^j)} \le \frac{2K_9\|s_k^j\|}{K_2\|C_k\| \min\{K_3\alpha,\ r\}}. \tag{7.14}$$

Since all the steps s_k^j for $j = 0, \ldots, i-1$ are rejected, it must be the case that

$$1 - \eta_1 < \left| \frac{Ared_k(s_k^j;\rho_k^j)}{Pred_k(s_k^j;\rho_k^j)} - 1 \right|. \tag{7.15}$$

So from (7.14) and (7.15), we have

$$\|s_k^j\| \geq \frac{(1-\eta_1)K_2 \mathrm{min}\{\alpha K_3,\, r\}}{2K_9}\|C_k\| \text{ for all } j = 0, \ldots, i-1. \tag{7.16}$$

Since $\delta_k^i = \alpha_1 \|s_k^{i-1}\|$ and since $\|C_k\| > \alpha\delta_k^0$, it follows that

$$\delta_k^i \;=\; \alpha_1\|s_k^{i-1}\| \geq \alpha_1 \left[\frac{(1-\eta_1)K_2 \min\{\alpha K_3,\, r\}}{2K_9}\right]\alpha\delta_k^0. \tag{7.17}$$

Now according to the rule for updating the trust-region radius, we know that $\delta_k^0 \geq \delta_{\min}$. Then

$$\delta_k^i \geq \frac{\alpha_1(1-\eta_1)K_2 \mathrm{min}\{\alpha K_3,\, r\}}{2K_9}\alpha\delta_{\min} = K_{12}. \tag{7.18}$$

(ii) If $\|C_k\| > \alpha\delta_k^j$ does not hold for all $j = 0, \ldots, i$, then there exists a largest index l, $0 \leq l < i$ such that $\|C_k\| \leq \alpha\delta_k^l$ holds.

If $i = l+1$, then from the way of updating the trust-region radius, $\delta_k^i = \alpha_1\|s_k^l\|$. On the other hand, if $i \neq l+1$, since $\|C_k\| > \alpha\delta_k^j$ for all $j = l+1, \ldots, i$, then from (7.16) we have

$$\|s_k^j\| \geq \frac{(1-\eta_1)K_2 \mathrm{min}\{\alpha K_3,\, r\}}{2K_9}\|C_k\| \text{ for all } j = l+1, \ldots, i-1.$$

Now because s_k^{i-1} and s_k^{l+1} are rejected trial steps and using $\|C_k\| > \alpha\delta_k^{l+1}$, we can write

$$\begin{aligned}
\delta_k^i &= \alpha_1\|s_k^{i-1}\| \\
&\geq \alpha_1 \frac{(1-\eta_1)K_2 \mathrm{min}\{\alpha K_3,\, r\}}{2K_9}\|C_k\| \\
&\geq \alpha_1\alpha \frac{(1-\eta_1)K_2 \mathrm{min}\{\alpha K_3,\, r\}}{2K_9}\delta_k^{l+1} \\
&\geq \alpha_1^2\alpha \frac{(1-\eta_1)K_2 \mathrm{min}\{\alpha K_3,\, r\}}{2K_9}\|s_k^l\|. \qquad (7.19)
\end{aligned}$$

So if we set

$$K_{13} = \min\left\{\alpha_1, \alpha_1^2\alpha\frac{(1-\eta_1)K_2 \mathrm{min}\{\alpha K_3,\, r\}}{2K_9}\right\},$$

then we have

$$\delta_k^i \geq K_{13}\|s_k^l\|. \tag{7.20}$$

Therefore, using the above inequality and Lemma 7.10,

$$\rho_k^l\,\|s_k^l\| \leq \rho_k^i \frac{\delta_k^i}{K_{13}} \leq \frac{K_{11}}{K_{13}} = K_{14}.$$

From (7.5) we have

$$|Ared_k(s_k^l;\rho_k^l) - Pred_k(s_k^l;\rho_k^l)| \leq [K_6 + (K_7 + \alpha K_8)\rho_k^l\|s_k^l\|]\|s_k^l\|\delta_k^l.$$

Therefore,

$$|Ared_k(s_k^l;\rho_k^l) - Pred_k(s_k^l;\rho_k^l)| \le [K_6 + (K_7 + \alpha K_8)K_{14}]\|s_k^l\|\delta_k^l. \tag{7.21}$$

Also, since $\|C_k\| \le \alpha\delta_k^l$, then from Lemma 7.8 we have

$$Pred_k(s_k^l;\rho_k^l) \ge K_{10}\delta_k^l. \tag{7.22}$$

Using (7.21), (7.22), and the fact that s_k^l is rejected, we obtain

$$1 - \eta_1 < \left|\frac{Ared_k(s_k^l;\rho_k^l)}{Pred_k(s_k^l;\rho_k^l)} - 1\right| \le \frac{[K_6 + K_7K_{14} + \alpha K_8K_{14}]\|s_k^l\|}{K_{10}}.$$

Hence

$$\|s_k^l\| \ge \frac{(1-\eta_1)K_{10}}{K_6 + K_7K_{14} + \alpha K_8K_{14}}. \tag{7.23}$$

Now, using (7.20) and (7.23), we obtain the bound

$$\delta_k^i \ge K_{13}\frac{(1-\eta_1)K_{10}}{K_6 + K_7K_{14} + \alpha K_8K_{14}} = K_{15}.$$

Defining

$$\tilde{\delta} = \min\{\delta_{\min}, K_{12}, K_{15}\}$$

we obtain the desired bound. □

Now we can show that the nondecreasing sequence of penalty parameters generated by the nonlinear programming Algorithm 6.1 is bounded.

LEMMA 7.12. *Under the problem assumptions, if the algorithm does not terminate then there is some $\rho^\star$, which depends on ε_{tol}, for which*

$$\lim_{k\to\infty} \rho_k = \rho^\star < \infty. \tag{7.24}$$

Furthermore, there exists some index k_ρ such that $\rho_k = \rho^\star$ for every $k \ge k_\rho$.

Proof. We need to show that $\rho^\star \ge \rho_k^i$ for all pairs k, i. Clearly, it suffices to consider the sequence ρ_k^i of different ρ_k's, where the double index k, i means that the penalty constant was increased to be ρ_k^i at the ith trial step of the kth iteration. Thus, there may be no terms or more than one term for a given k. Then from Lemmas 7.10 and 7.11, we have

$$\rho_k^i \le \frac{K_{11}}{\delta_k^i} \le \frac{K_{11}}{\tilde{\delta}}.$$

Therefore $\{\rho_k\}$ is a bounded sequence, and since it is nondecreasing, there exists $\rho^\star < \infty$ such that

$$\lim_{k\to\infty} \rho_k = \rho^\star.$$

Now since the existence of $\rho^\star$ ensures that ρ_k is bounded, and since we know that every increase is by at least β, there must be at most finitely many increases, and the proof is complete. □

This last result and the following one play crucial roles in the proof of the global convergence of Algorithm 6.1.

LEMMA 7.13. *Under the problem assumptions, if the algorithm does not terminate then the augmented Lagrangian is bounded on Ω.*

Proof. The proof is immediate from the boundedness of the penalty constant and the problem assumptions. □

8. The main global convergence results.

This section is devoted to presenting our main global convergence results. We start with the finite termination theorem, where we show that the general nonlinear programming algorithm is well defined. In section 8.2, we present more properties of the trust-region radius sequence generated by the algorithm under the assumption that it does not terminate. In section 8.3, we prove global convergence of our algorithm.

8.1. The finite termination theorem.

The following lemma shows that the nonlinear programming Algorithm 6.1 is well defined in the sense that at each iteration we can find an acceptable step after a finite number of trial step computations or, equivalently, trust-region reductions. This allows us to drop the consideration of trial steps and only consider "successful trial steps," $\{s_k\}$.

THEOREM 8.1. *Under the problem assumptions, unless some iterate x_k satisfies the termination condition of Algorithm 6.1, an acceptable step from x_k is found after finitely many trial steps.*

Proof. The proof follows from Theorem 5.1 of El-Alem [9]. □

LEMMA 8.2. *Under the problem assumptions, assume that the algorithm does not terminate. Then there exists $\delta_\star > 0$, which depends on ε_{tol} but does not depend on the iterates, such that for all k, i,*

$$\delta_k^i \geq \delta_\star. \tag{8.1}$$

Proof. The proof is very similar to the proof of Lemma 7.11.

To begin, we note that if the first trial step is acceptable, then by Algorithm 5.1, δ_k cannot have become smaller than $\delta_{\min}$ during the course of the iteration. Thus, we can restrict our attention to the case where there is at least one unsuccessful trial step. Let us assume then that we have j unsuccessful steps. Our proof consists of showing the existence of $\tilde{\delta}$ such that $\delta_k^j \geq \tilde{\delta}$ whether or not s_k^j is acceptable, i.e., is s_k. Remember that for all the rejected trial steps we have $\delta_k^{j+1} = \alpha_1 \|s_k^j\| < \delta_k^j$.

We consider two cases:

(i) $\|C_k\| > \alpha\delta_k^i$ for all $i = 0, \ldots, j$.

(ii) $\|C_k\| > \alpha\delta_k^i$ does not hold for some i such that $0 < i \leq j$.

The proof of (i) is exactly the same as in the proof of Lemma 7.11, so let us proceed to (ii).

(ii) Now if $\|C_k\| > \alpha\delta_k^i$ does not hold for all $i = 0, \ldots, j$, as in Lemma 7.11, we let l be the largest index such that $\|C_k\| \leq \alpha\delta_k^l$ holds. Now, since $\|C_k\| \leq \alpha\delta_k^i$ for all $i \leq l$, it follows from Lemma 7.8 that for all such i, $Pred_k(s_k^i; \rho_k^i) \geq K_{10}\delta_k^i$. Furthermore, from Lemma 7.5, $|Ared_k(s_k^i; \rho_k^i) - Pred_k(s_k^i; \rho_k^i)| \leq K_9 \rho_k^i \|s_k^i\|^2$, and because the step s_k^i is an unacceptable step, we have

$$1 - \eta_1 < \left| \frac{Ared_k(s_k^i; \rho_k^i)}{Pred_k(s_k^i; \rho_k^i)} - 1 \right| \leq \frac{K_9 \rho_k^i \|s_k^i\|^2}{K_{10}\delta_k^i} \leq \frac{K_9 \rho^\star \|s_k^i\|}{K_{10}}.$$

The above inequality implies that for all $i \leq l$,

$$\delta_k^i \geq \|s_k^i\| \geq \frac{(1-\eta_1)K_{10}}{K_9 \rho^\star}.$$

For all $i > l$ we have from (7.20) and the above inequality,

$$\delta_k^i \geq K_{13}\|s_k^l\| \geq K_{13}\frac{(1-\eta_1)K_{10}}{K_9 \rho^\star}.$$

It remains only to collect the constants as in Lemma 7.11. □

8.2. The global convergence results. Now we present our main global convergence result. Namely, under the problem assumptions, the general nonlinear programming algorithm generates a sequence of iterates $\{x_k\}$, which has at least a subsequence that converges to a stationary point of problem (EQC). We start with a proof that if the algorithm does not terminate it will converge to a feasible point.

THEOREM 8.3. *Under the problem assumptions, if there exists $\varepsilon_{tol} > 0$ such that*

$$\|W_k^T \nabla_x \ell_k\| + \|C_k\| > \varepsilon_{tol}$$

for all k, then

$$\lim_{k \to \infty} \|C_k\| = 0. \tag{8.2}$$

Proof. We prove (8.2) by contradiction. We begin by assuming that there exists an infinite sequence of indices $\{k_j\}$ such that $\|C_k\|$ is bounded away from zero for all $k \in \{k_j\}$. This implies that there exists $\tau > 0$ such that for all $k \in \{k_j\}$, $\|C_k\| \geq \tau$. Now for each $k_j \geq k_\rho$, where k_ρ is as in Lemma 7.12, we have from (5.8) and (7.2) that

$$\begin{aligned} Pred_{k_j} &\geq \frac{\rho_{k_j}}{2}[\|C_{k_j}\|^2 - \|C_{k_j} + \nabla C_{k_j}^T s_{k_j}\|^2] \\ &\geq \frac{K_2 \rho^\star}{2}\|C_{k_j}\| \min\{K_3\|C_{k_j}\|, r\delta_{k_j}\} \\ &\geq \frac{K_2 \rho^\star \tau}{2} \min\{K_3 \tau, r\delta_\star\} = K_{16} > 0. \end{aligned}$$

Remember that we are only looking at successful steps at this point in the analysis, so

$$\mathcal{L}_{k_j} - \mathcal{L}_{k_j+1} = Ared_{k_j} \geq \eta_1 Pred_{k_j} \geq \eta_1 K_{16} > 0. \tag{8.3}$$

Since $\{\mathcal{L}_k\}$ is bounded below, a contradiction arises if we let k_j go to infinity. □

THEOREM 8.4. *Under the problem assumptions, given any $\varepsilon_{tol} > 0$, the algorithm terminates because*

$$\|W_k^T \nabla_x l_k\| + \|C_k\| < \varepsilon_{tol}. \tag{8.4}$$

Proof. Notice that if we suppose that the algorithm does not terminate and that some subsequence of $\{\|W_k^T \nabla_x \ell_k\|\}$ converges to zero, then nontermination is immediately contradicted by Theorem 8.3.

So, let us suppose that $\|W_k^T \nabla_x \ell_k\| \geq \tau_1$, for some $\tau_1 > 0$. Since $\|C_k\|$ goes to zero by Theorem 8.3 and the sequence of trust-region radii is bounded below by $\delta_\star$, there exists an index $N_1 > k_\rho$ such that for all $k \geq N_1$, $\|C_k\| \leq \alpha\delta_\star \leq \alpha\delta_k$, with α as in (7.8). Therefore, by Lemma 7.8 with the i taken so that $s_k^i = s_k$ was the successful step and by Lemma 8.2, we have again an infinite sequence of steps in which the actual decrease in $\mathcal{L}$ is at least $\eta_1 K_{10} \delta_\star$. This contradicts the boundedness of $\mathcal{L}$ and completes the proof. □

9. An example algorithm. In this section we propose, as an example, a particular step choice algorithm for step 2 of Algorithm 6.1. We include different ways for computing s_c^n according to the dimension of the problem. We then state the complete algorithm for finding the trial step. Finally, in section 9.5 we show that the trial step generated by this algorithm satisfies the pair of fraction of Cauchy decrease conditions and (5.1).

The step choice algorithm we propose in this section is based on a conjugate directions method. It can be viewed as a generalization of the Steihaug–Toint dogleg algorithm for the unconstrained problem. This algorithm is much like a trust-region version of an algorithm due to Nash [20].

9.1. The Steihaug–Toint dogleg algorithm. This section describes the generalized dogleg algorithm introduced by Steihaug [27] and Toint [30] for approximating the solution of problem (TRS) (see section 2). This algorithm is based on the linear conjugate gradient method.

Algorithm 9.1. **Steihaug–Toint dogleg algorithm for (TRS)**

```
Given $x_c$, $\delta_c$, and $\xi_c \le \xi < 1$.
    step 0: (Initialization)
        Set $\hat{s}_0 = 0$.
        Set $r_0 = -(G_c\hat{s}_0 + \nabla f_c)$.
        Set $d_0 = r_0$.
        Set $i = 0$.
    step 1: Compute $\gamma_i = d_i^T G_c d_i$.
        If $\gamma_i > 0$ then go to step 2 .
        Otherwise    (* $d_i$ is a direction of negative or zero curvature *)
        compute $\tau > 0$ such that $\|\hat{s}_i + \tau d_i\| = \delta_c$.
        Set $s_c = \hat{s}_i + \tau d_i$ and terminate.
    step 2: Compute $\alpha_i = \frac{\|r_i\|^2}{\gamma_i}$.
        Set $\hat{s}_{i+1} = \hat{s}_i + \alpha_i d_i$.
        If $\|\hat{s}_i\| < \delta_c$ go to step 3:
        Otherwise    (* the step is too long, take the dogleg step *)
        compute $\tau > 0$ such that $\|\hat{s}_i + \tau d_i\| = \delta_c$.
        Set $s_c = \hat{s}_i + \tau d_i$ and terminate.
    step 3: Compute $r_{i+1} = r_i - \alpha_i G_c d_i$.
        If  $\frac{\|r_{i+1}\|}{\|r_0\|} \le \xi_c$,    then
        set $s_c = \hat{s}_{i+1}$ and terminate.
    step 4: Compute $\beta_i = \frac{\|r_{i+1}\|^2}{\|r_i\|^2}$.
        Set $d_{i+1} = r_{i+1} + \beta_i d_i$.
        Set $i = i + 1$ and go to step 1:
```

The Steihaug–Toint dogleg algorithm is well known for being suitable for large-scale unconstrained problems. It can be used in the framework of any general trust-region algorithm for solving problem (UCMIN).

9.2. Computing a quasi-normal component. We start our proposed step choice algorithm by finding a quasi-normal component s_c^n of the trial step. This step must satisfy a fraction of Cauchy decrease condition on the constraint norm inside the inner trust region. It determines for us which translate of the null space of the constraint Jacobian will be the one in which we choose the next iterate.

We repeat, because it is so important, that we do not require that s_c^n be normal to the tangent space but only that it satisfy (5.1). In fact, we will see below that one way we might choose the quasi-normal component is by finding a linearly feasible point and just scaling it back onto the inner trust region.

9.2.1. Via Craig's algorithm. First we note that we can solve for a linearly feasible point by using Craig's algorithm on the underdetermined linear system $\nabla C_c^T s +$

$C_c = 0$ (see [5]). Craig's algorithm consists of making the transformation $s = \nabla C_c y$ and applying the standard conjugate gradient algorithm to the following $m \times m$ linear system:

$$\nabla C_c^T \nabla C_c y + C_c = 0.$$

This implies that

$$s_c^{\mathrm{craig}} = s_c^{\mathrm{mn}} = -\nabla C_c (\nabla C_c^T \nabla C_c)^{-1} C_c.$$

Furthermore, the result is the Moore–Penrose pseudoinverse constraint normal, and it requires no more than m iterations. Preconditioning is very important of course, but how to do it certainly depends on the particular application.

Therefore, we can find the step s_c^n by a Steihaug–Toint version of Craig's algorithm in the inner trust region of radius $r\delta_c$. In this algorithm, iterates are generated until we find the desired constraint normal s_c^{mn} such that $\|s_c^{\mathrm{mn}}\| \le r\delta_c$ or until s_j^{craig} and s_{j+1}^{craig} straddle the $r\delta_c$ trust-region boundary. For the first case, we set $s_c^n = s_c^{\mathrm{mn}}$. For the second case, we choose the dogleg step: $s_c^{\mathrm{dog}} \in [s_j^{\mathrm{craig}}, s_{j+1}^{\mathrm{craig}}] \cap \{s : \|s\| = r\delta_c\}$ and set $s_c^n = s_c^{\mathrm{dog}}$.

It is not difficult to prove that each Craig iterate is the ℓ_2 projection of the origin onto the subspace of the tangent space spanned by the steps up to that point and that each $\{s_j^{\mathrm{craig}}\}$ satisfies (5.1). Now, the Craig steps may not give monotone increasing ℓ_2 length, so a more aggressive strategy that works perfectly well with our theory is to take the last pair of Craig iterates that straddle the trust-region boundary. In either case, by convexity, s_c^{dog} also satisfies (5.1). Furthermore, it is clear that $s_c^n = s_c^{\mathrm{dog}}$ satisfies the fraction of Cauchy decrease condition required by step 2 of Algorithm 6.1.

9.2.2. Via a linearly feasible point. There are some problems for which Craig's method might be too slow and too hard to precondition to use the "inner Steihaug–Toint" algorithm given above. Or someone might prefer to do an implementation that computes a linearly feasible point s_c^{lf} either by Craig's method or by some special application-dependent methods. When this is the case, s_c^n can be taken to be the projection of s_c^{lf} back onto the inner trust region. If s_c^{lf} satisfies (5.1), then so does s_c^n.

Suppose we have any linearly feasible point s_c^{lf} that satisfies (5.1). Then, if it is inside the inner trust region, we can take s_c^n to be that point, and it clearly satisfies the fraction of Cauchy decrease condition required by step 2 of Algorithm 6.1. If $\|s_c^{\mathrm{lf}}\| \ge r\delta_c$, then we take

$$s_c^n = \frac{r\delta_c}{\|s_c^{\mathrm{lf}}\|} \cdot s_c^{\mathrm{lf}} \ .$$

A classical mathematical programming way to compute a linearly feasible point that encompasses some special purpose methods we have seen for certain inverse problems is as follows. Divide s into so-called basic and nonbasic components. Let us assume that we have done so, and using column pivoting we write ∇C^T as $\nabla C^T = [B|N]$, where B is a nonsingular matrix corresponding to the basic components of s. This corresponds to $W_c = [\begin{smallmatrix} -B_c^{-1}N_c \\ I_{n-m} \end{smallmatrix}]$. Now since

$$\nabla C_c^T s = B_c s_B + N_c s_N = -C_c,$$

we have

$$s_B = -B_c^{-1}(C_c + N_c s_N),$$

and then if we choose $s_N = 0$ and $s_B = -B_c^{-1}C_c$, a feasible point will be

$$s_c^{\mathrm{lf}} = (s_B, s_N)^T = (-B_c^{-1}C_c, 0)^T.$$

As long as $\{\|B_k^{-1}\|\}$ is uniformly bounded by some constant γ_*, s_c^{lf} satisfies (5.1) where the constant here is γ_*. This is a standard assumption for important classes of discretized optimal control problems, though it is stronger than our assumption that $[\nabla C(x_c)^T \nabla C(x_c)]^{-1}$ is uniformly bounded.

9.3. Computing the tangential component. We now assume that we have the quasi-normal component step s_c^n. We start the process of computing the tangent space component s_c^t by formatting the basis matrix $W_c \in \Re^{n\times(n-m)}$. The columns of W_c form a basis to the null space of the constraints $\mathcal{N}(\nabla C_c^T)$.

We then transfer the constrained problem into an unconstrained trust-region problem of dimension $n-m$, in the following form:

$$\begin{cases} \text{minimize} & \frac{1}{2}\bar{s}^{tT}\bar{H}_c\bar{s}^t + \nabla q_c(s_c^n)^T W_c \bar{s}^t + q(s_c^n) \\ \text{subject to} & \|W_c\bar{s}^t + s_c^n\| \leq \delta_c, \end{cases}$$

where $\bar{s}_c^t \in R^{n-m}$, and set $s_c^t = W_c\bar{s}_c^t$. The step s_c^t is the component in the tangent space of the constraints and the matrix $\bar{H}_c = W_c^T H_c W_c \in \Re^{(n-m)\times(n-m)}$ is the reduced Hessian matrix. Now we use the Steihaug–Toint algorithm to determine $\bar{s}_c^t$ such that $\|W_c\bar{s}^t + s_c^n\| \leq \delta_c$.

The complete algorithm for finding the trial step is presented in the following section.

9.4. Conjugate reduced gradient algorithm for EQC. Here we write, in more detail, the example algorithm for computing a trial step.

Algorithm 9.2. **The CRG step choice algorithm**

```
Given $x_c \in \Re^n$, $\delta_c > 0$, and $\xi_c \leq \xi < 1$.
  I. FEASIBILITY:
      (1) If $x_c$ is feasible go to II.
      (2) Determine $s_c^n$. (* Use, for example, $s_c^n = s_c^{\mathrm{dog}}$ or $s_c^n = \frac{r\delta}{\|s_c^{\mathrm{lf}}\|} s_c^{\mathrm{lf}}$ and
          $s_c^{\mathrm{lf}} = (-B_c^{-1}C_c, 0)^T$. *)
  II. MINIMIZATION:
      (* Find $s_c$ by applying the CRG/Steihaug–Toint algorithm, to

          $\begin{cases} \text{minimize} & q_c(s) \\ \text{subject to} & \nabla C_c^T(s - s_c^n) = 0 \\ & \|s\| \leq \delta_c. \end{cases}$

      starting from $s = s_c^n$ *)
        step 0: (Initialization)
            Set $\hat{s}_0 = s_c^n$.
            Set $r_0 = -W_c^T(H_c s_c^n + \nabla_x \ell_c)$.
            Set $d_0 = r_0$.
            Set $i = 0$.
```

```
step 1: Compute $\gamma_i = d_i^T H_c d_i$.
    If $\gamma_i > 0$ then go to step 2:,
    otherwise    (* $d_i$ is a direction of negative or zero curvature *)
    compute $\tau > 0$ such that $\|\hat{s}_i + \tau d_i\| = \delta_c$.
    Set $s_c = \hat{s}_i + \tau d_i$ and terminate.
step 2: Compute $\alpha_i = \frac{\|r_i\|^2}{\gamma_i}$.
    Set $\hat{s}_{i+1} = \hat{s}_i + \alpha_i d_i$.
    If $\|\hat{s}_i\| < \delta_c$ go to step 3:,
    otherwise    (* the step is too long, take the dogleg step *)
    compute $\tau > 0$ such that $\|\hat{s}_i + \tau d_i\| = \delta_c$.
    Set $s_c = \hat{s}_i + \tau d_i$ and terminate.
step 3: Compute $r_{i+1} = r_i - \alpha_i W_c^T H_c d_i$.
    If  $\frac{\|r_{i+1}\|}{\|r_0\|} \le \xi_c$,    then
    set $s_c = \hat{s}_{i+1}$ and terminate.
step 4: Compute $\beta_i = \frac{\|r_{i+1}\|^2}{\|r_i\|^2}$.
    Set $d_{i+1} = r_{i+1} + \beta_i d_i$.
    Set $i = i + 1$ and go to step 1:
```

It is worth noting here that this way of computing the tangent step does not have the property that once a step goes outside the trust region it cannot come back in if the cg iteration were continued. This means that the relaxed SQP step might lie inside the trust region, but the algorithm above might not return this more desirable step if the gradient scale and trust-region scale are inconsistent.

It would be better otherwise, of course, but the steps given here lead to convergence, and we hope that near the solution, when it becomes important to take SQP steps, the trust region will be large enough to compensate for the difference in shape. If the implementer wanted to be more aggressive, there are various ways to deal with this situation that fit our theory. For example, we could take the dogleg step based on the last time the cg iteration leaves the trust region rather than the first. Our concern here is to prove convergence theorems for the weakest conditions on the algorithm and to show that reasonable algorithms satisfy those conditions, not to advocate particular implementation details of no consequence to the theory.

9.5. Sufficient decrease by the steps. In this section we show that the conjugate reduced gradient algorithm produces steps that satisfy the conditions we impose on the steps in step 2 of Algorithm 6.1. In particular, we show that both the quasi-normal and the tangential components of the trial steps satisfy their respective fraction of Cauchy decrease conditions.

The following lemma gives a bound on the reducer matrix W_c. The proof is straightforward, so we omit it.

LEMMA 9.3. *Under the problem assumptions, if there is a uniform bound on the matrix $B(x)^{-1}$, then the reducer matrix*

$$W(x) = \begin{bmatrix} -B(x)^{-1}N(x) \\ I_{n-m} \end{bmatrix}$$

is bounded for all $x \in \Omega$.

The following lemma shows that the quasi-normal component s_c^n satisfies a fraction of Cauchy decrease condition on the quadratic model of the linearized constraints.

LEMMA 9.4. *Let s_c be a step generated by Algorithm 9.2 at the current iterate. Then s_c satisfies a fraction of Cauchy decrease condition on the quadratic model of* *the linearized constraints, i.e.,*

$$\|C_c\|^2 - \|C_c + \nabla C_c^T s_c\|^2 \geq K_2 \|C_c\| \min\{r\delta_c \ , \ K_3 \|C_c\|\}, \tag{9.1}$$

where K_2 and K_3 are constants independent of the iterates.

Proof. Suppose that we are applying Craig's algorithm to find s_c^n. Let $\{s_1, s_2, \ldots\}$ be the sequence of iterates generated by the algorithm, hence for all i.

$$s_i = arg\,min\{\|\nabla C_c^T s + C_c\|, \, s \in span\{p_1, \ldots, p_i\}\}.$$

Assume that $\|s_i\| \leq r\delta_c$ and $\|s_{i+1}\| \geq r\delta_c$. Therefore

$$s_c^{\text{dog}} = \alpha s_i + (1-\alpha) s_{i+1} \quad \text{ with } \quad \alpha \in [0,1].$$

It is easy to see that

$$\|\nabla C_c^T s_i + C_c\| \leq \|\nabla C_c^T s_c^{\text{cp}} + C_c\|$$

and

$$\|\nabla C_c^T s_{i+1} + C_c\| \leq \|\nabla C_c^T s_c^{\text{cp}} + C_c\|.$$

By convexity,

$$\|\nabla C_c^T s_c^{\text{dog}} + C_c\| \leq \|\nabla C_c^T s_c^{\text{cp}} + C_c\|.$$

Thus,

$$\|C_c\|^2 - \|C_c + \nabla C_c^T s_c^{\text{dog}}\|^2 \geq \|C_c\|^2 - \|C_c + \nabla C_c^T s_c^{\text{cp}}\|^2.$$

Thus we can apply Lemma 2.1.

Now suppose that s_c^n is given by $s_c^n = \gamma_c s_c^{\text{lf}}$, with $\gamma_c = \frac{r\delta_c}{\|s_c^{\text{lf}}\|}$ when $\|s_c^{\text{lf}}\| > r\delta_c$ and $\gamma_c = 1$ otherwise. When $\gamma_c = 1$, we have

$$\|C_c\|^2 - \|\nabla C_c^T s_c^n + C_c\|^2 = \|C_c\|^2 - \|\nabla C_c^T s_c^{\text{lf}} + C_c\|^2 = \|C_c\|^2.$$

When $\gamma_c < 1$, we have

$$\begin{aligned} \|C_c\|^2 - \|C_c + \nabla C_c^T s_c^n\|^2 &= \|C_c\|^2 - \|C_c + \gamma_c \nabla C_c^T s_c^{\text{lf}}\|^2 \\ &\geq \|C_c\|^2 - [(1-\gamma_c) \ \|C_c\| + \gamma_c \ \|C_c + \nabla C_c^T s_c^{\text{lf}}\|]^2 \\ &= [1 - (1-\gamma_c)^2] \ \|C_c\|^2 \geq \gamma_c \|C_c\|^2. \end{aligned}$$

The desired result follows from the definition of s_c^{lf} and Lemma 9.3. □

The following lemma shows that the null-space component s_c^t satisfies a fraction of Cauchy decrease condition on the quadratic model of the Lagrangian.

LEMMA 9.5. *Let s_c be a trial step generated by the algorithm. Then under the problem assumptions, there exists a positive constant K_4 which does not depend on x_c such that*

$$q_c(s_c^n) - q_c(s_c) \geq \frac{\sigma}{2} \|W_c^T \nabla q_c(s_c^n)\| \min\left\{K_4 \|W_c^T \nabla q_c(s_c^n)\| \ , \ \frac{(1-r)}{\nu_6}\delta_c\right\}.$$

Proof. Since we are solving the reduced problem

$$\begin{cases} \text{minimize} & \frac{1}{2}\bar{s}^{tT} \bar{H}_c \bar{s}^t + \nabla q_c(s_c^n)^T W_c \bar{s}^t + q(s_c^n) \\ \text{subject to} & \|W_c \bar{s}^t + s_c^n\| \leq \delta_c, \end{cases}$$

which is an unconstrained trust-region subproblem, the proof is immediate from Theorem 2.5 of Steihaug [27] followed by the use of the problem assumptions and Lemma 9.3. □

We state the following lemma here for completeness.

LEMMA 9.6. *The quasi-normal component computed by our proposed step choice algorithm satisfies*

$$\|s_c^n\| \leq K_1 \|C_c\|,$$

where K_1 is a positive constant independent of c.

Proof. The proof is given with the discussion of how to compute a quasi-normal component. See section 9.2. □

10. Discussion and concluding remarks.

We have established a global convergence theory for a broad class of nonlinear programming algorithms for the smooth problem with equality constraints. The class includes algorithms based on the full-space approach and the tangent-space approach. The family is characterized by generating steps that satisfy very mild conditions on the normal and tangential components. The normal component satisfies a fraction of Cauchy decrease condition on the quadratic model of the linearized constraints and the tangential component satisfies a fraction of Cauchy decrease condition on the quadratic model of the Lagrangian function associated with the problem, reduced to the tangent space of the constraints. Of course the step, which is the sum of these components, satisfies both conditions.

The augmented Lagrangian was chosen as a merit function. The scheme for updating the penalty parameter is the one proposed by El-Alem [9] since it predicts that the merit function is decreased at each iteration by at least a fraction of Cauchy decrease on the quadratic model of the linearized constraints. This indicates compatibility with the fraction of Cauchy decrease conditions imposed on the trial steps.

In presenting the algorithm, we have left open the way of computing the trial steps to satisfy the double fraction of Cauchy decrease condition. This allows the inclusion of a wide variety of trial step calculation techniques. For the same reason we have left unspecified the way of approximating the Lagrange multiplier vector and the Hessian matrix.

With respect to the trial steps, we have suggested an algorithm of the class that should work quite well for large problems. The algorithm is a generalization of the Steihaug–Toint dogleg algorithm for the unconstrained case. This algorithm was one we had in mind as motivation for the convergence theory.

The least-squares or projection formula can be used as a scheme for estimating the multiplier since it fits the condition imposed on the multiplier updating scheme. Namely, under the standard assumptions, it produces bounded multipliers for the local models. For large problems, $\lambda = -B^{-1}\nabla_B f$ is likely to be a much preferable formula because of the cost of the least-squares solution. Furthermore, this matches better with the reducer matrix W, especially for problems where B can be easily identified; see Dennis and Lewis [6]. In either case, the uniform boundedness of $\{\lambda_k\}$ follows from the problem assumptions.

The exact Hessian matrix perhaps can be gotten by using automatic differentiation or an adjoint integration approach. See Bischof et al. [1]. However, an approximation to the Hessian of the Lagrangian can be used. Also, for example, setting H_k to a fixed matrix (e.g., $H_k = 0$) for all k is valid. The question of how to use a secant approximation of the Hessian of the Lagrangian in order to produce a more efficient algorithm is a research topic. We believe that Tapia [29] will be of considerable value here.

A related question that has to be looked at is the search for preconditioners to produce more efficient algorithms. We believe that the reducer matrix W should play a role in that search; see Dennis and Lewis [6].

This theory is developed for the equality constrained case, but it can be applied to the general case by one of the strategies known as EQP and IQP. Here, we mean that in the EQP strategy the choice of the active set is made outside the algorithm that determines the step, whereas in the IQP strategy, that choice is made inside the procedure that determines the step. Since the active set may change at each iteration, the choice of the submatrix B, will be strongly affected. Certainly, this is an important topic that deserves to be investigated.

Acknowledgments. We wish to thank Richard Byrd for many helpful comments and the referees for pointing out many unclear points, which we hope have been clarified. We thank Robert Michael Lewis and the referees for pointing out the importance of dealing with the quasi-normal component. We especially thank Luis Vicente and Doug Moore for careful and insightful reading.

REFERENCES

[1] C. Bischof, A. Carle, G. Corliss, A. Griewank, and P. Hovland, *Adifor – generating derivative codes from fortran programs*, Sci. Programming, 1 (1992), pp. 11–29.

[2] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, *A trust region algorithm for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 24 (1987), pp. 1152–1170.

[3] R. Carter, *Multi-Model Algorithms for Optimization*, Ph.D. thesis, Department of Mathematical Sciences, Rice University, Houston, TX, 1986.

[4] M. R. Celis, J. E. Dennis Jr., and R. A. Tapia, *A trust region strategy for nonlinear equality constrained optimization*, in Numerical Optimization 1984, SIAM, Philadelphia, PA, 1985.

[5] E. Craig, *The n-step iteration procedures*, J. Math. Physics, 34 (1955), pp. 64–73.

[6] J. E. Dennis Jr. and R. M. Lewis, *A Comparison of Nonlinear Programming Approaches to an Elliptic Inverse Problem and a New Domain Decomposition Approach*, Tech. report, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1994, in preparation.

[7] J. E. Dennis Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983. Russian edition, Mir Publishing Office, Moscow, 1988, O. Burdakov, translator.

[8] M. M. El-Alem, *A Global Convergence Theory for a Class of Trust Region Algorithms for Constrained Optimization*, Ph.D. thesis, Department of Mathematical Sciences, Rice University, Houston, TX, 1988.

[9] M. M. El-Alem, *A global convergence theory for the Celis-Dennis-Tapia trust region algorithm for constrained optimization*, SIAM J. Numer. Anal., 28 (1991), pp. 266–290.

[10] M. M. El-Alem, *A robust trust-region algorithm with nonmonotonic penalty parameter scheme for constrained optimization*, SIAM J. Optim., 5 (1995), pp. 348–378.

[11] M. El-Hallabi and R. Tapia, *A Global Convergence Theory for Arbitrary Norm Trust Region Methods for Nonlinear Equations*, Tech. report TR93-41(replaces 87-25), Department of Computational and Applied Mathematics, Rice University, Houston, TX, submitted.

[12] R. Fletcher, *A class of methods for nonlinear programming with termination and convergence properties*, in Integer and Nonlinear Programming, J. Abadie, ed., North-Holland, Amsterdam, 1970.

[13] R. Fletcher, *An exact penalty function for nonlinear programming with inequalities*, Math. Programming, 5 (1973), pp. 129–150.

[14] R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, New York, 1987.

[15] P. Gill, W. Murray, M. Saunders, and M. Wright, *Some Theoretical Properties of an Augmented Lagrangian Merit Function*, Tech. Report SOL 86-6, Stanford University, Stanford, CA, 1986.

[16] P. Gill, W. Murray, and M. Wright, *Practical Optimization*, Academic Press, New York, 1981.

[17] M. C. Maciel, *A Global Convergence Theory for a General Class of Trust Region Algorithm for Equality Constrained Optimization*, Ph.D. thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1992.

[18] J. J. Moré, *Recent developments in algorithms and software for trust region methods*, in Mathematical Programming. The State of the Art, A. Bachem, M. Grotschel, and B. Korte, eds., Springer-Verlag, New York, 1983, pp. 258–287.

[19] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[20] S. G. Nash, *Preconditioning of truncated-Newton methods*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 599–616.

[21] E. O. Omojukun, *Trust Region Strategies for Optimization with Nonlinear Equality and Inequality Constraints*, Ph.D. thesis, Department of Computer Science, University of Colorado, Boulder, CO, 1989.

[22] M. J. D. Powell, *Convergence properties of a class of minimization algorithms*, in Nonlinear Programming 2, O. Mangasarian, R. Meyer, and S. Robinson, eds., Academic Press, New York, 1975, pp. 1–27.

[23] M. J. D. Powell and Y. Yuan, *A trust region algorithm for equality constrained optimization*, Math. Programming, 49 (1991), pp. 189–211.

[24] K. Schittkowski, *On the convergence of a sequential quadratic programming method with an augmented Lagrangian line search function*, Math. Operationsforch. Statist. Ser. Optim., 14 (1983), pp. 197–216.

[25] G. A. Shultz, R. B. Schnabel, and R. H. Byrd, *A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties*, SIAM J. Numer. Anal., 22 (1985), pp. 47–67.

[26] D. Sorensen, *Newton's method with a model trust region modification*, SIAM J. Numer. Anal., 19 (1982), pp. 409–426.

[27] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[28] R. A. Tapia, *Quasi-Newton methods for equality constrained optimization: Equivalence of existing methods and a new implementation*, in Nonlinear Programming 3, O. Mangasarian, R. Meyer, and S. Robinson, eds., Academic Press, New York, 1978, pp. 125–164.

[29] R. A. Tapia, *On secant update for use in general constrained optimization*, Math. Comp., 51 (1988), pp. 181–202.

[30] P. L. Toint, *Towards an efficient sparsity exploiting Newton method for minimization*, Tech. Report 80/4, Departément de Mathématique, Facultés Universitaires de Namur, Belgium, 1980.

[31] A. Vardi, *Trust Region Strategies for Unconstrained and Constrained Minimization*, Ph.D. thesis, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1980.

[32] A. Vardi, *A trust region algorithm for equality constrained minimization: Convergence properties and implementation*, SIAM J. Numer. Anal., 22 (1985), pp. 575–591.

[33] K. A. Williamson, *A robust trust region algorithm for nonlinear programming*, Ph.D. thesis, Department of Mathematical Sciences, Rice University, Houston, TX, 1990.

[34] J. Zhang, N.-H. Kim, and L. Lasdon, *An improved successive linear programming algorithm*, Management Sci., 31 (1985), pp. 1312–1331.

SIAM J. OPTIM.
Vol. 7, No. 1, pp. 208–224, February 1997

011

ALGORITHMS FOR CONSTRAINED AND WEIGHTED NONLINEAR LEAST SQUARES[*]

MÅRTEN GULLIKSSON[†], INGE SÖDERKVIST[‡], AND PER-ÅKE WEDIN[†]

Abstract. A hybrid algorithm consisting of a Gauss–Newton method and a second-order method for solving constrained and weighted nonlinear least squares problems is developed, analyzed, and tested. One of the advantages of the algorithm is that arbitrarily large weights can be handled and that the weights in the merit function do not get unnecessarily large when the iterates diverge from a saddle point. The local convergence properties for the Gauss–Newton method are thoroughly analyzed and simple ways of estimating and calculating the local convergence rate for the Gauss–Newton method are given. Under the assumption that the constrained and weighted linear least squares subproblems attained in the Gauss–Newton method are not too ill conditioned, global convergence towards a first-order KKT point is proved.

Key words. nonlinear least squares, optimization, parameter estimation, weights

AMS subject classifications. 65K, 49D

PII. S1052623493248809

1. Introduction. Assume that $f : \mathbf{R}^n \rightarrow \mathbf{R}^m$ is a twice continuously differentiable function and that $W = \operatorname{diag}(\omega_1, \ldots, \omega_m)$ is a diagonal matrix with weights $\omega_i \geq 0$. We will discuss the Gauss–Newton method and a second-order method for solving the problem

$$\min_{x\in\mathbf{R}^n} \frac{1}{2}\|W^{1/2}f(x)\|^2, \tag{1.1}$$

where $\|\cdot\|$ denotes the 2-norm. For simplicity and without loss of generality, we assume that the weights are normalized and sorted such that

$$\omega_1 \geq \cdots \geq \omega_m \geq 1. \tag{1.2}$$

The normalization is easily done by first sorting out the zero weights, reducing the problem, and then dividing the remaining nonzero weights with the smallest positive weight.

To our knowledge, all existing algorithms for solving (1.1) are based on the unweighted problem

$$\min_{x\in\mathbf{R}^n} \frac{1}{2}\|g(x)\|^2, \tag{1.3}$$

where $g(x) = W^{1/2}f(x)$; see also [1]. Assume that the ordinary Gauss–Newton method is used to solve (1.3). The search direction, p, is then obtained by solving

$$\min_{p} \frac{1}{2}\|Kp + g\|^2, \tag{1.4}$$

[*] Received by the editors May 7, 1993; accepted for publication (in revised form) October 11, 1995.

http://www.siam.org/journals/siopt/7-1/24880.html

[†] Institute of Information Processing, University of Umeå, S–901 87 Umeå, Sweden (marten@cs.umu.se, pwedin@cs.umu.se).

[‡] Department of Mathematics, Lulea University of Technology, S–97187 Lulea, Sweden (inge@sm.luth.se).

where $K = \nabla g$. Note that (1.4) is solved as an unweighted problem and thus the condition of this problem is determined by $\|K\|\,\|K^{\dagger}\|$, where $K^{\dagger}$ is the pseudoinverse of K.

If, on the other hand, we linearize (1.1) without explicitly multiplying $J = \nabla f$ with the weights, we solve the *weighted* linear least squares problem

$$\min_{p} \frac{1}{2}\|W^{1/2}(Jp + f)\|^2 \tag{1.5}$$

to obtain the search direction p. The condition for the problem (1.5) is mainly determined by $\|B\|\,\|J\|$, where $BJ = I_n$. For a more detailed discussion on condition numbers for (1.5), see [12]. The problem (1.4) may be very ill conditioned (regarded as an *unweighted* linear least squares problem) despite the fact that (1.5) is well conditioned (regarded as a *weighted* linear least squares problem). Obviously it is very important to look at (1.1) as the class of *weighted* nonlinear least squares problem.

Another important advantage of using (1.1) instead of (1.3) is that the former defines a more general problem class than the latter. This is evident if we allow the weights to be infinitely large. To be more precise, we define the vector $\lambda \in \mathbf{R}^m$ by the equations

$$M\lambda = f, \; M = \operatorname{diag}(\mu_1, \ldots, \mu_m), \tag{1.6}$$

where $\mu_i = 1/\omega_i$ and infinite weights correspond to zero elements in M. Note that if $\mu_i = 0$ then λ_i is the Lagrange multiplier corresponding to the ith constraint and consequently λ_i is not defined by (1.6). We will return to the proper way of calculating these Lagrange multipliers. Problem (1.1) is rewritten, using (1.6), as

$$\min_{\lambda, x} \; \frac{1}{2}\lambda^T M\lambda \text{ s.t. } M\lambda = f(x). \tag{1.7}$$

Hence, by allowing infinite weights, our original problem formulation (1.1) defines the class of *weighted nonlinear least squares problems with nonlinear equality constraints.* To be even more specific, we assume that we have p infinite weights such that

$$M = \operatorname{diag}(0_p, M_2), \quad M_2 = \operatorname{diag}(\mu_{p+1}, \ldots, \mu_m),$$

where $\mu_{p+1} > 0$. Problem (1.7) can now be stated as

$$\min_{\lambda, x} \; \frac{1}{2}\lambda_2^T M_2 \lambda_2 \text{ s.t. } f_1(x) = 0, \quad M_2\lambda_2 = f_2(x), \tag{1.8}$$

where $\lambda = \left[\lambda_1^T, \lambda_2^T\right]^T$ and $f = \left[f_1^T, f_2^T\right]^T$. An equivalent formulation of problem (1.8) without using λ is

$$\min_{x} \frac{1}{2}\|W_2^{1/2} f_2(x)\|^2 \text{ s.t. } f_1(x) = 0, \tag{1.9}$$

where $W_2 = M_2^{-1}$. Of course, we could have started by defining our problem as the one in (1.9) instead of (1.5) (without the need of (1.7) and (1.8)), but then the notations would get unnecessarily complicated.

In the next section we describe the Gauss–Newton method for solving (1.1). The local convergence properties of the Gauss–Newton method are analyzed in section 3, and in section 4 we show that, under certain assumptions on nondegeneracy, global convergence is achieved. If the Gauss–Newton method is too slow or does not converge and second derivatives are available at a reasonable cost, then the Newton method may be used to solve (1.1). However, when there are large and possibly infinite weights, a pure Newton method based on forming the Hessian of $g(x)$ may not work or, with infinite weights, is not even defined. The natural approach is then to use the perturbation method [9] that we will call the generalized Newton–Raphson method (the gNR method). In section 5 we construct and analyze an algorithm for solving (1.1) based on the gNR method. Computational experiments are presented in section 6 and, finally, we discuss our results and give hints of possible future work.

2. The Gauss–Newton method using the system equations. In the Gauss–Newton method, the nonlinear least squares problem (1.1) is linearized around the current iteration point, x_k, and the search direction, p_k, is computed as the solution to

$$\min_{p_k} \frac{1}{2}\|W^{1/2}(f_k + J_k p_k)\|^2, \tag{2.1}$$

where $f_k = f(x_k), J_k = \nabla f(x_k)$. The next iterate is $x_{k+1} = x_k + \alpha_k p_k$, where α_k is the steplength. In the presence of large weights, possibly infinite, it is adequate to reformulate (2.1) as

$$\begin{bmatrix} M & J \\ J^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ p \end{bmatrix} = \begin{bmatrix} -f \\ 0 \end{bmatrix}, \tag{2.2}$$

where for simplicity we have dropped the iteration index k. There are several names for the linear system of equations in (2.2) such as the equilibrium equations, the system equations, or the augmented system equations. We call (2.2) the *system equations* and the matrix in (2.2) is called the *system matrix.* A less obvious reason for using (2.2) is that the elements in λ corresponding to infinite weights are approximations to the Lagrange multipliers and that λ can be used in a second-order method as described in section 5.

The following lemma gives the relevant conditions for the system matrix to be nonsingular.

LEMMA 2.1. *The system matrix in* (2.2) *is nonsingular if and only if the rows in J that correspond to infinite weights are linearly independent and J has full column rank.*

There exist several stable algorithms that solve (2.2); see, e.g., [5] for further references. We have chosen to use the modified QR decomposition, see [5], and the reasons are the following. The modified QR decomposition is simple and easy to compute and it is identical to the ordinary QR decomposition when the weights are equal. The modified QR decomposition is also easily reused in the second order gNR method; see section 5.

The modified QR decomposition of $J \in \mathbf{R}^{m\times n}$ is defined as

$$J\Pi = Q \begin{bmatrix} R \\ 0 \end{bmatrix}, \quad QMQ^T = M, \tag{2.3}$$

where $Q \in \mathbf{R}^{m\times m}$, $R \in \mathbf{R}^{n\times n}$ is an upper triangular matrix and Π is a permutation matrix. The decomposition in (2.3), with Q and R nonsingular, exists if and only if the system matrix in (2.2) is nonsingular (see Lemma 2.1).

The system equations are solved with the modified QR decomposition in the following way. Using the decomposition (2.3) in (2.2) we get

$$\begin{bmatrix} M & \begin{bmatrix} R \\ 0 \end{bmatrix} \\ \begin{bmatrix} R^T, 0 \end{bmatrix} & 0 \end{bmatrix} \begin{bmatrix} Q^T\lambda \\ \Pi^T p \end{bmatrix} = \begin{bmatrix} -Q^{-1}f \\ 0 \end{bmatrix}. \tag{2.4}$$

If we make the partition $M = \operatorname{diag}(M_n, M_{m-n})$, the solution to (2.4) is

$$p = -\Pi \left[R^{-1}, 0\right] Q^{-1}f, \ \ \lambda = -Q^{-T} \begin{bmatrix} 0 & \\ & M_{m-n}^{-1} \end{bmatrix} Q^{-1}f. \tag{2.5}$$

3. The local rate of convergence for the Gauss–Newton method. In this section we will describe the local convergence properties of the Gauss–Newton method described in the previous section. Our analysis depends upon the perturbation analysis of the constrained and weighted linear least squares problem done in [12, 11]. After having defined the inverse of the system matrix, using the same notation as in [12], we state and prove two important theorems on the local convergence rate for $x_k - \widehat{x}$ and $J_k p_k$ (the projected residual). In fact, the local convergence properties of these two quantities are, as we shall see, very similar. Finally, we show that $J_k p_k$ and the local convergence rate for $x_k - \widehat{x}$ and $J_k p_k$ are independent of the parametrization in $\mathbf{R}^n$.

Assuming that $\widehat{x}$ is a solution of (1.1), we define $\widehat{f} = f(\widehat{x})$ and the corresponding notation for other quantities evaluated at $\widehat{x}$.

A necessary condition for our algorithm to converge without regularization is that the system matrix in (2.2) has full rank, and it is convenient to make the following definition.

DEFINITION 3.1. *If the system matrix in* (2.2) *is nonsingular at x we say that x is a* nondegenerate *point.*

At a nondegenerate point the inverse of the system matrix in (2.2) is given by

$$\begin{bmatrix} M & J \\ J^T & 0 \end{bmatrix}^{-1} = \begin{bmatrix} Y & B^T \\ B & -BMB^T \end{bmatrix}, \tag{3.1}$$

where $BJ = I_n$; i.e., B is a generalized inverse of J; see [12]. From (2.5) we immediately get

$$B = \Pi \left[R^{-1}, 0\right] Q^{-1}. \tag{3.2}$$

The following theorem describes the local behavior of $x_k - \widehat{x}$ where $x_{k+1} = x_k + p_k$.

THEOREM 3.1. *Assume that $\{p_k\}$ are generated by solving* (2.2) *and that all points $x_k = x_{k-1} + p_{k-1}$ are nondegenerate. If $\widehat{x}$ is the solution of* (1.1) *and $\widehat{\lambda}$ is the vector λ from* (2.2) *at $\widehat{x}$, then*

$$q_{k+1} = B_k M B_k^T \sum_{i=1}^{m} \widehat{\lambda}_i \bar{T}_i q_k + \frac{1}{2} B_k \begin{bmatrix} q_k^T \underline{T}_1 q_k \\ \vdots \\ q_k^T \underline{T}_m q_k \end{bmatrix}, \tag{3.3}$$

where $q_k = x_k - \widehat{x}$, $\bar{T}_i = \int_0^1 f_i''(\widehat{x} + \tau q_k) d\tau$, and $\underline{T}_i = 2\int_0^1 (1-\tau) f_i''(x_k - \tau q_k) d\tau$.

Proof. From $x_{k+1} = x_k - B_k f_k$ we get

$$q_{k+1} = \left[q_k - B_k(f_k - \widehat{f})\right] - B_k \widehat{f}. \tag{3.4}$$

Using the Taylor expansion

$$f(\widehat{x}) = f(x_k - q_k) = f_k - J_k q_k + \int_0^1 (1-\tau) \begin{bmatrix} q_k^T f_1''(x_k - \tau q_k) q_k \\ \vdots \\ q_k^T f_m''(x_k - \tau q_k) q_k \end{bmatrix} d\tau,$$

the first term in (3.4) can be expressed as

$$q_k - B_k(f_k - \widehat{f}) = \frac{1}{2} B_k \begin{bmatrix} q_k^T \underline{T}_1 q_k \\ \vdots \\ q_k^T \underline{T}_m q_k \end{bmatrix}. \tag{3.5}$$

To express the second term, $-B_k\widehat{f}$, in (3.4), we use the perturbation identity (2.2) (p. 16 in [12]), which says that

$$\widehat{p} - p_k = -\widehat{B}\widehat{f} + B_k f_k = B_k(J_k - \widehat{J})\widehat{p} - B_k M B_k^T (J_k - \widehat{J})^T \widehat{\lambda} + B_k(f_k - \widehat{f}).$$

Since $\widehat{p} = -\widehat{B}\widehat{f} = 0$, we get

$$-B_k\widehat{f} = B_k M B_k^T (J_k - \widehat{J})^T \widehat{\lambda}. \tag{3.6}$$

Using the identity

$$(J_k - \widehat{J})^T = \int_0^1 [f_1''(\widehat{x} + \tau q_k) q_k, \dots, f_m''(\widehat{x} + \tau q_k) q_k]\, d\tau,$$

equation (3.6) becomes

$$-B_k\widehat{f} = B_k M B_k^T \sum_{i=1}^{m} \widehat{\lambda}_i \bar{T}_i q_k. \tag{3.7}$$

The equations (3.5) and (3.7) inserted into (3.4) give the theorem. □

The Gauss–Newton method can be written as

$$x_{k+1} = \vartheta(x_k), \quad \vartheta(x) = x - B(x) f(x), \tag{3.8}$$

and with $\widehat{x} = \vartheta(\widehat{x})$ we get

$$q_{k+1} = x_{k+1} - \widehat{x} = \vartheta(x_k) - \vartheta(\widehat{x}) = \nabla\vartheta(\widehat{x}) q_k + \mathcal{O}(\|q_k\|^2). \tag{3.9}$$

From Theorem 3.1 we conclude that

$$\nabla\vartheta(\widehat{x}) = \widehat{B} M \widehat{B}^T \sum_{i=1}^{m} \widehat{\lambda}_i \widehat{f}_i'', \tag{3.10}$$

and from [8] we get the following theorem.

THEOREM 3.2. *Define*

$$H_x = \nabla\vartheta(\widehat{x}) = \widehat{B} M \widehat{B}^T \sum_{i=1}^{m} \widehat{\lambda}_i \widehat{f}_i'' \tag{3.11}$$

and κ_i as the eigenvalues of H_x. Then

$$\limsup_{k\to\infty} \frac{\|x_{k+1} - \widehat{x}\|}{\|x_k - \widehat{x}\|} \le \max_i |\kappa_i|.$$

It is easy to get an estimation of the local convergence rate if we use the matrix B defined by (3.2), because then $\widehat{B}M\widehat{B}^T = \Pi\widehat{R}^{-1}M_n\widehat{R}^{-T}\Pi^T$.

A useful quantity for estimating how close x_k is to the solution, $\widehat{x}$, is the projected residual $J_kp_k = -J_kB_kf_k$, where $J_kB_kf_k$ is the oblique projection of f_k onto $\mathcal{R}(J_k)$. The following theorem shows that J_kp_k locally has the same convergence behavior as $x_k - \widehat{x}$.

Theorem 3.3. *Assume that $\{p_k\}$ are generated by solving* (2.2) *and that all points $x_k = x_{k-1} + p_{k-1}$ are nondegenerate. If $\lambda(x_k)$ is the vector λ from* (2.2) *at x_k then*

$$s_{k+1} = M\sum_{i=1}^{m}\lambda_i(x_k)\bar{S}_is_k + \frac{1}{2}J_{k+1}B_{k+1}\begin{bmatrix} s_k^T\underline{S}_1s_k \\ \vdots \\ s_k^T\underline{S}_ms_k \end{bmatrix}, \tag{3.12}$$

where $s_k = -J_kp_k$, $\bar{S}_i = B_{k+1}^T\int_0^1 f_i''(x_k + \tau p_k)d\tau B_k$, and $\underline{S}_i = 2B_k^T\int_0^1(1-\tau)f_i''(x_k + \tau p_k)d\tau B_k$.

Proof. Denote the projection J_kB_k by P_k. Then we have

$$s_k = -J_kp_k = J_kB_kf_k = P_kf_k.$$

Using the Taylor expansion

$$f_{k+1} = f_k + J_kp_k + \int_0^1(1-\tau)v(\tau)d\tau, \tag{3.13}$$

where $v(\tau) = [p_k^Tf_1''(x_k + \tau p_k)p_k, \ldots, p_k^Tf_m''(x_k + \tau p_k)p_k]^T$, by multiplying with P_{k+1} we obtain

$$s_{k+1} = P_{k+1}(I - P_k)f_k + P_{k+1}\int_0^1(1-\tau)v(\tau)d\tau. \tag{3.14}$$

Since $B_kJ_k = I$, the equality $B_ks_k = -p_k$ holds, and we can identify the last term in equation (3.14) as

$$P_{k+1}\int_0^1(1-\tau)v(\tau)d\tau = \frac{1}{2}J_{k+1}B_{k+1}\begin{bmatrix} s_k^T\underline{S}_1s_k \\ \vdots \\ s_k^T\underline{S}_ms_k \end{bmatrix}. \tag{3.15}$$

From (3.1) we get

$$YJ = 0, \quad JB = I - MY, \quad YMY = Y, \tag{3.16}$$

and hence

$$\begin{aligned} P_{k+1}(I - P_k) = (P_{k+1} - P_k)(I - P_k) = \\ (J_{k+1}B_{k+1} - J_kB_k)(I - J_kB_k) = -M(Y_{k+1} - Y_k)(I - J_kB_k). \end{aligned} \tag{3.17}$$

From the perturbation identity (2.1) (p. 16 in [12]) we get

$$Y_{k+1} - Y_k = -Y_{k+1}\delta J_kB_k - B_{k+1}^T(\delta J_k)^TY_k, \tag{3.18}$$

where $\delta J_k = J_{k+1} - J_k$. Using (3.18) and the fact that $Y_{k+1}\delta J_kB_k(I - J_kB_k)$ vanishes, the equation (3.17) becomes

$$P_{k+1}(I - P_k) = MB_{k+1}^T(\delta J_k)^TY_k(I - J_kB_k) = MB_{k+1}^T(\delta J_k)^TY_k,$$

where the last equality follows from (3.16). The identities $\lambda_k = -Y_k f_k$ and $B_k s_k = -p_k$ together with a Taylor expansion of $(\delta J_k)^T$ give

$$P_{k+1}(I - P_k)f_k = -MB_{k+1}^T(\delta J_k)^T\lambda_k = M\sum_{i=1}^{m}\lambda_i(x_k)\bar{S}_i s_k. \tag{3.19}$$

The theorem follows by inserting (3.15) and (3.19) into (3.14). □

The matrix corresponding to H_x for the projected residual, s_k, is

$$H_s = M\widehat{B}^T\sum_{i=1}^{m}\widehat{\lambda}_i\widehat{f}_i''\widehat{B},$$

and it is easy to show that H_x and H_s have the same nonzero eigenvalues. Hence, we have the following corollary from Theorem 3.3.

COROLLARY 3.1. *Define B_k from the inverse of the system matrix in* (3.1). *If $s_k = -J_k B_k f_k$ then*

$$\limsup_{k\to\infty}\|s_{k+1}\|/\|s_k\| \le \max_i|\kappa_i|,$$

where κ_i are the eigenvalues of the matrix H_x defined in (3.11).

The relation (3.12) can also be used to determine when $\|s_{k+1}\|/\|s_k\|$ reflects the linear convergence rate and if a second-order method should be used. If the convergence of the Gauss–Newton method is slow, we use a higher order method if

$$\frac{1}{2}\|J_k p_k\| \le \|M\|\ \|\lambda(x_k)\|. \tag{3.20}$$

See also Algorithm 6.1.

Several of the above quantities are invariant under a change of parametrization $x = x(\theta)$, and as an example we have the following theorem.

THEOREM 3.4. *The matrix*

$$H_s = M\widehat{B}^T\sum_{i=1}^{m}\widehat{\lambda}_i\widehat{f}_i''\widehat{B}$$

is independent of the parametrization in $\mathbf{R}^n$.

Proof. Assume that $x = x(\theta)$ and $\widehat{x} = x(\widehat{\theta})$. Define $C = \partial x/\partial\theta$ and $y(\theta) = f(x(\theta))$. We want to show that

$$H_y = M\widehat{B}_y^T\sum_{i=1}^{m}\widehat{\lambda}_i\widehat{y}_i''\widehat{B}_y = H_s,$$

where $\widehat{B}_y$ is the generalized inverse of $\nabla y(\widehat{\theta})$. Now, consider the Taylor expansion

$$f(x(\theta + \Delta\theta)) = f(x) + JC\Delta\theta + \frac{1}{2}(Jh_x + h_f) + \mathcal{O}(\|\Delta\theta\|^3), \tag{3.21}$$

where

$$h_x = \begin{bmatrix} \Delta\theta^T x_1''(\theta)\Delta\theta \\ \vdots \\ \Delta\theta^T x_n''(\theta)\Delta\theta \end{bmatrix}, h_f = \begin{bmatrix} (C\Delta\theta)^T f_1''(x)C\Delta\theta \\ \vdots \\ (C\Delta\theta)^T f_m''(x)C\Delta\theta \end{bmatrix}.$$

By comparing the Taylor expansion (3.21) with the Taylor expansion

$$y(\theta + \Delta\theta) = y(\theta) + \nabla y \,\Delta\theta + \frac{1}{2} h_y + \mathcal{O}(\|\Delta\theta\|^3), \tag{3.22}$$

where $h_y = [\Delta\theta^T y_1''(\theta)\Delta\theta, \ldots, \Delta\theta^T y_m''(\theta)\Delta\theta]^T$, and using $J^T\lambda = 0$ we conclude that

$$B_y = C^{-1}B, \quad \sum_{i=1}^{m} \lambda_i y_i''(\theta) = \sum_{i=1}^{m} \lambda_i C^T f_i''(x) C. \tag{3.23}$$

From (3.23) we finally get

$$B_y^T \sum_{i=1}^{m} \lambda_i y_i'' B_y = B^T C^{-T} \sum_{i=1}^{m} \lambda_i C^T f_i'' C C^{-1} B = B^T \sum_{i=1}^{m} \lambda_i f_i'' B,$$

which proves the theorem. □

A consequence of Theorem 3.4 is that the local convergence for $\theta_k - \widehat{\theta}$ is the same as for $x_k - \widehat{x}$.

The main argument for choosing Jp as a measure of the closeness to the solution is the following theorem which is a direct consequence of (3.23).

Theorem 3.5. *The projection of f on $\mathcal{R}(J)$, $JBf = -Jp$, is independent of the parametrization in $\mathbf{R}^n$.*

4. Global convergence. In this section we assume that x_k, where k is the iteration index, is nondegenerate and that p_k is the solution of (2.2) at x_k. If nothing else is stated we assume that all limits denoted by → are when $k \to \infty$ and that all sums with no explicitly stated upper or lower limit are from one to infinity.

4.1. The merit function. As a merit function we have chosen

$$\Phi(x, D) = \frac{1}{2} f(x)^T D f(x), \tag{4.1}$$

where $D = \operatorname{diag}(d_1, \ldots, d_m)$, $1 \le d_i \le \omega_i$.

The goal is to find a matrix D_k of merit weights and a steplength α_k at each iteration such that global convergence towards a first-order Kuhn–Tucker point can be proved. To compute D_k we will use the approximation

$$\Psi(x_k, p, D) = \frac{1}{2}(f_k + J_k p)^T D (f_k + J_k p)$$

of $\Phi(x_k + p, D)$. For a fixed matrix D, we define $\phi(\alpha) = \Phi(x_k + \alpha p_k, D)$. Obviously a sufficient condition on p_k to be a descent direction to $\Phi(x, D)$ at x_k is that $\phi'(0) = p_k^T J_k^T D f_k < 0$.

We realize that we can determine a good matrix, $\Upsilon(x_k)$, of merit weights by solving

$$\min_{\Upsilon = diag(u_1,\ldots,u_m)} \|\Upsilon\| \;\text{ s.t. } \begin{cases} 1 - \delta \le \arg\{\, \min_\alpha \Psi(x_k, \alpha p_k, \Upsilon) \,\} \\ \xi_i \le u_i \le \omega_i, i = 1, \ldots, m, \end{cases} \tag{4.2}$$

where δ is a small positive constant and ξ_i is a lower limit for the weights determined by some previously computed weights; see below. There is always a solution to (4.2) because

$$\lim_{\Upsilon \to W} \arg\{\, \min_\alpha \Psi(x_k, \alpha p_k, \Upsilon) \,\} = 1.$$

Note that keeping the weights not too large is important in practice, but for the global convergence it is only the constraints in (4.2) that must be satisfied. We will now describe the algorithm for computing the merit weights D_k, using $\Upsilon(x_k)$, such that D_k does not become unnecessarily large. We first describe a method for solving (4.2) and then an algorithm for computing the actual merit weights D_k.

When solving (4.2) we have chosen to use the max-norm since this gives a simple algorithm. The problem (4.2) can be rewritten as

$$\min \ \|u\|_\infty \ \text{ s.t. } y^T u \geq 0, \quad \xi \leq u \leq \omega, \tag{4.3}$$

where u is the diagonal in $\Upsilon(x_k)$, ω is the diagonal in W, and $y_i = -f_i s_i - (1-\delta)s_i^2$ with $s = Jp$. Note that when Jp is given the problem, (4.3) consists of only vectors and no matrices. The first step in our algorithm is to reduce (4.3) such that $u_i = \xi_i$ if $y_i \leq 0$. We then get the new problem

$$\min \ \|\bar{u}\|_\infty \ \text{ s.t. } \bar{y}^T \bar{u} \geq \rho, \quad \bar{\xi} \leq \bar{u} \leq \bar{\omega}, \tag{4.4}$$

where $\bar{u}, \bar{\xi}, \bar{\omega}$, and $\bar{y}$ are the corresponding parts of u, ξ, ω, and y left after the reduction and $\rho = -\sum_{i|y_i \leq 0} y_i \xi_i$. If $\bar{y}^T \bar{\xi} \geq \rho$ we are ready with the solution $\bar{u} = \bar{\xi}$. Otherwise we choose $\bar{u}_i = \rho / e^T \bar{y}$, where e is a vector of ones, and thus attain equality in the constraints. If $\bar{u}_i > \omega_i$ or $\bar{u}_i \leq \xi_i$ we set $\bar{u}_i = \omega_i$ and $\bar{u}_i = \xi_i$, respectively. Again we can reduce the problem to a copy of (4.4) but where the vectors are shorter and ρ is smaller. The procedure is then repeated until the whole of u is found. It is easily realized that the infinite weights in ω do not change the algorithm and the algorithm will terminate with a solution of (4.4).

We determine the actual merit weights D_k from the solution $\Upsilon(x_k)$ of (4.2). The weights may get large close to a saddle point, and when the iterates diverge from this saddle point (that is always the case with the Gauss–Newton method) we would like the weights to decrease. This is accomplished by saving say, t older versions, $V_1, \ldots, V_t$, of the merit weight matrices. Initially, at iteration $k = 1$, we have $V_i := I_m, i = 1, \ldots, t$ and at the kth iteration we update $V_i = \text{diag}(\nu_1^{(i)}, \ldots, \nu_m^{(i)})$, as in Algorithm 4.1.

Algorithm 4.1.

```
*Solve* (4.2) *for the vector* $u(x_k)$.
**for** $i = 1{:}m$
    $d_i^{(k)} := \max\{u_i(x_k), \nu_i^{(t)}\}$
    **If** $d_i^{(k)} > d_i^{(k-1)}$
        *Let* $\nu_i^{(1)} \geq \cdots \geq \nu_i^{(j-1)} \geq d_i^{(k)} \geq \nu_i^{(j)} \geq \cdots \geq \nu_i^{(t-1)}$ *be the new*
        *sequence* $\nu_i^{(1)}, \ldots, \nu_i^{(t)}$.
    **end**
**end**
```

In Algorithm 4.2 our Gauss–Newton algorithm is described with line search and quadratic merit function.

Algorithm 4.2.

```
$k := 1$; *Initiate the start vector* $x_k$.
**while not** *convergence*
    *Compute* $J_k$ *and* $f_k$.
    *Compute* $p_k$ *from* (2.2) *using the modified QR decomposition of* $J_k$.
    *Determine* $D_k$ *from Algorithm* 4.1.
    *Determine the steplength* $\alpha_k$ *such that*
    $\phi(\alpha_k) \leq \phi(0) + \mu\alpha_k\phi'(0)$, $0 < \mu < 1$.
    $x_{k+1} := x_k + \alpha_k p_k$;  $k := k + 1$
```

end

4.2. Proving global convergence. We will need the following two technical lemmas to prove that our algorithm is globally convergent. In the lemmas we use d_k as an arbitrary diagonal element in D_k.

LEMMA 4.1. *Assume that* $d_k \geq 0, k = 1, \dots$ *and that* $\{d_k\}$ *is bounded. Let* $\{d_{k_j}\}$ *be the subsequence of* $\{d_k\}$ *such that* $d_{k_{j+1}} > d_{k_j}$. *Then the positive series* $\sum (d_{k_{j+1}} - d_{k_j})$ *converges if and only if* $\sum |d_{k+1} - d_k|$ *converges.*

Proof. Take

$$b_N^+ = \sum_{\substack{a_k > 0 \\ 1 \leq k \leq N-1}} a_k, \qquad b_N^- = -\sum_{\substack{a_k \leq 0 \\ 1 \leq k \leq N-1}} a_k,$$

where $a_k = d_{k+1} - d_k$. Obviously $b_N^+ = \sum_{k_j \leq N-1} (d_{k_{j+1}} - d_{k_j})$, $b_N^+ - b_N^- = d_N - d_1$ and $b_N^+ + b_N^- = \sum_{k=1}^{N-1} |d_{k+1} - d_k|$. Hence, if $\sum_{k=1}^{N-1} |d_{k+1} - d_k|$ converges, then $b_N^+ = \sum_{k_j \leq N-1} |d_{k_j+1} - d_{k_j}|$ converges too. Now assume that b_N^+ converges to b^+, $d_N - d_1 = b_N^+ - b_N^- \geq -d_1$, and $b^+ \geq b_N^+$ imply $b^+ + d_1 \geq b_N^-$. Hence, $\{b_k^-\}$ is a bounded sequence that increases to a limit b^- and $\sum |d_k - d_{k-1}|$ converges to $b^+ + b^-$. □

LEMMA 4.2. *Assume that an arbitrary component,* d_k, *in the diagonal of* D_k *stays bounded as* $k \to \infty$ *and let* v_k *be the corresponding diagonal element in* V_t. *Then* $\lim_{k\to\infty} v_k = \lim_{k\to\infty} d_k$ *and the series* $\sum |d_{k+1} - d_k|$ *converges.*

Proof. Let us first exclude the trivial case where v_k becomes equal to the upper bound ω for a finite k.

The sequence $\{v_k\}$ is an increasing infinite sequence. Hence, $\lim v_k$ exists and is denoted v. Take $\underline{d} = \liminf d_k$ and $\bar{d} = \limsup d_k$. Let ϵ be an arbitrary small but fixed positive number. Then $d_k > \bar{d} - \epsilon$ for more than t k-values. Hence, $v > \bar{d} - \epsilon$ and since $\epsilon > 0$ was arbitrary, this implies that $v \geq \bar{d}$. From $d_k \geq v_k$ it follows that $\underline{d} \geq v$ and thus we have $v \geq \bar{d} \geq \underline{d} \geq v$ and consequently $d_k \to v$.

Let $\{d_{i_k}\}$ be the subsequence of $\{d_k\}$ with $d_{i_{k+1}} > d_{i_k}$. From Lemma 4.1 we know that the series $\sum |d_{k+1} - d_k|$ converges if and only if $\sum (d_{i_{k+1}} - d_{i_k})$ converges. Let us now prove that the latter series converges. From $v_{i_k} \leq d_{i_k}$ it follows that

$$d_{i_{k+1}} - d_{i_k} \leq (d_{i_{k+1}} - v_{i_{k+1}}) + (v_{i_{k+1}} - v_{i_k})$$

and hence

$$\sum (d_{i_{k+1}} - d_{i_k}) \leq \sum [\, (d_{i_{k+1}} - v_{i_{k+1}}) + (v_{i_{k+1}} - v_{i_k}) \,].$$

Since $\sum (v_{i_{k+1}} - v_{i_k})$ is a subseries of $\sum (v_{k+1} - v_k)$ and v_k increases to v, the series $\sum (v_{i_{k+1}} - v_{i_k})$ converges. Since $\sum (d_{i_{k+1}} - d_{i_k})$ is a positive series, it is sufficient to prove that it is bounded. Hence, it only remains to prove that the series $\sum (d_{i_{k+1}} - v_{i_{k+1}})$ converges. Since $d_{i_2} > d_{i_1}$, the saved older weights are updated in step i_1. When we reach $d_{i_{1+t}}$ there have been t updates, and $v_{i_{1+t}}$ equals one of the earlier $d_{i_j}, j = 1, \dots, t$. In this way we can eliminate both this $v_{i_{1+t}}$ and the corresponding d_{i_j}. In the same way it is seen that $v_{i_{1+t+1}}$ equals one of the d_{i_j}. That pair can also be eliminated from the series. We go on and eliminate elements in this way to get

$$\sum (d_{i_k} - v_{i_k}) = (d_{j_1} + \dots + d_{j_q}) - (v_{k_1} + \dots + v_{k_q}),$$

where $q \leq t$. Thus the positive series $\sum (d_{i_{k+1}} - v_{i_{k+1}})$ is bounded and so converges. That completes the proof. □

Our main global convergence theorem covers both bounded and unbounded sequences of merit weights.

THEOREM 4.3. *Let* $\{x_k\}$ *and* $\{D_k\}$ *be generated by Algorithm* 4.2. *Assume that* $\{x_k\}$ *is bounded and that the system matrix in* (2.2) *is nonsingular in the closure of* $\{x_k\}$. *Then the sequence* $\{x_k\}$ *has either finite termination at a KKT point or an accumulation point that is a KKT point of* (1.1).

Proof. It is trivial that there is finite termination just at KKT points. Let us now assume that we have an infinite sequence. Algorithm 4.2 implies that it is sufficient to consider the following two cases:

(i) $\|D_k\| \to \infty$,

(ii) $\{\|D_k\|\}$ is bounded.

These cases will now be treated separately.

(i) There exists a subsequence $\{x_{i_k}\}$ of $\{x_k\}$ such that $\|D_{i_k}\| \to \infty$ monotonically. Since $\{x_{i_k}\}$ is bounded, it is possible to choose a subsequence $\{x_{j_k}\}$ of $\{x_{i_k}\}$ such that $x_{j_k} \to \widetilde{x}$ for some $\widetilde{x}$. From Algorithm 4.2 it follows that $\|D_k\| \to \infty$ only when $\|\Upsilon(x_{j_k})\| \to \infty$. Since $\Upsilon(x)$ is continuous for all points in the closure of $\{x_k\}$ *except* KKT points, $\widetilde{x}$ is both an accumulation point of $\{x_k\}$ and a KKT point.

(ii) From the inequality

$$\Phi(x_N, D_N) - \Phi(x_1, D_1) \le$$
$$\frac{1}{2}\sup\{\|f_k\|^2\} \sum_{i=1}^{N-1} \|D_{i+1} - D_i\| - \sum_{k=1}^{N-1} (\Phi(x_k, D_k) - \Phi(x_{k+1}, D_k)),$$

one can prove that a point $\widetilde{x}$ cannot be an accumulation point of $\{x_k\}$ if there exist constants $\epsilon > 0$ and $\delta > 0$ such that

$$\Phi(x_k, D_k) - \Phi(x_{k+1}, D_k) \ge \epsilon, \;\; \|x_k - \widetilde{x}\| \le \delta. \tag{4.5}$$

(The proof of (4.5) is a trivial extension of a similar proof in [7, pp. 21–22].)

From Lemma 4.2 we know that $\sum \|D_{i+1} - D_i\|$ converges and from the Goldstein–Armijo condition in Algorithm 4.2 for a given D_k, it follows that for every point $\widetilde{x}$ in the closure of $\{x_k\}$ that is not a KKT point, there exist constants $\epsilon > 0$ and $\delta > 0$ such that (4.5) is satisfied. Hence, only KKT points remain as possible accumulation points. That proves the theorem in case (ii). ▯

4.3. Line search. We have chosen to keep things simple and therefore we use a standard cubic interpolation from [3] to approximate the minimum of our merit function $\phi(\alpha)$. Another more efficient line search algorithm can be found in [6].

4.4. Regularization. We use a simple form of subspace minimization described for the unweighted and constrained case in [7]. We have not been able to prove a general global convergence result such as the one in Theorem 4.3, but as we shall see in the computational experiments, our regularization seems to work appropriately.

5. The generalized Newton–Raphson method. A constrained Newton method for solving (1.9) can be based on the quadratic subproblem

$$\min_p \; p^T J_2^T W_2 f_2 + \frac{1}{2} p^T (J_2^T W_2 J_2 + \bar{G}) p \;\; \text{s.t. } f_1 + J_1 p = 0, \tag{5.1}$$

where $\bar{G} = -\sum_{i=1}^{p} \lambda_i f_i'' + \sum_{i=p+1}^{m} \omega_i f_i f_i''$ and $\lambda_i, i = 1, \ldots, p$, are first-order approximations of the Lagrange multipliers. The solution, $\bar{p}$, to (5.1) is given by the linear system of equations

$$\begin{bmatrix} M & J \\ J^T & -\bar{G} \end{bmatrix} \begin{bmatrix} \bar{\nu} \\ \bar{p} \end{bmatrix} = \begin{bmatrix} -f \\ 0 \end{bmatrix}. \tag{5.2}$$

The main disadvantage with using (5.2) is that for very large weights in W_2, the quadratic subproblem (5.1) and the matrix in (5.2) may be very ill conditioned.

To avoid the ill conditioning due to large weights in W_2, we solve

$$\begin{bmatrix} M & J \\ J^T & -G \end{bmatrix} \begin{bmatrix} \nu \\ p \end{bmatrix} = \begin{bmatrix} -f \\ 0 \end{bmatrix}, \tag{5.3}$$

where $G = -\sum_{i=1}^{m} \lambda_i f_i''$ and λ is from (2.2). This method is the gNR method.

The gNR method has an interesting theoretical motivation. Assume that we have reached a point x_k. From the first-order approximation (1.5) it is known that x_k solves the perturbed problem

$$\min_{x\in\mathbf{R}^n} \frac{1}{2}\|W^{1/2}z(x)\|^2, \tag{5.4}$$

where $z(x) = f(x) - P_k f_k$ and $P_k = J_k B_k$ is a projection onto $\mathcal{R}(J_k)$. Hence, we know the solution x_k of (5.4) and want to compute the solution of the perturbed problem

$$\min_{x\in\mathbf{R}^n} \frac{1}{2}\|W^{1/2}(z(x) + P_k f_k)\|^2. \tag{5.5}$$

Then we can use the quadratic approximation of $z(x)$ at x_k to compute a solution of problem (5.5) whose error is $\mathcal{O}(\|P_k f_k\|^2)$. If we change back to the original notations in $f(x)$, this perturbed solution is found by solving problem (5.3) for $f = f_k$.

From (5.3) it is seen that there exists a matrix N_k such that $p_k = -N_k f_k = -N_k P_k f_k$. With $\widehat{x}$ as the solution to (1.1) and $z(\widehat{x}) - z(x_k) \equiv f(\widehat{x}) - f(x_k)$, we have

$$f(\widehat{x}) = f(x_k) - J_k N_k P_k f_k + \mathcal{O}(\|P_k f_k\|^2). \tag{5.6}$$

Take $x_{k+1} = x_k - N_k f_k$. Then from the quadratic approximation in (5.6) we get $\|x_{k+1} - \widehat{x}\| = \mathcal{O}(\|P_k f_k\|^2) = \mathcal{O}(\|x_k - \widehat{x}\|^2)$.

From (5.6) it is also seen that $J_k N_k$ only depends on the surface and not on the parametrization in x, and consequently $J_k p_k$ is independent of the parametrization in $\mathbf{R}^n$. The gNR method is in fact the only quadratically convergent method with $J_k p_k$ independent of the parametrization. To see this we assume that there exists another method which computes $\widetilde{p}_k = -\widetilde{N}_k f_k$ and hence $J_k \widetilde{p}_k = -J_k \widetilde{N}_k f_k$. The series expansion (5.6) is unique and we have $J_k \widetilde{N}_k = J_k N_k$, which implies that $\widetilde{N}_k = N_k$.

If we define Z_1 as a matrix whose columns span the null space of J_1, we call p a *descent direction* if $p^T Z_1^T J_2^T f_2 < 0$. A drawback with both the constrained Newton method based on (5.2) and the gNR method is that a nonsingular matrix in (5.2) or (5.3) is not sufficient for p to be a descent direction. However, we use the gNR method only when we are close to the solution, see (3.20) and Algorithm 6.1, and therefore we use the gNR method undamped. From (2.2) we get λ, needed for G, *and* the Gauss–Newton search direction and if the matrix in (5.3) is singular we use the already available Gauss–Newton direction.

If we use the modified QR decomposition to solve (2.2), it is possible to reduce the size of the system in (5.3). Ignoring the permutation matrix, it is possible to rewrite (5.3) as

$$\begin{bmatrix} Q^{-1}MQ^{-T} & \begin{bmatrix} R \\ 0 \end{bmatrix} \\ \left[R^T, 0\right] & -G \end{bmatrix} \begin{bmatrix} Q^T\nu \\ p \end{bmatrix} = \begin{bmatrix} -Q^{-1}f \\ 0 \end{bmatrix}. \tag{5.7}$$

Now $QMQ^T = M$ implies that $Q^{-1}MQ^{-T} = M$ and we can reduce (5.7) to

$$\begin{bmatrix} M_n & R \\ R^T & -G \end{bmatrix} \begin{bmatrix} \eta \\ p \end{bmatrix} = \begin{bmatrix} -\xi \\ 0 \end{bmatrix}, \tag{5.8}$$

where $M_n = \mathrm{diag}(\mu_1, \dots, \mu_n)$ and η and ξ are the first n elements in $Q^T\nu$ and $Q^{-1}f$, respectively.

The matrix in (5.8) may be indefinite and we must either use a stable method for indefinite systems (see, e.g., [4]) or add some condition on the submatrices in (5.8). One possibility of the latter kind is to assume that R is well conditioned and use R^T to reduce (5.8) to

$$\begin{bmatrix} R^T & -G \\ 0 & R + M_nR^{-T}G \end{bmatrix} \begin{bmatrix} \eta \\ p \end{bmatrix} = \begin{bmatrix} 0 \\ -\xi \end{bmatrix}. \tag{5.9}$$

The solution is $p = -(R + M_nR^{-T}G)^{-1}\xi$ if the matrix $R + M_nR^{-T}G$ is nonsingular; otherwise, we take a Gauss–Newton step.

6. Computational experiments. The algorithm we use in our tests is shown below.

Algorithm 6.1.

```
$k := 1$; $Close :=$ false; $Second :=$ false
Initialize $x, V_j, j = 1, \dots, t, Tol, Maxiter$; $\beta_k := 10 \cdot Tol$
while $\beta_k > Tol$ and $k < Maxiter$
    Determine the Jacobian $J$ and the vector $f$.
    Compute the GN direction $p$ and $\lambda$ by solving (2.2).
    $\alpha := 1$; $\beta_{k+1} := \|Jp\|$; $Rate := \beta_{k+1}/\beta_k$; $GN :=$ true
    If regularization was needed then $Second :=$ false.
    If $Close$ and $Second$ and $Rate > 0.5$
        Compute the gNR direction, $p_{gNR}$, by solving (5.3).
        If the matrix in (5.3) is nonsingular then
        $p := p_{gNR}$; $GN :=$ false.
    end
    If $GN$
        Compute the merit weights by Algorithm 4.1.
        Determine the steplength $\alpha$ using the line search described
        in section (4.3) with the merit function $\phi(\alpha)$.
    end
    $x := x + \alpha p$; $Close := \beta_{k+1} \leq 2\|M\| \, \|\lambda\|$; $k := k + 1$; $Second :=$ true
end
```

To use a pure Gauss–Newton method then the variable *Second* has a fixed value of **false**.

We have tested our algorithm on three different problems described in the Appendix: Schittkowski 308 [10], Boggs 2, and Boggs 8 [2]. The intention with the tests is not to show that the algorithms are faster than other existing algorithms but to show how our algorithms handle large weights and inadequate models (ill conditioning in the linear problems). Another important aim with the tests is to verify our

Table 1
Schittkowski 308 *with the Gauss–Newton method.*

k	$\|\lambda_k - \widehat{\lambda}\|$	$\|x_k - \widehat{x}\|$	$\|J_k p_k\|$	γ_k	η_k	Ψ_k	$\max d_i^{(k)}$	α_k
1	7.6	0.55	3.0	0.55	3.0	5.0	1.0	1.0
2	2.8	0.34	0.55	0.63	0.19	2.8	3.9	0.32
3	1.6	3.1e-2	0.39	9.2e-2	0.72	3.2	13	1.0
4	0.16	1.4e-3	6.1e-2	4.6e-2	0.15	3.0	1.0	1.0
5	7.3e-3	6.6e-5	2.7e-3	4.6e-2	4.4e-2	3.0	99	1.0
6	3.4e-4	3.0e-6	1.2e-4	4.6e-2	4.6e-2	2.9	3.9	1.0
7	1.5e-5	1.4e-7	5.7e-6	4.7e-2	4.6e-2	3.0	1.0e+2	1.0
8	7.2e-7	6.5e-9	2.7e-7	4.6e-2	4.7e-2	3.0	1.0e+2	1.0
9	3.3e-8	3.0e-10	1.2e-8	4.6e-2	4.6e-2	3.0	1.0e+2	1.0
10	1.5e-9	1.4e-11	5.6e-10	4.6e-2	4.6e-2	3.0	1.0e+2	1.0
11	7.0e-11	6.3e-13	2.6e-11	4.6e-2	4.6e-2	3.0	1.0e+2	1.0

Table 2
Schittkowski 308 *with the gNR method.*

k	$\|\lambda_k - \widehat{\lambda}\|$	$\|x_k - \widehat{x}\|$	$\|J_k p_k\|$	α_k
1	7.6	0.55	3.0	1.0
2	2.8	0.34	0.55	0.32
3*	1.6	1.7e-2	0.39	1.0
4*	8.6e-2	1.2e-5	3.2e-2	1.0
5*	5.9e-5	5.6e-12	2.2e-5	1.0
6*	2.9e-11	2.3e-16	1.1e-11	1.0

theoretical results on the local convergence rate. Therefore it has been natural to use small and simple test problems.

We define $\gamma_k = \|x_{k+1} - \widehat{x}\| / \|x_k - \widehat{x}\|$ and $\eta_k = \|J_{k+1} p_{k+1}\| / \|J_k p_k\|$ as two different measures of the convergence rate for the Gauss–Newton method. We emphasize that η_k is an excellent way of estimating the convergence rate when regularization is not needed and when $\widehat{x}$ is not known.

The first problem, Schittkowski 308, is first solved with the Gauss–Newton method and the result is in Table 1. The largest weight is 10^{20} and if the weights are multiplied explicitly with f, forming $g = W^{1/2} f$, then the algorithm breaks down because of numerical instability. Note the slow growth of the merit weights. The first problem solved with the gNR method is shown in Table 2. The asterisk indicates that the gNR method was used in that step. The second problem, Boggs 2, is a constrained problem and it has been solved with the Gauss–Newton method, Table 3, and the gNR method, Table 4. All the merit weights for the Gauss–Newton method were equal to one and are not shown in Table 3. The remaining two test problems illustrate the regularization. The rank of the problem is shown under the headline Rank. In Table 5 the second test problem, Boggs 2, is solved with the Gauss–Newton method when the Jacobian is rank deficient at the starting point. In the third problem, Boggs 8, the Jacobian at the solution is rank deficient and the result is shown in Table 6.

7. Discussion.

We claim that we have developed an efficient and fairly robust algorithm for solving (1.1) (with possibly infinite weights as discussed in the introduction). However, it is difficult for us to measure the effectiveness of the algorithm because there are, to our knowledge, no other algorithms that can solve such a general

TABLE 3
Boggs 2 with the Gauss–Newton method.

k	$\Vert\lambda_k - \widehat{\lambda}\Vert$	$\Vert x_k - \widehat{x}\Vert$	$\Vert J_k p_k\Vert$	γ_k	η_k	Ψ_k	α_k
1	0.18	0.46	5.2	0.46	5.2	14	0.1
2	7.1e-2	0.22	4.6	0.47	0.88	11	0.37
3	1.2e-2	2.8e-2	2.3	0.13	0.50	2.6	1.0
4	1.1e-3	1.3e-3	0.39	4.5e-2	0.17	9.3e-2	1.0
5	2.4e-4	2.5e-4	7.3e-3	0.10	1.9e-2	1.6e-2	1.0
6	6.4e-5	6.6e-5	3.0e-4	0.27	4.2e-2	1.6e-2	1.0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
14	1.8e-9	1.8e-9	8.3e-9	0.27	0.27	1.6e-2	1.0
15	4.7e-10	4.9e-10	2.2e-9	0.27	0.27	1.6e-2	1.0
16	1.3e-10	1.3e-10	6.0e-10	0.27	0.27	1.6e-2	1.0
17	3.4e-11	3.5e-11	1.6e-10	0.27	0.27	1.6e-2	1.0
18	9.2e-12	9.5e-12	4.3e-11	0.27	0.27	1.6e-2	1.0

TABLE 4
Boggs 2 with the gNR method.

k	$\Vert\lambda_k - \widehat{\lambda}\Vert$	$\Vert x_k - \widehat{x}\Vert$	$\Vert J_k p_k\Vert$	α_k
1	0.18	0.46	5.2	0.1
2	7.1e-2	0.22	4.6	0.37
3	1.2e-2	2.8e-2	2.3	1.0
4	1.1e-3	1.3e-3	0.39	1.0
5	2.4e-4	2.5e-4	7.3e-3	1.0
6	6.4e-5	6.6e-5	3.0e-4	1.0
7*	1.7e-5	1.4e-9	8.1e-5	1.0
8*	4.4e-10	1.4e-15	1.1e-8	1.0
9*	8.2e-16	1.0e-15	1.9e-15	1.0

TABLE 5
Boggs 2, Gauss–Newton, and rank deficient at the starting point.

k	$\Vert J_k p_k\Vert$	γ_k	Ψ_k	$\max d_i^{(k)}$	α_k	Rank
1	10	10	1.1e+2	4.0	0.10	2
2	5.6	0.54	15	1.0	0.10	3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
7	1.7	0.58	1.5	1.0	0.47	3
8	0.66	0.39	0.27	1.0	0.30	3
9	0.47	0.71	0.14	1.0	0.12	3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
14	0.26	0.81	4.2e-2	1.0	0.23	3
15	0.16	0.63	2.7e-2	1.0	1.0	3
16	0.13	0.81	2.5e-2	1.0	1.0	3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
31	1.3e-10	0.27	1.6e-2	1.0	1.0	3
32	3.4e-11	0.27	1.6e-2	1.0	1.0	3

TABLE 6
Boggs 8, Gauss–Newton, and rank deficient at the solution.

k	$\|J_k p_k\|$	γ_k	Ψ_k	$\max d_i^{(k)}$	α_k	Rank
1	2.0	2.0	2.0	1.0	0.44	5
2	1.3	0.64	0.83	1.0	5.0e-2	5
⋮	⋮	⋮	⋮	⋮	⋮	⋮
28	1.1	1.0	0.65	1.0	3.4e-10	5
29	1.1	0.99	1.6	3.5	9.5e-7	4
⋮	⋮	⋮	⋮	⋮	⋮	⋮
38	0.29	0.99	0.65	3.5	1.1e-5	4
39	0.51	1.7	0.84	8.0	1.0	3
40	0.27	0.53	0.64	3.5	1.0	3
41	0.21	0.79	0.53	3.5	5.3e-5	4
⋮	⋮	⋮	⋮	⋮	⋮	⋮
53	4.2e-9	0.25	0.50	3.5	1.0	4
54	1.0e-9	0.25	0.50	3.5	1.0	3
55	2.2e-16	2.1e-7	0.50	3.5	1.0	3

problem as (1.1).

The local convergence properties are well understood for the Gauss–Newton algorithm. It is especially interesting that the local convergence results are valid for the whole problem class defined by (1.1) and that they are independent of the parametrization in $\mathbf{R}^n$.

The merit function is especially suited for our weighted and constrained problem, and our technique for choosing the merit weights is effective and does not lead to unnecessarily large weights.

As for robustness, we have shown that our algorithm is globally convergent when the iteration points are nondegenerate. It remains to find a way to regularize when the rows in J corresponding to very large weights become (almost) linearly dependent. We believe that this is a difficult and challenging problem to solve.

Appendix: Test problems. In this appendix we define our three test problems and the weight sequences. We also give the starting points, x_{start}, solutions, $\widehat{x}$, and the residuals $f(\widehat{x})$. The examples are from [10] and [2] and include unconstrained as well as constrained problems.

Schittkowski 308 [10] . This is an unconstrained problem which we have modified by incorporation of weights.

$$\begin{aligned}
f &= [\cos(x_2),\ \sin(x_1),\ x_1^2 + x_2^2 + x_1 x_2]^T \\
W &= \mathrm{diag}(10^{20}, 10^2, 1) \\
x_{start} &= [1, 1]^T \\
\widehat{x} &= [-0.036173, 1.5708]^T \\
f(\widehat{x}) &= [-1.6081 \cdot 10^{-16}, -0.036165, 2.4119]^T
\end{aligned}$$

Boggs 2 [2] . This is a constrained problem where the Jacobian is rank deficient at the second starting point, x_{start2}.

$$\begin{array}{lcl} f & = & [x_1(1+x_2^2)+x_3^4-4-3\sqrt{2},\ x_1-1,\ x_1-x_2,\ (x_2-x_3)^2]^T \\ W & = & \mathrm{diag}(\infty,1,1,1) \\ x_{start1} & = & [1,1,1]^T \\ x_{start2} & = & [1,0,0]^T \\ \widehat{x} & = & [1.1049, 1.1967, 1.5353]^T \\ f(\widehat{x}) & = & [-1.7764\cdot 10^{-15}, 0.10486, -0.091815, 0.11464]^T \end{array}$$

Boggs 8 [2] . This is a constrained problem where the Jacobian is rank deficient at the solution.

$$\begin{array}{lcl} f & = & [x_1+x_4^2-1,\ x_1^2+x_2^2-x_5^2-1,\ x_1,\ x_2,\ x_3]^T \\ W & = & \mathrm{diag}(\infty,\infty,1,1,1) \\ x_{start} & = & [1,1,1,1,1]^T \\ \widehat{x} & = & [1,0,0,-2.2352\cdot 10^{-7}, -3.2386\cdot 10^{-5}]^T \\ f(\widehat{x}) & = & [4.1949\cdot 10^{-10}, -2.0974\cdot 10^{-10}, 1, 0, 0]^T \end{array}$$

REFERENCES

[1] A. Björck, *Least Squares Methods*, Elsevier–North Holland, Amsterdam, 1988.

[2] P. Boggs and J. Tolle, *A strategy for global convergence in a sequential quadratic programming algorithm*, SIAM J. Numer. Anal., 26 (1989), pp. 600–623.

[3] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1983.

[4] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[5] M. E. Gulliksson and P.-Å. Wedin, *Modifying the QR decomposition to weighted and constrained linear least squares*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1298–1313.

[6] P. Lindström and P.-Å. Wedin, *A new linesearch algorithm for unconstrained least squares problems*, Math. Programming, 29 (1984), pp. 268–296.

[7] P. Lindström and P.-Å. Wedin, *Methods and Software for Nonlinear Least Squares Problems*, Tech. report UMINF-133.87, Inst. of Info. Proc., Univ. of Umeå, Umeå, Sweden, 1988.

[8] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[9] H. Ramsin and P.-Å. Wedin, *A comparison of some algorithms for the nonlinear least squares problem*, BIT, 17 (1977), pp. 72–90.

[10] K. Schittkowski, *More test examples for nonlinear programming codes*, Lecture Notes in Economics and Mathematical System 282, Springer-Verlag, Berlin, Heidelberg, 1987.

[11] P.-Å. Wedin, *Perturbation Results and Condition Numbers for Outer Inverses and Especially for Projections*, Tech. report UMINF 124.85, Inst. of Info. Proc., Univ. of Umeå, Umeå, Sweden, 1985.

[12] P.-Å. Wedin, *Perturbation Theory and Condition Numbers for Generalized and Constrained Linear Least Squares Problems*, Tech. report UMINF 125.85, Inst. of Info. Proc., Univ. of Umeå, Umeå, Sweden, 1985.

SIAM J. OPTIM.
Vol. 7, No. 1, pp. 225–247, February 1997

A NEW MERIT FUNCTION FOR NONLINEAR COMPLEMENTARITY PROBLEMS AND A RELATED ALGORITHM[*]

FRANCISCO FACCHINEI[†] AND JOÃO SOARES[‡]

Abstract. We investigate the properties of a new merit function which allows us to reduce a nonlinear complementarity problem to an unconstrained global minimization one. Assuming that the complementarity problem is defined by a P_0-function, we prove that every stationary point of the unconstrained problem is a global solution; furthermore, if the complementarity problem is defined by a uniform P-function, the level sets of the merit function are bounded. The properties of the new merit function are compared with those of Mangasarian–Solodov's implicit Lagrangian and Fukushima's regularized gap function. We also introduce a new simple active-set local method for the solution of complementarity problems and show how this local algorithm can be made globally convergent by using the new merit function.

Key words. nonlinear complementarity problem, merit function, semismoothness, global convergence, quadratic convergence

AMS subject classifications. Primary, 90C33; Secondary, 65K05, 90C30

PII. S1052623494279110

1. Introduction.

We consider the nonlinear complementarity problem

$$\text{(NC)} \qquad F(x) \geq 0, \qquad x \geq 0, \qquad F(x)^T x = 0,$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable everywhere. Recent research on the numerical solution of problem (NC) has focused on the development of *globally* convergent algorithms. To this end, two approaches have been investigated: the reformulation of the nonlinear complementarity problem as a minimization one and the use of continuation methods. Strictly related to the first approach is the equation-reduction approach, which tries to solve problem (NC) by solving an equivalent system of equations, while interior-point methods are close to the continuation approach. There exists a considerable body of literature on the theoretical properties of continuation methods, and interior-point methods appear to be valuable in the practical solution of linear complementarity problems. However, in past years the minimization approach seems to have raised much more interest, and most if not all of the proposals and developments of practical algorithms for the solution of nonlinear complementarity problems follow this approach.

The minimization approach is based on the introduction of a *merit function* whose (possibly constrained) global minima are the solutions of the nonlinear complementarity problem; the latter problem is then solved by means of suitable minimization algorithms. The definition of a merit function is often, if not always, based on a preliminary equation reformulation of the complementarity problem. More precisely, one first defines a system of equations $H(x) = 0$, whose solutions coincide with the solutions of the complementarity problem, and then uses as merit function $\|H(x)\|^2$

[*] Received by the editors December 28, 1994; accepted for publication (in revised form) September 19, 1995.
http://www.siam.org/journals/siopt/7-1/27911.html

[†] Università di Roma "La Sapienza," Dipartimento di Informatica e Sistemistica, Via Buonarroti 12, 00185 Roma, Italy (soler@dis.uniroma1.it).

[‡] Columbia University, Graduate School of Business, 804 Uris Hall, New York, NY 10027 (jsoares@mat.uc.pt).

(or $\|H(x)\|$). Before continuing our discussion we give a formal definition of merit function.

DEFINITION 1.1. *Let $C \subseteq \mathbb{R}^n$ be given. A merit function for problem* (NC) *is a nonnegative function $M : C \to \mathbb{R}$ such that $\bar{x}$ is a solution of problem* (NC) *iff $\bar{x} \in C$ and $M(\bar{x}) = 0$, i.e., iff the solutions of problem* (NC) *coincide with the* global *solutions of the problem*

$$\text{(PM)} \qquad \min\ M(x), \quad x \in C,$$

with zero optimal value.

Note that if the complementarity problem has no solution then a merit function must either have global solutions with positive objective value or no global solutions at all.

It is not difficult to find a merit function for problem (NC); the challenging task is to find a merit function which enjoys useful properties from the computational point of view. For example, one could consider the merit function $M(x) = F(x)^T x$, whose global minimizers on the set $C := \{x | x \geq 0, F(x) \geq 0\}$ are the solutions of the complementarity problem (NC). But seeking these global minimizers is not easy because, even in very simple cases, the structure of C may be very complicated and the minimization problem can have stationary points which are not global solutions. There have been several proposals of merit functions (or equation reformulations); the seminal work is [21], where a smooth equation reformulation is given. Other papers related to smooth reformulations include, e.g., [8, 12, 13, 16, 17, 18, 23, 25, 38]; nonsmooth reformulations, instead, are used in, e.g., [5, 9, 14, 28, 34, 37, 41, 42]. It is often difficult, if at all possible, to compare different merit functions; however, we think that the main points which should be considered when assessing a merit function M are as follows:

1. the conditions under which every stationary point of problem (PM) is a global solution;
2. the conditions under which the level sets $L(\alpha) := \{x \in \mathbb{R}^n : x \in C, M(x) \leq \alpha\}$ are bounded;
3. the degree of smoothness of M;
4. the structure of the set C.

Obviously, all of these points have a great practical significance. The numerical performance of algorithms based on problem (PM) should also be considered, even if one should always keep in mind that the numerical results are also dependent on the particular algorithm chosen to solve problem (PM).

There is generally a trade-off between simplicity of problem (PM) and its properties. In what concerns the function M, differentiable merit functions tend to be more ill conditioned than nondifferentiable ones and do not generally allow us to develop superlinearly convergent algorithms for degenerate problems. On the other hand, nondifferentiable merit functions do not have these drawbacks but generally require ad hoc complex minimization algorithms. In what concerns the set C, constrained reformulations are usually valid under weaker assumptions than their unconstrained counterparts, but solving a constrained minimization problem is more difficult than solving an unconstrained one.

In order to put this work in perspective and also to illustrate the points discussed above, we now briefly recall the properties of two recently proposed merit functions: the *implicit Lagrangian* of Mangasarian and Solodov [23] and Fukushima's *regularized gap function* [12] (see also [2]). These two merit functions are, in our opinion, among the most interesting proposals in the field.

The implicit Lagrangian is defined as follows (to simplify we have fixed a free parameter):

$$M_{ms}(x) := x^T F(x) + \frac{1}{4}\left(||[x-2F(x)]_+||^2 - \|x\|^2 + ||[F(x)-2x]_+||^2 - \|F(x)\|^2\right),$$

where $[x]_+ = \max(x,0)$, taken componentwise. M_{ms} is a merit function with $C = \mathbb{R}^n$, so that solving (NC) is equivalent to finding the unconstrained global solutions of the problem $\{\min\ M_{ms}(x)\}$. Furthermore, the merit function M_{ms} enjoys the following properties.

- M_{ms} is continuously differentiable.
- If the Jacobian of F is a positive definite matrix for every x then every stationary point of problem (PM) is a global minimum point of problem (PM) [43].
- If F is strongly monotone and globally Lipschitzian then the level sets $L(\alpha)$ are bounded [43].

The regularized gap function of Fukushima is defined for variational inequalities. When specialized to nonlinear complementarity problems it becomes the following (to simplify, we have fixed a free parameter):

$$M_{fa}(x) := x^T F(x) + \frac{1}{2}\left(||[x-F(x)]_+||^2 - \|x\|^2\right).$$

M_{fa} is a merit function with $C = \mathbb{R}^n_+$, so solving (NC) is equivalent to finding the global solutions of the simply constrained minimization problem $\{\min\ M_{fa}(x) : x \in \mathbb{R}^n_+\}$. Furthermore, the merit function M_{fa} enjoys the following properties.

- M_{fa} is continuously differentiable.
- If the Jacobian of F is a positive definite matrix for every x in C, then every stationary point of Problem (PM) is a global minimum point of problem (PM) [12].
- If F is strongly monotone then the level sets $L(\alpha)$ are bounded [39].

We note that the implicit Lagrangian merit function is simpler than the regularized gap function, since it only requires an unconstrained minimization, but the condition for having bounded level sets is stronger for the implicit Lagrangian than for the regularized gap function.

The purpose of this paper is twofold: on the one hand, we study a new merit function which can be used to reformulate the nonlinear complementarity problem as a *smooth, unconstrained* minimization problem; on the other hand, we propose a globally convergent algorithm for the solution of problem (NC) and study its theoretical properties.

The new merit function is based on the following two-variable convex function:

$$\phi(a,b) := \sqrt{a^2+b^2} - (a+b).$$

The most interesting property of this function is that, as is easily verified,

$$\phi(a,b) = 0 \qquad \Longleftrightarrow \qquad a \geq 0,\ b \geq 0,\ ab = 0. \tag{1}$$

Note also that ϕ is continuously differentiable everywhere but in the origin. The function ϕ was introduced by Fischer [10] in 1992; since then it has attracted the attention of many researchers, and it has proved to be a valuable tool [7, 11, 13, 16, 18, 32, 40].

By exploiting (1), it is readily seen that the following system of nonsmooth equations is equivalent to the nonlinear complementarity problem:

$$\Phi(x) = \begin{bmatrix} \phi(x_1, F_1(x)) \\ \vdots \\ \phi(x_i, F_i(x)) \\ \vdots \\ \phi(x_n, F_n(x)) \end{bmatrix} = 0.$$

It is then obvious that the function

$$\Psi(x) := \frac{1}{2}\|\Phi(x)\|^2 = \frac{1}{2}\sum_{i=1}^{n} \phi(x_i, F_i(x))^2$$

is a merit function with $C = \mathbb{R}^n$, so solving (NC) is equivalent to finding the unconstrained global solutions of the problem $\{\min\ \Psi(x)\}$. We shall prove that the merit function Ψ enjoys the following properties.

- Ψ is continuously differentiable; furthermore, if every F_i is an SC^1 function, then Ψ is also an SC^1 function (we recall that this means that Ψ is continuously differentiable and its gradient is semismooth; see section 2 for a formal definition).
- If F is a P_0-function then every stationary point of problem (PM) is a global minimum point of problem (PM).
- If F is a uniform P-function then the level sets $L(\alpha)$ are bounded.

Furthermore, we should also add that Φ is semismooth (see [24, 33]), and this is a significant analytical property; in particular, we note that semismoothness is a stronger and more far reaching property than B-differentiability, the latter being a property often used in recent years in the study of nonlinear complementarity problems [14, 28, 41, 42].

The theoretical properties of Ψ seem to be superior to those of the implicit Lagrangian and of the regularized gap function. In fact, the function Ψ allows us to solve the nonlinear complementarity problem by an *unconstrained* minimization, and the conditions under which every stationary point of the merit function is a global minimizer and the level sets are bounded are substantially weaker. Actually, as pointed out by one of the referees, both the implicit Lagrangian and the new merit function are in the same order of the natural residual [20, 40] so that also the level sets of the implicit Lagrangian are bounded if F is a uniform P-function. In this particular respect the function Ψ simply appears to be easier to analyze than the implicit Lagrangian.

Also, the differentiability properties of Ψ seem more interesting, and actually the semismoothness of Φ and the SC^1 property of Ψ are very important from an algorithmic point of view [6, 24, 29, 30, 31, 33]. It is worth noting that the system $\Phi(x) = 0$ is nonsmooth, but the merit function $\Psi(x) = \frac{1}{2}\|\Phi(x)\|^2$ is, surprisingly, smooth. Thus our reformulation of the complementarity problem as a minimization one seems to inherit the advantages of both nonsmooth and smooth merit functions, while mitigating their drawbacks. In particular we note that since Ψ is continuously differentiable, it is very easy to force global convergence of algorithms by using the gradient of the merit function. On the other hand, we are able to prove, for the first time in the case of smooth merit functions, quadratic convergence even to degenerate solutions.

The merit function Ψ has also been independently introduced by Geiger and Kanzow [13]. Their results are, however, weaker than those reported above or simply different. In particular they showed that every stationary point of the merit function is a global minimum point if F is monotone, while the level sets of Ψ are bounded if F is strongly monotone. Their analysis of the differential properties of Ψ is cruder than ours and, to define superlinear convergent algorithms for the solution of the complementarity problem, they require the solutions to be nondegenerate. On the other hand, Geiger and Kanzow describe an interesting algorithm for the solution of strongly monotone complementarity problems which does not require the evaluation of the Jacobian of F.

In this paper we also illustrate the usefulness of the merit function through the description of a technique for globalizing a local algorithm for the solution of complementarity problems. The local algorithm itself is, we think, worthy of attention; it is an active-set algorithm which reduces the solution of the complementarity problem to the solution of a lower-dimensional system of smooth equations by Newton's method. This local algorithm is quadratically convergent under a mild assumption, which is weaker than the classical regularity assumption required by the method of Robinson [36] and Josephy [15]; in particular it does not require nondegeneracy of the solution, as opposed to the methods of [8, 12, 13, 14, 16, 17, 23, 25, 38]. Furthermore, it requires just the solution of a reduced linear system at each iteration. The local algorithm is globalized in a very cheap and simple way by using the merit function Ψ. We show that the overall algorithm is globally convergent and, under appropriate, mild assumptions, eventually reduces to the local, fast algorithm, thus retaining its convergence rate. Furthermore, the algorithm is finitely convergent on a wide class of linear complementarity problems. The numerical behavior of the algorithm is illustrated in [7] and the results reported there show that the algorithm is quite promising.

This paper is organized as follows. In the next section we recall various definitions related to complementarity problems and to differentiability of functions. In section 3 we analyze the differential properties of Φ and Ψ, while in section 4 we prove the main properties of the function Ψ. A local algorithm for the solution of problem (NC) and its globalization through the merit function Ψ are discussed in section 5. Finally, in the last section we make some conclusive remarks.

We close this section by giving a list of the notation employed.

If $f\colon \mathbb{R}^n \to \mathbb{R}$ is differentiable at x then the column vector $\nabla f(x)$ is the gradient of the function f at the point x. If f is a locally Lipschitz function at x then the set of column vectors $\partial f(x)$ is the set of subgradients of f at x, i.e., the generalized gradient.

If $F\colon \mathbb{R}^n \to \mathbb{R}^m$ is differentiable at x then the $m \times n$ matrix $\nabla F(x)$ is a matrix whose ith column is the gradient of F_i at the point x. If F is a Lipschitz mapping at x then the set of $m \times n$ matrices $\partial F(x)$ is the generalized Jacobian of F at x.

We remark that, as usual, there is an inconsistency in this notation. If F is single valued then the generalized Jacobian is a set of row vectors and does not coincide with the generalized gradient but with the set of all the transpose subgradients. Also note that if F is differentiable then its generalized Jacobian is not $\nabla F(x)$ but $\nabla F(x)^T$.

The standard notation used in nonsmooth analysis for "operations" between sets is also used here. In particular, if A and B are sets of n-dimensional vectors then

$$A + B = \{c \in \mathbb{R}^n | c = a + b \quad \text{with } a \in A, b \in B\}.$$

If A is a set of $n \times n$ matrices and B is a set of n-vectors,

$$AB = \{c \in \mathbb{R}^n | c = ab \quad \text{with } a \in A, b \in B\}.$$

The Euclidean norm is denoted by $\|\cdot\|$, and $S(\bar{x}, \delta) \subseteq \mathbb{R}^n$ denotes the closed Euclidean sphere of center $\bar{x}$ and radius δ, i.e., $S(x, \delta) = \{x \in \mathbb{R}^n | \|x - \bar{x}\| \leq \delta\}$. If Ω is a nonempty subset of $\mathbb{R}^n$, $dist\{x|\Omega\} := \inf_{y \in \Omega} \|y - x\|$ denotes the (Euclidean) distance of x to Ω.

If M is an $n \times n$ matrix with elements M_{ij}, $i, j = 1, \ldots n$, and I and J are index sets such that $I, J \subseteq \{1, \ldots n\}$, we denote by M_{IJ} the $|I| \times |J|$ submatrix of M consisting of elements M_{ij}, $i \in I$, $j \in J$. If w is an n vector, we denote by w_I the subvector with components w_i, $i \in I$.

2. Background material. In this section we review some definitions related to nonlinear complementarity problems and to differential properties of functions which will be used in the sequel.

A solution to the nonlinear complementarity problem (NC) is a vector $\bar{x} \in \mathbb{R}^n$ such that

$$F(\bar{x}) \geq 0, \qquad \bar{x} \geq 0, \qquad F(\bar{x})^T \bar{x} = 0.$$

We define the following three index sets which are associated with the solution $\bar{x}$

$$\alpha := \{i | \bar{x}_i > 0\}, \qquad \beta := \{i | \bar{x}_i = 0 = F_i(\bar{x})\}, \qquad \gamma := \{i | F_i(\bar{x}) > 0\}.$$

The solution $\bar{x}$ is said to be nondegenerate if $\beta = \emptyset$.

In the following definition, we introduce two notions of regularity which play a central role in our analysis and which have also been widely used in the analysis of nonlinear complementarity problems.

DEFINITION 2.1. *We say that the solution $\bar{x}$ is*

- b-regular *if, for every index set δ such that $\alpha \subseteq \delta \subseteq \alpha \cup \beta$, the principal submatrix $\nabla F_{\delta\delta}(\bar{x})$ is nonsingular*;
- R-regular *if $\nabla F_{\alpha\alpha}(\bar{x})$ is nonsingular and the Schur complement of $\nabla F_{\alpha\alpha}(\bar{x})$ in*

$$\begin{pmatrix} \nabla F_{\alpha\alpha}(\bar{x}) & \nabla F_{\alpha\beta}(\bar{x}) \\ \nabla F_{\beta\alpha}(\bar{x}) & \nabla F_{\beta\beta}(\bar{x}) \end{pmatrix}$$

is a P-matrix (see below).

We recall that the above mentioned Schur complement is defined by

$$\nabla F_{\beta\beta}(\bar{x}) - \nabla F_{\beta\alpha}(\bar{x}) \nabla F_{\alpha\alpha}(\bar{x})^{-1} \nabla F_{\alpha\beta}(\bar{x}).$$

Note that R-regularity coincides with the notion of regularity introduced by Robinson in [36] (see also [35], where the same condition is called strong regularity) and is strictly related to similar conditions used, e.g., in [9, 25, 28]. If $\bar{x}$ is a nondegenerate solution then the b-regularity condition can be equivalently stated as follows: the vectors $\nabla F_i(\bar{x})$, $i \in \alpha$, and e_i, $i \in \gamma$ are linearly independent (e_i indicates the ith column of the identity matrix); b-regularity has been employed, e.g., in [16, 23, 25]. It is known that R-regularity implies b-regularity [28] and local uniqueness of the solution $\bar{x}$ [35]; furthermore, b-regularity also implies the local uniqueness of the solution $\bar{x}$; see [19] or Proposition 5.4.

We make use of the following linear algebra definitions and properties.

DEFINITION 2.2. *A matrix $M \in \mathbb{R}^{n \times n}$ is a*

- *P_0-matrix if every of its principal minors is nonnegative*;
- *P-matrix if every of its principal minors is positive*;
- *R_0-matrix if the linear complementarity problem*

$$Mx \geq 0, \quad x \geq 0, \quad x^T Mx = 0,$$

has 0 as its unique solution.

It is obvious that every P-matrix is also a P_0-matrix, and it is known [4] that every P-matrix is an R_0-matrix. We shall also need the following characterization of P_0-matrices [4].

Proposition 2.3. *A matrix $M \in \mathbb{R}^{n \times n}$ is a P_0-matrix iff for every nonzero vector x there exists an index i such that $x_i \neq 0$ and $x_i(Mx)_i \geq 0$.*

We need the following concepts which concern nonlinear functions.

Definition 2.4. *A function $F : \mathbb{R}^n \to \mathbb{R}^n$ is a*

- *P_0-function if, for every x and y in $\mathbb{R}^n$ with $x \neq y$, there is an index i such that*

$$x_i \neq y_i, \qquad (x_i - y_i)[F_i(x) - F_i(y)] \geq 0;$$

- *P-function if, for every x and y in $\mathbb{R}^n$ with $x \neq y$, there is an index i such that*

$$(x_i - y_i)[F_i(x) - F_i(y)] > 0;$$

- *uniform P-function if there exists a positive constant μ such that, for every x and y in $\mathbb{R}^n$, there is an index i such that*

$$(x_i - y_i)[F_i(x) - F_i(y)] \geq \mu \|y - x\|^2;$$

- *monotone function if, for every x and y in $\mathbb{R}^n$,*

$$(x - y)^T [F(x) - F(y)] \geq 0;$$

- *strictly monotone function if, for every x and y in $\mathbb{R}^n$ with $x \neq y$,*

$$(x - y)^T [F(x) - F(y)] > 0;$$

- *strongly monotone function if there is a positive constant μ such that, for every x and y in $\mathbb{R}^n$,*

$$(x - y)^T [F(x) - F(y)] \geq \mu \|y - x\|^2.$$

It is obvious that every monotone function is a P_0-function, every strictly monotone function is a P-function, and every strongly monotone function is a uniform P-function. Furthermore, it is known that the Jacobian of every continuously differentiable P_0-function is a P_0-matrix [26] and that if the Jacobian of a continuously differentiable function is a P-matrix for every x, then the function is a P-function [26]. If F is affine (that is, if $F(x) = Mx + q$) then F is a P_0-function iff M is a P_0-matrix, while F is a (uniform) P-function iff M is a P-matrix (note that in the affine case, the concept of uniform P-function and P-function coincide).

In the remaining part of this section we recall some basic definitions about semismoothness and SC^1 functions.

Semismooth functions were introduced in [24] and they immediately showed to be relevant to optimization algorithms. Recently, the concept of semismoothness has been extended to vector-valued functions [33].

Definition 2.5. *Let $F : \mathbb{R}^n \to \mathbb{R}^m$ be locally Lipschitz at $x \in \mathbb{R}^n$. We say that F is* semismooth *at x if*

$$\lim_{\substack{H\in\partial F(x+tv') \\ v'\to v, t\downarrow 0}} Hv' \tag{2}$$

exists for any $v \in \mathbb{R}^n$.

Semismooth functions lie between Lipschitz functions and C^1 functions. Note that this class is strictly contained in the class of B-differentiable functions.

It is known (see [24, 33]) that

(a) continuously differentiable functions and convex functions are semismooth. The composites of semismooth functions are semismooth;

(b) if a function F is semismooth at x, then F is directionally differentiable at x, and the directional derivative $F'(x; d)$ is equal to the limit (2).

We can now give the definition of SC^1 function.

Definition 2.6. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be an SC^1 function if f is continuously differentiable and its gradient is semismooth.*

SC^1 functions can be viewed as functions which lie between C^1 and C^2 functions. Semismooth systems of equations form an important class, since they often occur in practice and many of the classical methods for their solution (e.g., Newton's method) can be extended to solve such problems [29, 31, 33]. Analogously, many classical results concerning the minimization of C^2 functions can be extended to the minimization of SC^1 functions (see, e.g., [6, 30] and references therein), which, in turn, play an important role in many optimization problems. Under very mild differentiability assumptions on F, the new merit function we will introduce in the next section is an SC^1 function.

3. Differential results.

In this section we study the differential properties of Φ and Ψ. In particular, we give an estimate of the generalized Jacobian of Φ and a sufficient condition for the nonsingularity of *all* its elements at a solution of (NC). We also establish that Φ is semismooth, Ψ is continuously differentiable, and Ψ is SC^1 if F is an SC^1 function. Unless stated otherwise, we assume that F is everywhere continuously differentiable.

Proposition 3.1.

$$\partial\Phi(x)^T \subseteq (A(x) - I) + \nabla F(x)(B(x) - I), \tag{3}$$

where I is the $n \times n$ identity matrix and $A(x)$ and $B(x)$ are possibly multivalued $n \times n$ diagonal matrices whose ith diagonal element is given by

$$A_{ii}(x) = \frac{x_i}{\|(x_i, F_i(x))\|}, \quad B_{ii}(x) = \frac{F_i(x)}{\|(x_i, F_i(x))\|}$$

if $(x_i, F_i(x)) \neq 0$ and by

$$A_{ii}(x) = \xi_i, \quad B_{ii}(x) = \rho_i \quad \textit{for every } (\xi_i, \rho_i) \textit{ such that } \|(\xi_i, \rho_i)\| \leq 1$$

if $(x_i, F_i(x)) = 0$.

Proof. By known rules on the evaluation of the generalized Jacobian (see [3], Proposition 2.6.2 (e)),

$$\partial\Phi(x)^T \subseteq (\partial\Phi_1(x) \times \cdots \times \partial\Phi_n(x)).$$

If i is such that $(x_i, F_i(x)) \neq 0$, then it is easy to check that $\Phi_i(x)$ is differentiable and

$$\nabla\Phi_i(x) = \nabla\phi(x_i, F_i(x)) = \left(\frac{x_i}{\|x_i, F_i(x)\|} - 1\right) e_i + \nabla F_i(x) \left(\frac{F_i(x)}{\|x_i, F_i(x)\|} - 1\right).$$

If i is such that $(x_i, F_i(x)) = 0$, by using the theorem on the generalized gradient of a composite function (see [3], Theorem 2.3.9 (iii)) and recalling that

$$\partial\|0,0\| = \{(\xi_i, \rho_i) : \|(\xi_i, \rho_i)\| \leq 1\},$$

we get

$$\partial\Phi_i(x) = \partial\phi(x_i, F_i(x)) = (\xi_i - 1)e_i + \nabla F_i(x)(\rho_i - 1).$$

From these equalities the proposition easily follows. □

By exploiting estimate (3), it is now possible to give a sufficient condition for the nonsingularity of all the elements in the generalized Jacobians of Φ at a solution of the nonlinear complementarity problem. This result is important from the algorithmic point of view; see section 5 and [33].

PROPOSITION 3.2. *Suppose that $\bar{x}$ is an R-regular solution of problem* (NC). *Then every matrix in $\partial\Phi(\bar{x})$ is nonsingular.*

Proof. Using expression (3) and taking into account that $\bar{x}$ is a solution of the nonlinear complementarity problem, any matrix C belonging to $\partial\Phi(x)^T$ can be written in the following partitioned form:

$$C = \begin{pmatrix} -\nabla F_{\alpha\alpha} & \nabla F_{\alpha\beta}(B_{\beta\beta} - I_{\beta\beta}) & 0_{\alpha\gamma} \\ -\nabla F_{\beta\alpha} & \nabla F_{\beta\beta}(B_{\beta\beta} - I_{\beta\beta}) + (A_{\beta\beta} - I_{\beta\beta}) & 0_{\beta\gamma} \\ -\nabla F_{\gamma\alpha} & \nabla F_{\gamma\beta}(B_{\beta\beta} - I_{\beta\beta}) & -I_{\gamma\gamma} \end{pmatrix}. \tag{4}$$

It is easy to see that these C are nonsingular iff the "left upper corner"

$$G = \begin{pmatrix} -\nabla F_{\alpha\alpha} & \nabla F_{\alpha\beta}(B_{\beta\beta} - I_{\beta\beta}) \\ -\nabla F_{\beta\alpha} & \nabla F_{\beta\beta}(B_{\beta\beta} - I_{\beta\beta}) + (A_{\beta\beta} - I_{\beta\beta}) \end{pmatrix} \tag{5}$$

is nonsingular. Showing that the matrix G is nonsingular is equivalent to showing that the only solution of the system

$$-Gy = -G\begin{pmatrix} y_\alpha \\ y_\beta \end{pmatrix} = 0$$

is the zero vector (we have changed sign for simplicity). This system can be rewritten as

$$\begin{cases} \nabla F_{\alpha\alpha} y_\alpha + \nabla F_{\alpha\beta}(I_{\beta\beta} - B_{\beta\beta}) y_\beta = 0, \\ \nabla F_{\beta\alpha} y_\alpha + \nabla F_{\beta\beta}(I_{\beta\beta} - B_{\beta\beta}) y_\beta = -(I_{\beta\beta} - A_{\beta\beta}) y_\beta, \end{cases}$$

from which, recalling that $\nabla F_{\alpha\alpha}$ is nonsingular by the R-regularity assumption, we obtain, solving the first equation with respect to y_α and substituting into the second equation,

$$\begin{cases} y_\alpha = -\nabla F_{\alpha\alpha}^{-1} \nabla F_{\alpha\beta}(I_{\beta\beta} - B_{\beta\beta}) y_\beta, \\ (\nabla F_{\beta\beta} - \nabla F_{\beta\alpha} \nabla F_{\alpha\alpha}^{-1} \nabla F_{\alpha\beta})(I_{\beta\beta} - B_{\beta\beta}) y_\beta = -(I_{\beta\beta} - A_{\beta\beta}) y_\beta, \end{cases} \tag{6}$$

where $(\nabla F_{\beta\beta}-\nabla F_{\beta\alpha}\nabla F_{\alpha\alpha}^{-1}\nabla F_{\alpha\beta})=(G/\nabla F_{\alpha\alpha})$ is by definition the Schur complement of $\nabla F_{\alpha\alpha}$ in G and is hence a P-matrix by the R-regularity assumption. Then, showing the nonsingularity of G is equivalent to showing that the only vector which solves the second equation of (6), i.e.,

$$(G/\nabla F_{\alpha\alpha})(I_{\beta\beta}-B_{\beta\beta})y_\beta=-(I_{\beta\beta}-A_{\beta\beta})y_\beta, \tag{7}$$

is $y_\beta=0$. We proceed by contradiction, assume that there exists a solution $y_\beta \neq 0$, and consider two cases.

(1) $(I_{\beta\beta}-B_{\beta\beta})y_\beta=0$. Define $I=\{i : (y_\beta)_i \neq 0\}$. Note that $I \neq \emptyset$ because we are assuming $y_\beta \neq 0$. This means that $B_{ii}=1$ for every $i \in I$, which, in turn, implies $A_{ii}=0$ for every $i \in I$ by the definition of the matrices A and B. Hence $-(I_{\beta\beta}-A_{\beta\beta})y_\beta \neq 0$ and this is absurd.

(2) $(I_{\beta\beta}-B_{\beta\beta})y_\beta \neq 0$. The components of $(I_{\beta\beta}-B_{\beta\beta})y_\beta$ and $-(I_{\beta\beta}-A_{\beta\beta})y_\beta$ which are both nonzero (if any) have opposite signs. This implies, by (7),

$$[(I_{\beta\beta}-B_{\beta\beta})y_\beta]_i\,[(G/\nabla F_{\alpha\alpha})(I_{\beta\beta}-B_{\beta\beta})y_\beta]_i \leq 0, \quad \forall i \in \beta.$$

Since $(G/\nabla F_{\alpha\alpha})$ is a P-matrix this is only possible if $(I_{\beta\beta}-B_{\beta\beta})y_\beta=0$, again we have a contradiction and the proof is complete. □

Another important property of Φ is that it is a semismooth function. Also, this property is very important from the computational point of view.

PROPOSITION 3.3. *The function Φ is semismooth.*

Proof. The function Φ is semismooth iff every of its components is semismooth [33]. But $\Phi_i(x)$ is the composite of the convex function $\phi : \mathbb{R}^2 \to \mathbb{R}$ and of the differentiable function $(x_i, F_i(x))^T : \mathbb{R}^n \to \mathbb{R}^2$. Since convex and differentiable functions are semismooth and the composite of semismooth functions is semismooth, the proposition is proved. □

We now pass to consider the differential properties of the function Ψ. The first result is somewhat surprising and states that Ψ is continuously differentiable.

PROPOSITION 3.4. *The function $\Psi(x)$ is continuously differentiable and its gradient is $\partial\Phi(x)^T\Phi(x)$.*

Proof. By known rules on the calculus of generalized gradients (see [3], Theorem 2.6.6), it holds that $\partial\Psi(x)=\partial\Phi(x)^T\Phi(x)$. Since it is easy to check that $\partial\Phi(x)^T\Phi(x)$ is single valued everywhere because the zero components of $\Phi(x)$ cancel the "multivalued columns" of $\partial\Phi(x)^T$, we have by the corollary to Theorem 2.2.4 in [3] that $\Psi(x)$ is continuously differentiable. □

The second result about the differentiability properties of Ψ is that if F is SC^1 then Ψ is also SC^1. This result will not be explicitly used in this paper, but we think that it is of great significance and that it also explains the good numerical behavior of algorithms based on the merit function Ψ.

PROPOSITION 3.5. *If every F_i is an SC^1 function, then $\Psi(x)$ is an SC^1 function.*

Proof. The function $\Psi=\frac{1}{2}\sum_{i=1}^n \phi(x_i,F_i)^2$ is SC^1 if every

$$\frac{1}{2}\phi^2=(x_i^2+F_i^2+x_iF_i)-(x_i+F_i)\|x_i,F_i\|$$

is SC^1. It is obvious that it is sufficient to show that the term $(x_i+F_i)\|x_i,F_i\|$ is SC^1. It is easy to check that $(x_i+F_i)\|x_i,F_i\|$ is continuously differentiable and that its gradient is

$$\begin{cases} (e_i+\nabla F_i)\,\|x_i,F_i\|+(x_i+F_i)\left[\left(\frac{x_i}{\|x_i,F_i\|}-1\right)e_i+\left(\frac{F_i}{\|x_i,F_i\|}-1\right)\nabla F_i\right] \\ \qquad\qquad \text{if } (x_i,F_i)\neq(0,0), \\ 0 \\ \qquad\qquad \text{if } (x_i,F_i)=(0,0). \end{cases}$$

Again, to check that this gradient is semismooth, we only need to check that every component is semismooth. This, in turn, reduces to checking that the "troublesome" terms

$$\begin{cases} \frac{(x_i+F_i)F_i}{\|x_i,F_i\|} & \text{if } (x_i,F_i)\neq(0,0), \\ 0 & \text{if } (x_i,F_i)=(0,0) \end{cases}$$

and

$$\begin{cases} \frac{(x_i+F_i)x_i}{\|x_i,F_i\|} & \text{if } (x_i,F_i)\neq(0,0), \\ 0 & \text{if } (x_i,F_i)=(0,0) \end{cases}$$

are semismooth (note that the term $\|x_i,F_i\|$ is semismooth since it is the composite of the convex and hence semismooth, norm function, and semismooth functions). We will check this only for the first term; the proof for the second one is analogous.

Since the composite of semismooth functions is semismooth we only have to show that

$$\eta(a,b)=\begin{cases} \frac{(a+b)b}{\|a,b\|} & \text{if } (a,b)\neq(0,0), \\ 0 & \text{if } (a,b)=(0,0) \end{cases}$$

is semismooth. First we show that it is locally Lipschitzian. This is obvious everywhere but in the origin, so let us consider this point. We first note that

$$(8)\quad \left|\frac{(a+b)b}{\|a,b\|}-0\right|=\frac{|(a+b)b|}{\|a,b\|}=\frac{|(a+b)|\,|b|}{\|a,b\|}\leq\sqrt{2}\frac{\|a,b\|\|a,b\|}{\|a,b\|}=\sqrt{2}\|(a,b)-(0,0)\|.$$

Furthermore, in points different from the origin, it is readily seen that $\nabla\eta(c,d)$ is given by

$$(9)\qquad \nabla\eta(c,d)=\begin{bmatrix} \frac{d\|c,d\|-(cd+d^2)\frac{c}{\|c,d\|}}{\|c,d\|^2} \\ \frac{(c+2d)\|c,d\|-(cd+d^2)\frac{d}{\|c,d\|}}{\|c,d\|^2} \end{bmatrix},$$

and it is easy to verify that the norm of $\nabla\eta(c,d)$ is bounded on any bounded set which does not contain the origin.

Consider now a convex, open bounded neighborhood Ω of the origin. We want to show that there exists a positive constant L such that for every pair of points y and z belonging to Ω we have $|\eta(z)-\eta(y)|\leq L\|z-y\|$. To this end we consider two cases.

(a) The origin does not belong to the closed segment $[y,z]$. In this case we can apply the theorem of the mean and obtain

$$|\eta(z)-\eta(y)|\leq|\eta(y)+\nabla\eta(w)^T(z-y)-\eta(y)|\leq M\|z-y\|,$$

where w is a point belonging to the open segment (z, y) and M is any positive constant majorizing the norm of the gradient of η on the bounded set $\Omega \setminus \{0\}$.

(b) The origin belongs to the closed segment $[y, z]$. In this case we have $\|z - y\| = \|z\| + \|y\|$, so that, exploiting (8), we can write

$$|\eta(z) - \eta(y)| \leq |\eta(z) - 0| + |\eta(y) - 0| \leq \sqrt{2}(\|z\| + \|y\|) = \sqrt{2}\|z - y\|.$$

Hence the local Lipschitzianity of η in the origin is proved with $L = \max\{\sqrt{2}, M\}$.

To check semismoothness we also only have to check semismoothness in $(0, 0)$, since in other points $\eta(a, b)$ is continuously differentiable and hence semismooth. To check semismoothness in $(0, 0)$ we employ Theorem 2.3 (iv) in [33] which states that the locally Lipschitzian function η is semismooth at $(0, 0)$ iff, for every $\zeta \in \partial\eta((0, 0) + (c, d))$ with $(c, d) \to 0$, it holds that

$$\zeta^T \begin{pmatrix} c \\ d \end{pmatrix} - \eta'((0, 0); (c, d)) = o(\|(c, d)\|). \tag{10}$$

To this end we first note that it is easy to check, using the very definition of directional derivative, that

$$\eta'((0, 0); (c, d)) = \eta(c, d). \tag{11}$$

Furthermore, taking into account that for every $(c, d) \neq (0, 0)$, $\eta((0, 0) + (c, d))$ is differentiable, the vector ζ in the theorem of Qi–Sun reduces to $\nabla\eta(c, d)$. Employing (11) and (9), it is now easy to check that the left-hand side of (10) is identically 0, so η is semismooth and the proof is complete. □

4. Properties of Ψ.

4. Properties of Ψ. In this section we prove two important results on the function Ψ. The first result states that if F is a P_0-function then every point such that $\nabla\Psi(x) = 0$ is a global minimum point of Ψ; the second result establishes that if F is a uniform P-function then Ψ has bounded level sets.

The importance of these two properties and relations to similar results in the literature have already been discussed in the introduction.

THEOREM 4.1. *Suppose that F is a P_0-function. Then every stationary point of Ψ is such that $\Psi(x) = 0$.*

Proof. Suppose that $\nabla\Psi(x) = 0$. This means that

$$[(A(x) - I) + \nabla F(x)(B(x) - I)]\, \Phi(x) = 0; \tag{12}$$

we want to show that $\Phi(x) = 0$.

Suppose the contrary. Consider the vector $(B(x) - I)\Phi(x)$. By its structure, it is easy to see that its ith component is different from 0 iff $\Phi_i(x) \neq 0$. In fact, if $\Phi_i(x) \neq 0$, $(B_{ii}(x) - 1)\Phi_i(x)$ can be 0 iff $B_{ii}(x) = 1$. But $\Phi_i(x) \neq 0$ means that one of the following situations occurs:

1. $x_i \neq 0$ and $F_i(x) \neq 0$;
2. $x_i = 0$ and $F_i(x) < 0$;
3. $x_i < 0$ and $F_i(x) = 0$.

In every case it is obvious, by the definition of B, that $B_{ii}(x) \neq 1$, so $(B_{ii}(x) - 1)\Phi_i(x) \neq 0$.

Similar reasonings can be repeated for the vector $(A(x) - I)\Phi(x)$. Then it is easy to verify that if $\Phi(x) \neq 0$, then $(B(x) - I)\Phi(x)$ and $(A(x) - I)\Phi(x)$ are both different from 0 and have their nonzero elements in the same positions; such nonzero elements have the same sign. But then for (12) to hold it would be necessary for $\nabla F(x)$ to "revert the sign" of all the nonzero elements of $(B(x) - I)\Phi(x)$, which, by Proposition 2.3, contradicts the fact that $\nabla F(x)$ is a P_0 matrix (because F is a P_0-function). □

The proof of the next theorem uses a technique which was introduced by Geiger and Kanzow [13] in order to prove the same theorem in the case of strongly monotone functions.

THEOREM 4.2. *Suppose that F is a uniform P-function. Then the level sets of Ψ are bounded.*

Proof. The proof is by contradiction. Assume that a sequence $\{x^k\}$ exists such that $\lim_{k\to\infty} \|x^k\| = \infty$ and

$$\Psi(x^k) \leq \Psi(x^0). \tag{13}$$

Define the index set $J = \{i : \{x_i^k\} \text{ is unbounded}\}$. Since $\{x^k\}$ is unbounded, $J \neq \emptyset$. Let $\{z^k\}$ denote a bounded sequence defined in the following way:

$$z_i^k = \begin{cases} 0 & \text{if } i \in J, \\ x_i^k & \text{if } i \notin J. \end{cases}$$

From the definition of $\{z^k\}$ and the assumption on F, we get

$$\begin{aligned} \mu \textstyle\sum_{i\in J} (x_i^k)^2 &= \mu\|x^k - z^k\|^2 \\ &\leq \max_{i\in\{1,.,n\}} \left(x_i^k - z_i^k\right)\left(F_i(x^k) - F_i(z^k)\right) \\ &= \max_{i\in J} x_i^k \left(F_i(x^k) - F_i(z^k)\right) \\ &= x_j^k \left(F_j(x^k) - F_j(z^k)\right) \\ &= |x_j^k| \left|F_j(x^k) - F_j(z^k)\right|, \end{aligned} \tag{14}$$

where μ is the positive constant of the definition of P-function and j is one of the indices for which the max is attained and which we have, without loss of generality, assumed to be independent of k. Since $j \in J$, we can assume, without loss of generality, that

$$\{|x_j^k|\} \to \infty. \tag{15}$$

Dividing by $|x_j^k|$, (14) then gives us $\mu|x_j^k| \leq \left|F_j(x^k) - F_j(z^k)\right|$; this in turn, since $F_j(z^k)$ is bounded, implies

$$\left\{\left|F_j(x^k)\right|\right\} \to \infty. \tag{16}$$

However, (15) and (16) imply

$$\left\{\left|\phi(x_j^k), F_j(x^k)\right|\right\} \to \infty,$$

which contradicts (13). □

In the linear case, the following stronger result easily can be derived from Theorem 2.1 (c) in [40] (see also [11]).

THEOREM 4.3. *Suppose that the F is affine, i.e., that $F(x) = Mx + q$. Then $\lim_{\|x\|\to\infty} \Psi(x) = \infty$ iff M is an R_0-matrix.*

As one of the referees pointed out, by using the results of this section one can easily obtain a new proof of the result of Aganagic and Cottle [1], which shows that a linear complementarity problem with a matrix M which is both a P_0- and an R_0-matrix has a nonempty solution set for all q. In fact, if M is an R_0-matrix, then the level sets of Ψ are bounded by Theorem 4.3, so the function Ψ has at least one global minimizer $\bar{x}$. On the other hand, $\bar{x}$ is a stationary point and hence, if M is also a P_0-matrix, $\bar{x}$ is a solution of the complementarity problem by Theorem 4.1.

5. The algorithm.

The merit function Ψ can be used in several ways to define globally convergent algorithms for the solution of nonlinear complementarity problems. For example, one could simply use an off-the-shelf algorithm to minimize Ψ. In this section we use the merit function in a different, but classical, way. We first define a fast, local algorithm for the solution of problem (NC). Then we globalize this local algorithm by performing an Armijo-type linesearch using the "local" direction, but we revert to the antigradient of Ψ when the "local" direction is not a good descent direction for the merit function. Note that this scheme follows exactly the same lines used in the classical stabilization scheme for Newton's method for the unconstrained minimization of a twice continuously differentiable function. The crucial point will be to show that eventually the gradient direction is never used and the stepsize of one is accepted, so, locally, the global algorithm coincides with the local one, thus ensuring a fast asymptotic convergence rate. We remark that this is neither the only way to exploit the function Ψ nor, possibly, the best one. However, we note that the local algorithm enjoys several interesting properties and that the overall global algorithm, in spite of its simplicity, performs surprisingly well. See [7].

5.1. The local algorithm.

In this section we describe a local algorithm for the solution of nonlinear complementarity problems. The algorithm generates a sequence of points $\{x^k\}$ defined by

$$x^{k+1} = x^k + d^k.$$

To motivate the local algorithm we first consider a simplified situation. Suppose that $\bar{x}$ is a solution of problem (NC), that $\bar{x}$ is nondegenerate, and that we know the sets A and N of variables which are 0 or positive at $\bar{x}$:

$$A := \{i | \bar{x}_i = 0\}, \qquad N := \{i | \bar{x}_i > 0\}.$$

Then, in order to determine $\bar{x}_N$, we would only need to solve the system of equations $F_i(x_N, 0_A) = 0, \quad i \in N$. Provided that $\nabla F_{NN}(\bar{x}_N, 0_A)$ is nonsingular, we could apply Newton's method to this system, thus setting $x_N^{k+1} = x_N^k + d_N^k$, where d_N^k is the solution of the following linear system:

$$(\nabla F_{NN}(x_N^k, 0_A))^T d_N^k = -F_N(x_N^k, 0_A). \tag{17}$$

Obviously, in general we do not know the sets A and N and, furthermore, we would like to avoid the nondegeneracy assumption, which is often not met in practice. We then define d^k in two steps. At each iteration we first estimate the sets A and N, thus fixing some of the components of d^k, then we calculate the remaining part of d^k by solving a reduced linear system. We approximate the sets A and N by the sets A^k and N^k defined by

$$A^k := \{i | x_i^k \leq \varepsilon F_i(x^k)\}, \qquad N^k := \{i | x_i^k > \varepsilon F_i(x^k)\},$$

where ε is a fixed positive constant. By exploiting continuity it is very easy to check that the following result holds.

Proposition 5.1. *Suppose that $\bar{x}$ is a solution of problem* (NC)*. Then for every fixed positive ε, there exists a neighborhood Ω of $\bar{x}$ such that for every x^k belonging to Ω,*

$$\gamma \subseteq A^k \subseteq \gamma \cup \beta,$$
$$\alpha \subseteq N^k \subseteq \alpha \cup \beta.$$

Furthermore, if $\bar{x}$ is nondegenerate, then $\gamma = A^k$ and $\alpha = N^k$.

Based on this result it seems reasonable to define d^k in the following way:

$$d^k_{A^k} = -x^k_{A^k}, \tag{18}$$

while $d^k_{N^k}$ is the solution of the linear system

$$(\nabla F_{N^k N^k}(x^k))^T d^k_{N^k} = -F_{N^k}(x^k) + (\nabla F_{A^k N^k}(x^k))^T x^k_{A^k}. \tag{19}$$

The definition of $d^k_{A^k}$ is very natural, since if we estimate that A^k is the set of variables which are zero at $\bar{x}$, by (18) we obtain

$$x^{k+1}_{A^k} = 0. \tag{20}$$

With regard to (19), we note that if $\bar{x}$ is nondegenerate, $A^k = \gamma$ by Proposition 5.1; since $x^{k+1}_{A^k} = 0$ by (20), (19) reduces to (17). Roughly speaking, the extra term $(\nabla F_{A^k N^k}(x^k))^T x^k_{A^k}$ in (19) is needed to deal with degeneracy. This will be clearer from the proof of the following theorem, where we have collected the main properties of this local algorithm.

Theorem 5.2. *Suppose that $\bar{x}$ is a b-regular solution of problem* (NC)*. Then there exists a neighborhood Ω of $\bar{x}$ such that, if x^o belongs to Ω, the algorithm defined above is such that*

a. *all the linear systems which have to be solved are uniquely solvable*;

b. $\{x^k\} \to \bar{x}$;

c. *the convergence rate of the sequence $\{x^k\}$ to $\bar{x}$ is at least superlinear. If the Jacobian of F is locally Lipschitzian at $\bar{x}$, then the convergence rate is quadratic.*

Proof. Let A be an index set such that

$$\gamma \subseteq A \subseteq \gamma \cup \beta \tag{21}$$

and denote by N its complement, i.e., $N = \{1, \dots, n\} \setminus A$. Consider the function

$$HA(x) = \left[\begin{array}{c} F_N(x) \\ x_A \end{array} \right].$$

By (21) we have that $HA(\bar{x}) = 0$; furthermore, we can write

$$\nabla HA(\bar{x}) = \begin{pmatrix} \nabla F_{NN}(\bar{x}) & O \\ \nabla F_{AN}(\bar{x}) & I_{AA} \end{pmatrix}, \tag{22}$$

which clearly shows, taking into account the b-regularity assumption and (21), that $\nabla HA(\bar{x})$ is nonsingular. Hence we can apply Newton's method to the solution of system $HA(\bar{x}) = 0$ and, thanks to the nonsingularity of the Jacobian at the solution $\bar{x}$, all standard results hold: in particular, there exists a neighborhood ΩA of $\bar{x}$ such that if x^0 belongs to ΩA, the sequence $\{x^k\}$ determined by the Newton methods is well defined and converges to $\bar{x}$. The convergence rate is at least superlinear, quadratic if the Jacobian of F is Lipschitz continuous.

We now note that it is readily seen, using (22), that the vector d^k defined by (18)–(19) can also be equivalently obtained as the Newton's direction for the solution of the system $HA^k(x) = 0$; that is,

$$d^k = -\left[\nabla HA^k(x^k)^T\right]^{-1} HA^k(x^k).$$

By Proposition 5.1 we have $\gamma \subseteq A^k \subseteq \gamma \cup \beta$, so the algorithm defined by (18)–(19) can be seen as a sequence of Newton's steps for a finite number of functions which all have the same solution $\bar{x}$ and whose Jacobians are all nonsingular at $\bar{x}$. The theorem then follows by taking

$$\Omega = \bigcap_{A:\gamma\subseteq A\subseteq\gamma\cup\beta} \Omega A. \qquad \square$$

An interesting feature of the local algorithm is that, under the b-regularity assumption, it is finitely convergent in the case of linear complementarity problems.

THEOREM 5.3. *Suppose that $\bar{x}$ is a b-regular solution of a linear complementarity problem. Then there exists a neighborhood Ω such that if x^0 belongs to Ω, the algorithm above finds the solution $\bar{x}$ in a single step.*

Proof. Take Ω to be any neighborhood of $\bar{x}$ for which

$$\gamma \subseteq A^0 \subseteq \gamma \cup \beta.$$

By reasoning as in the proof of the previous theorem and using the same notation introduced there, we see that d^0 can be seen as the Newton's direction for the solution of the nonsingular *linear* system

$$HA^0(x) = \begin{bmatrix} F_{N^0}(x) \\ x_{A^0} \end{bmatrix} = 0,$$

which has the unique solution $\bar{x}$. The assertion then easily follows by the fact that Newton's method solves nonsingular linear systems in one iteration. $\square$

The properties reported in the two previous theorems are the natural extensions of the classical results for Newton's method for systems of smooth equations. It is worth pointing out the following points.

- No nondegeneracy assumption is needed.
- Only reduced linear systems are solved at each iteration.
- The points generated can violate the constraint $x \geq 0$.

Although it's very simple, we think the local algorithm outlined above enjoys some interesting properties. If we compare it to the classical local linearization method of Josephy [15] and Robinson [36], we see that we have two advantages: the regularity assumption required (b-regularity) is weaker than the R-regularity assumption used in [36]; furthermore, the methods described in [36] require, at each iteration, the solution of a full dimensional linear complementarity problem, which is obviously a computationally more intensive task than solving a linear system. Recently Pang [27] has shown that it is possible to relax the R-regularity assumption in a Josephy–Robinson scheme. However, using this weaker assumption, the linear complementarity problem that has to be solved at each iteration can have multiple solutions and a suitable one has to be selected; this is by no means an easy task. There exist other local methods which solve, at each iteration, only a *(full dimensional)* linear system.

See, e.g., [8, 13, 16, 17, 23, 38]; however, as far as we are aware, all of these methods require nondegeneracy of the solution to get superlinear convergence.

We conclude this section by pointing out a simple by-product of the proof technique used in Theorem 5.2, namely a new and simple proof that b-regularity implies local uniqueness of the solution $\bar{x}$. This result slightly improves on Corollary 4.7 in [22] by relaxing the twice continuous differentiability of F used there and was first obtained in [19].

PROPOSITION 5.4. *Suppose that a solution $\bar{x}$ of an* (NC) *is b-regular. Then $\bar{x}$ is a locally unique solution.*

Proof. The proof is by contradiction. Suppose that we can find a solution $\tilde{x}$ to the complementarity problem as close as we want to $\bar{x}$, and define the following index sets:

$$\tilde{\alpha} := \{i | \tilde{x}_i > 0\}, \qquad \tilde{\beta} := \{i | \tilde{x}_i = 0 = F_i(\tilde{x})\}, \qquad \tilde{\gamma} := \{i | F_i(\tilde{x}) > 0\}.$$

By continuity it is easy to check that, if $\tilde{x}$ is sufficiently close to $\bar{x}$, we have

$$\alpha \subseteq \tilde{\alpha}, \quad \tilde{\beta} \subseteq \beta, \quad \gamma \subseteq \tilde{\gamma},$$

from which we easily get

$$\gamma \subseteq \tilde{\gamma} \subseteq \tilde{\gamma} \cup \tilde{\beta} \subseteq \gamma \cup \beta. \tag{23}$$

However, (23) implies that we can find a set A such that

$$\gamma \subseteq A \subseteq \gamma \cup \beta \quad \text{and} \quad \tilde{\gamma} \subseteq A \subseteq \tilde{\gamma} \cup \tilde{\beta},$$

which in turn, using the notation of the proof of Theorem 5.2, implies that both $\bar{x}$ and $\tilde{x}$ are solutions of the system of equations $HA(x) = 0$. But we have already observed in the proof of Theorem 5.2 that the b-regularity of $\bar{x}$ implies the nonsingularity of the Jacobian $\nabla HA(\bar{x})$ (see (22)). So $\bar{x}$ is a locally unique solution of the system $HA(x) = 0$, and this contradicts the arbitrary closeness of $\tilde{x}$, thus proving the proposition. □

5.2. The global algorithm. In this section we exploit the merit function to globalize, in a simple way, the local algorithm.

GLOBAL ALGORITHM.

```
Data: $x^0 \in \mathbb{R}^n$, $\varepsilon > 0$, $\rho > 0$, $p > 1$, $\beta \in (0, \frac{1}{2})$, $\sigma \in (0, 1)$.
Step 0: Set $k = 0$
Step 1: (stopping criterion) If the stopping criterion is satisfied stop.
Step 2: Calculate the "local direction" $d^k$ according to (18)–(19).
    If system (19) is not solvable set $d^k = -\nabla\Psi(x^k)$.
Step 3: If
    (24)    $\Psi(x^k + d^k) \leq \sigma\Psi(x^k)$,
    set $x^{k+1} = x^k + d^k$
    set $k \leftarrow k + 1$ and go to Step 1.
Step 4: (linesearch) If $d^k$ does not satisfy the following test
    (25)    $\nabla\Psi(x^k)^T d^k \leq -\rho\|d^k\|^p$,
    set $d^k = -\nabla\Psi(x^k)$. Find the smallest $i^k = 0, 1, 2, \ldots$ such that
    (26)    $\Psi(x^k + 2^{-i^k} d^k) \leq \Psi(x^k) + \beta 2^{-i^k} \nabla\Psi(x^k)^T d^k$
```

set $x^{k+1} = x^k + 2^{-i^k} d^k$

set $k \leftarrow k+1$ and go to Step 1.

A few comments are in order. At Step 1 any reasonable stopping criterion can be used. Note that in our case we can use not only classical measures of optimality, like the norm of the vector of the residual, but also measures connected to the merit function like, for example, the norm of the gradient of Ψ. At Step 2 we try to calculate the "local" search direction defined by (18)–(19). If this direction is not well defined we switch to the antigradient of the merit function. Then we exploit the fact that if the nonlinear complementarity problem is solvable, the optimal value of Ψ is 0. So if for some constant $\sigma \in (0,1)$, test (24) is satisfied, we accept the stepsize of one. If this test is passed an infinite number of times this will obviously lead to the function value tending to zero as desired. Should test (24) not be satisfied, we perform in Step 4 a classical linesearch procedure to determine the step size. In this latter case we possibly switch to the antigradient, see test (25), in order to ensure that the search direction is "sufficiently" downhill.

The aim of the acceptability test of Step 3 is twofold. On the one hand it gives us one more chance to accept the stepsize of one; on the other hand it makes it easier to prove the superlinear convergence rate of the algorithm. A test close to (24) has been proposed, with similar purposes, in [31].

To prove the convergence properties of the global algorithm we need three lemmas. The first one is similar to a result contained in Theorem 3.1 of [31]; however, we use a stronger assumption obtaining, correspondingly, a stronger result.

LEMMA 5.5. *Let $H : \mathbb{R}^n \to \mathbb{R}^n$ be a semismooth function and let $\bar{x} \in \mathbb{R}^n$ be such that $H(\bar{x}) = 0$ and such that every matrix in the generalized Jacobian of H at $\bar{x}$ is nonsingular. Suppose that we are given two sequences $\{x^k\}$ and $\{d^k\}$ such that*

(i) $\{x^k\} \to \bar{x}$,

(ii) $\lim \frac{\|x^k + d^k - \bar{x}\|}{\|x^k - \bar{x}\|} = 0$.

Then

$$\lim_{k\to\infty} \frac{\|H(x^k + d^k)\|}{\|H(x^k)\|} = 0. \tag{27}$$

Proof. Since H is semismooth at $\bar{x}$, we can write, by Proposition 1 of [29],

$$H(x^k + d^k) = H(\bar{x}) + W^k(x^k + d^k - \bar{x}) + o(\|x^k + d^k - \bar{x}\|)$$

for all $W^k \in \partial H(x^k + d^k)$, and

$$H(x^k) = H(\bar{x}) + Z^k(x^k - \bar{x}) + o(\|x^k - \bar{x}\|)$$

for all $Z^k \in \partial H(x^k)$. Since $H(\bar{x}) = 0$, these relations imply that

$$\begin{aligned}
\lim_{k\to\infty} \frac{\|H(x^k + d^k)\|}{\|H(x^k)\|} &= \lim_{k\to\infty} \frac{\|W^k(x^k + d^k - \bar{x}) + o(\|x^k + d^k - \bar{x}\|)\|}{\|Z^k(x^k - \bar{x}) + o(\|x^k - \bar{x}\|)\|} \\
&\le \lim_{k\to\infty} \frac{\|W^k(x^k + d^k - \bar{x})\| + \|o(\|x^k + d^k - \bar{x}\|)\|}{|\,\|Z^k(x^k - \bar{x})\| - \|o(\|x^k - \bar{x}\|)\|\,|} \\
&= \lim_{k\to\infty} \frac{\|W^k(x^k + d^k - \bar{x})\| \left(1 + \frac{\|o(\|x^k + d^k - \bar{x}\|)\|}{\|W^k(x^k + d^k - \bar{x})\|}\right)}{\left|\|Z^k(x^k - \bar{x})\| \left(1 - \frac{\|o(\|x^k - \bar{x}\|)\|}{\|Z^k(x^k - \bar{x})\|}\right)\right|}
\end{aligned}$$

$$\leq \lim_{k\to\infty} \frac{2\|W^k(x^k+d^k-\bar{x})\|}{\frac{1}{2}\|Z^k(x^k-\bar{x})\|} = 0,$$

where we have taken into account (ii) and the fact that, by the nonsingularity assumption on the generalized Jacobians of H at $\bar{x}$ and by its boundedness, the sequences of matrices $\{W^k\}$ and $\{Z^k\}$ are such that there exist two positive constants c_m and c_M such that $c_m \leq \|W^k\| \leq c_M$ and $c_m \leq |\mathrm{det} Z^k| \leq c_M$ for all k, so that

$$\|W^k(x^k+d^k-\bar{x})\| = \Theta(\|x^k+d^k-\bar{x}\|), \qquad \text{and} \qquad \|Z^k(x^k-\bar{x})\| = \Theta(\|x^k-\bar{x}\|).$$

The chain of inequalities obviously implies the thesis. □

LEMMA 5.6. *Let $\bar{x}$ be an R-regular solution of the nonlinear complementarity problem, and suppose that we are given two sequences $\{x^k\}$ and $\{d^k\}$ such that*

(i) $\{x^k\} \to \bar{x}$,

(ii) $\lim \frac{\|x^k+d^k-\bar{x}\|}{\|x^k-\bar{x}\|} = 0$.

Then

$$\lim_{k\to\infty} \frac{\Psi(x^k+d^k)}{\Psi(x^k)} = 0.$$

Proof. From Proposition 3.2, every matrix in the generalized Jacobian of Φ at $\bar{x}$ is nonsingular. From Proposition 3.3, Φ is semismooth so that Lemma 5.5 applies to the system $\Phi(x) = 0$. Therefore, the assertion follows by squaring (27). □

LEMMA 5.7. *Suppose that $\{x^k\}$ and $\{d^k\}$ are subsequences of points and corresponding directions generated by the global algorithm. If $\{x^k\} \to \bar{x}$ and $\{d^k\} \to 0$, then $\nabla\Psi(\bar{x}) = 0$.*

Proof. If $d^k = -\nabla\Psi(x^k)$ for an infinite number of indices k, the lemma trivially follows by the continuity of the gradient of Ψ. So, without loss of generality, we examine the case in which d^k is always generated according to (18)–(19). Furthermore, we can also assume, subsequencing if necessary, that $A^k = A$ and $N^k = N$; i.e., the sets A^k and N^k are independent of the iteration. Since we are assuming $\{d^k\} \to 0$, (18) implies

$$\bar{x}_A = 0, \tag{28}$$

which, by (19) and the boundedness of $\nabla F_{NN}(x^k)$, implies that

$$F_N(\bar{x}) = 0. \tag{29}$$

By continuity and by the definition of the sets A and N, (28) and (29) imply

$$\bar{x}_N \geq 0, \qquad F_A(\bar{x}) \geq 0. \tag{30}$$

Equations (28), (29), and (30) imply that $\bar{x}$ is a solution of the nonlinear complementarity problem, so that it is a (global) minimum point of Ψ and hence $\nabla\Psi(\bar{x}) = 0$. □

We can now prove the main result of this section.

THEOREM 5.8. *Let $\{x^k\}$ be the sequence of points generated by the global algorithm. Then*

a. *every accumulation point of $\{x^k\}$ is a stationary point of Ψ*;

b. *if one of the limit points of $\{x^k\}$ is a b-regular solution of problem* (NC) *then the sequence converges to this point*;

c. *if $\{x^k\} \to \bar{x}$ where $\bar{x}$ is an R-regular solution of problem* (NC) *and ∇F is locally Lipschitzian in a neighborhood of $\bar{x}$ then*

1. *eventually d^k is always the "local" direction defined in the previous section (i.e., the antigradient is never used eventually)*;
2. *eventually the stepsize of one is always accepted so that $x^{k+1} = x^k + d^k$*;
3. *the convergence rate is quadratic.*

Proof. (a): the proof is by contradiction. Suppose (renumber if necessary) that $\{x^k\} \to \bar{x}$ and that $\nabla\Psi(\bar{x}) \neq 0$; then we can assume without loss of generality that test (24) is never passed and that

$$0 < \delta \leq \|d^k\| \leq D. \tag{31}$$

If in fact, for a certain subsequence of points test (24) is passed, this would imply that $\{\Psi(x^k)\} \to 0$, recalling that at each step $\Psi(x^{k+1}) < \Psi(x^k)$, so $\bar{x}$ is a global minimum point and hence $\nabla\Psi(\bar{x}) = 0$. If on the other hand, for some subsequence K, $\{\|d^k\|\} \to 0$, we have that $\nabla\Psi(\bar{x}) = 0$ by Lemma 5.7. Taking into account that $\nabla\Psi(x^k)$ is bounded and $p > 1$, $\|d^k\|$ cannot be unbounded because this would contradict (25).

Then, since at each iteration (26) holds and Ψ is bounded from below on the bounded sequence $\{x^k\}$, we have that $\{\Psi(x^{k+1}) - \Psi(x^k)\} \to 0$. This implies, by the linesearch test,

$$\{2^{-i^k}\nabla\Psi(x^k)^T d^k\} \to 0. \tag{32}$$

We want to show that 2^{-i^k} is bounded away from 0. Suppose the contrary. Then, subsequencing if necessary, we have that $\{2^{-i^k}\} \to 0$ so that at each iteration the stepsize is reduced at least once and (26) gives

$$\frac{\Psi(x^k + 2^{-(i^k-1)}d^k) - \Psi(x^k)}{2^{-(i^k-1)}} > \beta\nabla\Psi(x^k)^T d^k. \tag{33}$$

By (31) we can assume, subsequencing if necessary, that $\{d^k\} \to \bar{d} \neq 0$. So, by passing to the limit in (33), we get

$$\nabla\Psi(\bar{x})^T\bar{d} \geq \beta\nabla\Psi(\bar{x})^T\bar{d}. \tag{34}$$

On the other hand, we also see, by (25), that $\nabla\Psi(\bar{x})^T\bar{d} \leq -\rho\|\bar{d}\|^p < 0$, which contradicts (34); hence 2^{-i^k} is bounded away from 0. But then (32) and (25) imply that $\{d^k\} \to 0$ so that $\nabla\Psi(\bar{x}) = 0$ by Lemma 5.7 and point (a) is proved.

(b): since $\bar{x}$ is a b-regular solution then $\bar{x}$ is an isolated global minimum point of Ψ by Proposition 5.4. Denote by Ω the set of limit points of the sequence $\{x^k\}$; we have that $\bar{x}$ belongs to Ω which is therefore a nonempty set. Let δ be the distance of $\bar{x}$ to $\Omega \setminus \bar{x}$ if $\bar{x}$ is not the only limit point of $\{x^k\}$, 1; otherwise, i.e.,

$$\delta = \begin{cases} dist\{\bar{x}|\Omega \setminus \bar{x}\} & \text{if } \Omega \setminus \bar{x} \neq \emptyset, \\ 1 & \text{otherwise;} \end{cases}$$

since $\bar{x}$ is an isolated solution $\delta > 0$. Let us now indicate by Ω_1 and Ω_2 the following sets:

$$\Omega_1 = \{x \in \mathbb{R}^n : dist\{x|\Omega\} \leq \delta/4\}, \qquad \Omega_2 = \{x \in \mathbb{R}^n : \|x\| \geq \|\bar{x}\| + \delta\}.$$

We have that for k sufficiently large, let us say for $k \geq \bar{k}$, x^k belongs at least to one of the two sets Ω_1 and Ω_2. Now let K be the subsequence of all k for which

$\|x^k - \bar{x}\| \leq \delta/4$ (this set is obviously nonempty because $\bar{x}$ is a limit point of the sequence). Since all points of the subsequence $\{x^k\}_K$ are contained in the compact set $S(\bar{x}, \delta/4)$ and every limit point of this subsequence is also a limit point of $\{x^k\}$, we have that all the subsequence $\{x^k\}_K$ converges to $\bar{x}$, the unique limit point of $\{x^k\}$ in $S(\bar{x}, \delta/4)$. Since $\bar{x}$ is b-regular, taking into account that $\nabla\Psi(\bar{x}) = 0$ and the definition of d^k, we have that $\{d^k\}_K \to 0$. We then can find $\tilde{k} \geq \bar{k}$ such that $\|d^k\| \leq \delta/4$ if $k \in K$ and $k \geq \tilde{k}$. Let now $\hat{k}$ be any fixed $k \geq \tilde{k}$ belonging to K; we can write

$$\begin{aligned} dist\{x^{\hat{k}+1}|\Omega \setminus \bar{x}\} &\geq \inf_{y\in\Omega\setminus\bar{x}}\{\|y - \bar{x}\|\} - (\|\bar{x} - x^{\hat{k}}\| + \|x^{\hat{k}} - x^{\hat{k}+1}\|) \\ &\geq \delta - \delta/4 - \delta/4 \\ &= \delta/2. \end{aligned} \tag{35}$$

This implies that $x^{\hat{k}+1}$ cannot belong to $\Omega_1 \setminus S(\bar{x}; \delta/4)$; on the other hand, since $x^{\hat{k}+1} = x^{\hat{k}} + \alpha^{\hat{k}} d^{\hat{k}}$ for some $\alpha^{\hat{k}} \in (0, 1]$, we have

$$\|x^{\hat{k}+1}\| \leq \|x^{\hat{k}}\| + \|\alpha^{\hat{k}} d^{\hat{k}}\| \leq \|\bar{x} + (x^{\hat{k}} - \bar{x})\| + \|d^{\hat{k}}\| \leq \|\bar{x}\| + \|x^{\hat{k}} - \bar{x}\| + \|d^{\hat{k}}\| \leq \|\bar{x}\| + \delta/4 + \delta/4,$$

so that $x^{\hat{k}+1}$ does not belong to Ω_2. Hence we get that $x^{\hat{k}+1}$ belongs to $S(\bar{x}; \delta/4)$. But then, by definition, we have that $\hat{k}+1 \in K$, so by induction (recall that $\hat{k}+1 > \tilde{k}$ also, so that $\|d^{\hat{k}+1}\| \leq \delta/4$) we have that every $k > \tilde{k}$ belongs to K and the whole sequence converges to $\bar{x}$.

(c): since $\bar{x}$ is R-regular, we have that it is also b-regular; so the local direction (18)–(19) is well defined. The three assertions then easily follow by Theorem 5.2, test (24), and Lemma 5.6. □

In general we can only guarantee that every limit point $\bar{x}$, if any, is a stationary point of Ψ. If $\Psi(\bar{x}) = 0$ then $\bar{x}$ is also a solution of the nonlinear complementarity problem. According to what was proved in section 3, we can ensure that every limit point of the sequence generated by the algorithm is a solution of problem (NC) if F is a P_0-function. If F is a uniform P-function, we can also guarantee the existence of a limit point. Actually, in this latter case it is elementary to show that the whole sequence converges to the unique solution of the complementarity problem. It is also possible to give conditions on the function F only at $\bar{x}$ which guarantee that $\bar{x}$ is a solution of the nonlinear complementarity problem; this leads to an analysis similar to the one carried out in [9, 25, 28]. The most obvious of these conditions is that $\nabla F(\bar{x})$ is a P_0-matrix, as can be easily seen from the proof of Theorem 4.1. However, we do not pursue this kind of analysis here and leave it for future research.

In the linear case, Theorem 5.3 and Theorem 5.8 readily give the following result.

THEOREM 5.9. *Let $\{x^k\}$ be the sequence of points generated by the global algorithm when applied to a linear complementarity problem. Suppose that one of the limit points of $\{x^k\}$, say $\bar{x}$, is a b-regular solution. Then the algorithm converges in a finite number of steps to $\bar{x}$.*

In particular, in the case of linear complementarity problems with $F(x) = Mx+q$, we see, by Theorems 4.1 and 4.2, that the algorithm converges to the unique solution of the problem in a finite number of steps if M is a P-matrix.

6. Conclusion. We have studied the properties of a new merit function which allows us to reduce a nonlinear complementarity problem to an unconstrained minimization one under conditions weaker than those previously known. Based on this merit function we have also defined a globally and superlinearly convergent algorithm for the solution of the nonlinear complementarity problem. The new algorithm has a low cost per iteration if compared to algorithms with similar characteristics, and its properties are established under very mild assumptions; the numerical results reported in [7] are very encouraging.

We think that these results along with those reported in [13, 40] indicate that the function Ψ is a very valuable tool in the solution of nonlinear complementarity problems. In particular, we feel that the semismoothness of Φ and the SC^1 property of Ψ have not been fully exploited yet, even if we think that, following recent results reported in [6, 29, 31, 33], they could lead to extremely interesting algorithms; we are currently investigating these topics and hope to report on this research in the near future.

REFERENCES

[1] M. Aganagic and R.W. Cottle, *A note on Q-matrices*, Math. Programming, 16 (1979), pp. 374–377.

[2] G. Auchmuty, *Variational principles for variational inequalities*, Numer. Funct. Anal. Optim., 10 (1989), pp. 863–874.

[3] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley & Sons, New York, 1983.

[4] R. W. Cottle, J.-S. Pang, and R. E. Stone, *The Linear Complementarity Problem*, Academic Press, New York, 1992.

[5] S. P. Dirkse and M. C. Ferris, *The PATH solver: A non-monotone stabilization scheme for mixed complementarity problems*, Optim. Methods Software, 5 (1995), pp. 123–156.

[6] F. Facchinei, *Minimization of SC^1 functions and the Maratos effect*, Oper. Res. Lett., 17 (1995), pp. 131–137.

[7] F. Facchinei and J. Soares, *Testing a new class of algorithms for nonlinear complementarity problems*, in Variational Inequalities and Network Equilibrium Problems, F. Giannessi and A. Maugeri, eds., Plenum Press, New York, 1995, pp. 69–83.

[8] M. C. Ferris and S. Lucidi, *Globally convergent methods for nonlinear equations*, J. Optim. Theory Appl., 81 (1994), pp. 53–71.

[9] M. C. Ferris and D. Ralph, *Projected gradient methods for nonlinear complementarity problems via normal maps*, in Recent Advances in Nonsmooth Optimization, D.Z. Du, L. Qi, and R.S. Womersley, eds., World Scientific Publishers, Singapore, 1995, pp. 57–87.

[10] A. Fischer, *A special Newton-type optimization method*, Optimization, 24 (1992), pp. 269–284.

[11] A. Fischer, *A special Newton-type method for positive semidefinite linear complementarity problems*, J. Optim. Theory Appl., 86 (1995), pp. 585–608.

[12] M. Fukushima, *Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems*, Math. Programming Ser. A, 53 (1992), pp. 99–110.

[13] C. Geiger and C. Kanzow, *On the resolution of monotone complementarity problems*, Comput. Optim. Appl., 5 (1996), pp. 155–173.

[14] P. T. Harker and B. Xiao, *Newton's method for the nonlinear complementarity problem: A B-differentiable equation approach*, Math. Programming Ser. A, 48 (1990), pp. 339–357.

[15] N. H. Josephy, *Newton's Methods for Generalized Equations*, MRC Technical report 1965, Mathematics Research Center, University of Wisconsin, Madison, WI, 1979.

[16] C. Kanzow, *Some equation-based methods for the nonlinear complementarity problem*, Optim. Methods Software, 3 (1994), pp. 327–340.

[17] C. Kanzow, *Nonlinear complementarity as unconstrained optimization*, J. Optim. Theory Appl., 88 (1996), pp. 139–155.

[18] C. Kanzow, *Global Convergence Properties of Some Iterative Methods for Linear Complementarity Problems*, Institute of Applied Mathematics, University of Hamburg, Hamburg, Germany, 1993, revised 1994, Preprint. To appear in SIAM J. Optim.

[19] J. Kyparisis, *Uniqueness and differentiability of solutions of parametric nonlinear complementarity problems*, Math. Programming Ser. A, 36 (1986), pp. 105–113.

[20] Z.-Q. Luo, O.L. Mangasarian, J. Ren, and M.V. Solodov, *New error bounds for the linear complementarity problem*, Math. Oper. Res., 19 (1994), pp. 880–892.

[21] O. L. Mangasarian, *Equivalence of the complementarity problem to a system of nonlinear equations*, SIAM J. Appl. Math., 31 (1976), pp. 89–92.

[22] O. L. Mangasarian, *Locally unique solutions of quadratic programs, linear and nonlinear complementarity problems*, Math. Programming, 19 (1980), pp. 200–212.

[23] O. L. Mangasarian and M. V. Solodov, *Nonlinear complementarity as unconstrained and constrained minimization*, Math. Programming Ser. B, 62 (1993), pp. 277–297.

[24] R. Mifflin, *Semismooth and semiconvex functions in constrained optimization*, SIAM J. Control Optim., 15 (1977), pp. 957–972.

[25] J. J. Moré, *Global Methods for Nonlinear Complementarity Problems*, Argonne National Laboratory, Mathematics and Computer Science Division, Argonne, IL, 1994, preprint MCS-P429-0494.

[26] J. J. Moré and W. C. Rheinboldt, *On P- and S-functions and related classes of n-dimensional nonlinear mappings*, Linear Algebra Appl., 6 (1973), pp. 45–68.

[27] J.-S. Pang, *Convergence of splitting and Newton methods for complementarity problems: An application of some sensitivity results*, Math. Programming Ser. A, 58 (1993), pp. 149–160.

[28] J.-S. Pang and S.A. Gabriel, *NE/SQP: A robust algorithm for the nonlinear complementarity problem*, Math. Programming Ser. A, 60 (1993), pp. 295–337.

[29] J.-S. Pang and L. Qi, *Nonsmooth equations: Motivation and algorithms*, SIAM J. Optim., 3 (1993), pp. 443–465.

[30] J.-S. Pang and L. Qi, *A globally convergent Newton method for convex SC^1 minimization problems*, J. Optim. Theory Appl., 85 (1995), pp. 633–648.

[31] L. Qi, *A convergence analysis of some algorithms for solving nonsmooth equations*, Math. Oper. Res., 18 (1993), pp. 227–244.

[32] L. Qi and H. Jiang, *Karush-Kuhn-Tucker Equations and Convergence Analysis of Newton Methods and Quasi-Newton Methods for Solving These Equations*, Tech. report AMR 94/5, School of Mathematics, University of New South Wales, Australia, 1994.

[33] L. Qi and J. Sun, *A nonsmooth version of Newton's methods*, Math. Programming Ser. A, 58 (1993), pp. 353–368.

[34] D. Ralph, *Global convergence of Damped Newton's method for nonsmooth equations via the path search*, Math. Oper. Res., 19 (1994), pp. 352–389.

[35] S. M. Robinson, *Strongly regular generalized equations*, Math. Oper. Res., 5 (1980), pp. 43–62.

[36] S. M. Robinson, *Generalized equations*, in Mathematical Programming: The State of the Art, A. Bachem, M. Groetschel, and B. Korte, eds., Springer-Verlag, Berlin, New York, 1983, pp. 346–367.

[37] S. M. Robinson, *Normal maps induced by linear transformations*, Math. Oper. Res., 5 (1980), pp. 43–62.

[38] P. K. Subramanian, *Gauss-Newton methods for the complementarity problem*, J. Optim. Theory Appl., 77 (1993), pp. 467–482.

[39] K. Taji, M. Fukushima, and T. Ibaraki, *A globally convergent Newton method for solving strongly monotone variational inequalities*, Math. Programming Ser. A, 58 (1993), pp. 369–383.

[40] P. Tseng, *Growth behavior of a class of merit functions for the nonlinear complementarity problem*, J. Optim. Theory Appl., 89 (1996), pp. 17–37.

[41] B. Xiao and P. T. Harker, *A nonsmooth Newton method for variational inequalities, I: Theory*, Math. Programming Ser. A, 65 (1994), pp. 151–194.

[42] B. Xiao and P. T. Harker, *A nonsmooth Newton method for variational inequalities, II: Numerical results*, Math. Programming Ser. A, 48 (1994), pp. 195–216.

[43] N. Yamashita and M. Fukushima, *On stationary points of the implicit Lagrangian for nonlinear complementarity problems*, J. Optim. Theory Appl., 86 (1995), pp. 653–663.

SIAM J. OPTIM. © 1997 Society for Industrial and Applied Mathematics
Vol. 7, No. 1, pp. 248–273, February 1997 013

THE ORTHOGONALITY THEOREM AND THE STRONG-F-MONOTONICITY CONDITION FOR VARIATIONAL INEQUALITY ALGORITHMS[*]

THOMAS L. MAGNANTI[†] AND GEORGIA PERAKIS[‡]

Abstract. We introduce an approach, called the orthogonality theorem, for establishing the convergence of several algorithms for solving variational inequalities. This theorem, as well as several basic convergence theorems from the literature, impose the condition of strong-f-monotonicity on the problem function. We analyze and introduce some new results concerning this condition and provide a general overview of its properties. For example, we show the relationship between strong-f-monotonicity and convexity.

Key words. variational inequalities, strong-f-monotonicity

AMS subject classifications. 90C30, 90C33, 90C25, 90A14

PII. S1052623493259227

1. Introduction.

We consider the variational inequality problem

$$\text{VI(f, K)}: \quad \text{find} \quad x^* \in K \subseteq R^n : \ f(x^*)^t(x - x^*) \geq 0 \text{ for all } x \in K, \tag{1}$$

defined over a closed, convex (constraint) set K in R^n. In this formulation, $f : K \subseteq R^n \rightarrow R^n$ is a given function and x^* denotes an (optimal) solution of the problem. Variational inequality theory provides a natural framework for unifying the treatment of equilibrium problems encountered in problem areas as diverse as economics, game theory, transportation science, and regional science. Variational inequality problems also encompass a wide range of generic problem areas, including mathematical optimization problems, complementarity problems, and fixed-point problems.

The literature contains many algorithms for solving variational inequality problems. The convergence results for these algorithms involve the entire sequence of iterates (e.g., [7], [29]), some subsequence of iterates (e.g., [19], [25]), or the sequence of averages of the iterates (e.g., [22], [30]). The review article by Harker and Pang [14], the more recent ones by Pang [28] and by Florian and Hearn [10], the Ph.D. thesis of Hammond [12], and the recent book by Nagurney [26] provide insightful surveys of numerous convergence results and citations to many references in the literature.

The variational inequality problem is closely related to the following fixed-point problem:

$$\text{FP(T, K)}: \quad \text{find} \quad x^* \in K \subseteq R^n \text{ satisfying } \ T(x^*) = x^*. \tag{2}$$

In this problem statement $T : K \subseteq R^n \rightarrow K$ is a given map defined over a closed, convex (constraint) set K in R^n. Let G be a given $n \times n$ positive definite and symmetric matrix and I denote the identity map on R^n, i.e., $I(x) = x$. Let Pr_K^G denote the projection operator onto the set K with respect to the norm $\|x\|_G = (x^t G x)^{1/2}$. That is, for any $y \in R^n$, $Pr_K^G(y)$ is the optimal solution to the problem $\min_{x \in K} \|x - y\|_G^2$.

[*] Received by the editors November 30, 1993; accepted for publication (in revised form) October 17, 1995. The preparation of this paper was supported in part by NSF grant 9312971-DDM.
http://www.siam.org/journals/siopt/7-1/25922.html

[†] Sloan School of Management and Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA 02139 (magnanti@eagle.mit.edu).

[‡] Operations Research Center, MIT, Cambridge, MA 02139 (georgiap@mit.edu).

Throughout this paper we will denote the $\|.\|_I$ norm by $\|.\|$. Nevertheless, most of our results easily extend to a more general $\|.\|_G$ norm. The following elementary proposition illustrates a relationship between the variational inequality (1) and the fixed-point problem (2).

PROPOSITION 1.1. *Let ρ be any positive constant, and T be the map $T = Pr_K^G(I-\rho G^{-1}f)$. Then the solutions of the fixed-point problem FP(T,K) are the same as the solutions of the variational inequality problem VI(f,K), if any.*

Several references, [14] and [28] for example, provide more details on the relationship between variational inequality and fixed-point problems.

Banach's fixed-point theorem has been a standard convergence theorem for establishing the convergence of algorithms in many problem settings, including variational inequalities. Two other results, Baillon's theorem [2] (see [22]) and Opial's lemma [27] (see [11], [24], [35]), have also proven to be useful tools for establishing convergence results for variational inequalities. In this paper, we briefly summarize the use of these convergence conditions, and we introduce a new convergence result, the orthogonality theorem. This theorem states that under certain conditions, whenever the map f, evaluated at an accumulation point of the sequence induced by an algorithm, is orthogonal to the line segment between that accumulation point and some variational inequality solution, then every accumulation point of that algorithm is a variational inequality solution. Moreover, if the algorithm map is nonexpansive around a solution for some appropriately defined potential L, then the entire sequence converges to a solution. As part of our discussion, we establish a relationship between the orthogonality theorem and Opial's lemma.

Some recent convergence results (see, for example, [11], [19], [22], [24], [35]) impose the condition of strong-f-monotonicity on the problem function f. These results and those in this paper suggest a natural question: *what are the characteristics of strongly f-monotone functions?* To help answer this question, we examine several properties of strong-f-monotonicity. In particular, we consider the relationship between convexity, or monotonicity in the general asymmetric case and a "weak" form of strong-f-monotonicity. We also examine the relationship between the weak and strong forms of strong-f-monotonicity.

The remainder of this paper is organized as follows. In section 2, we review several convergence theorems, some algorithms that use them, and the conditions required to ensure convergence for these algorithms. We also introduce and prove the orthogonality theorem and compare it with Opial's lemma. In section 3, we examine properties of strong-f-monotonicity.

2. The orthogonality theorem.

Banach's celebrated fixed-point theorem is a classical result that has been extensively used in the literature to establish the convergence of many algorithms, including those for solving variational inequality problems. The basic condition in this theorem is that the algorithm map is a contraction. Other convergence theorems, which we summarize below, have also proven to be important tools to researchers in establishing various convergence results. To state these convergence results, we will impose several conditions on the underlying problem map f and the algorithm map T. The norm $\|x\|_G = (x^tGx)^{1/2}$ induces an operator norm on any operator B. Namely,

$$\|B\|_G = \sup_{\|x\|_G=1} \|Bx\|_G.$$

When $G = I$, we let $\|B\|$ denote $\|B\|_I$.

Definition 1. *A map T is* a contraction map on K *relative to the* $\|.\|_G$ *norm if for some contraction constant* $0 < c < 1$,

$$\|T(x) - T(y)\|_G^2 \leq c\|x - y\|_G^2 \ \textit{for all} \ x, y \in K.$$

The use of this condition usually requires that the problem function f be *strongly monotone on* K relative to the $\|.\|_G$ norm; that is, for some constant $b > 0$,

$$[f(x) - f(y)]^t G[x - y] \geq b\|x - y\|_G^2 \ \text{ for all } x, y \in K.$$

Other convergence results involve nonexpansive estimates.

Definition 2. *A map T is a* nonexpansive map on K *relative to the* $\|.\|_G$ *norm if* $\|T(x) - T(y)\|_G^2 \leq \|x - y\|_G^2$ *for all* $x, y \in K$.

The use of this condition on the algorithm map T usually requires that the problem function f be *strongly-f-monotone on* K relative to the $\|.\|_G$ norm in the sense that for some constant $a > 0$,

$$[f(x) - f(y)]^t G[x - y] \geq a\|f(x) - f(y)\|_G^2 \ \text{ for all } x, y \in K.$$

We refer to the constant a as the monotonicity constant. Furthermore, researchers often require the following condition of ordinary *monotonicity* on the problem function f relative to the $\|.\|_G$ norm:

$$[f(x) - f(y)]^t [x - y] \geq 0 \ \text{ for all } x, y \in K.$$

Contraction, nonexpansiveness, monotonicity, and strong monotonicity are standard conditions in the literature. Gabay [11] implicitly introduced the concept of strong-f-monotonicity and Tseng [35], using the name co-coercivity, explicitly stated this condition. Magnanti and Perakis [19], [20], [22] and Perakis [31] have used the term strong-f-monotonicity for this condition, a choice of terminology that highlights the similarity between this concept and the terminology strong monotonicity, which has become so popular in the literature.

Researchers have established convergence properties for variational inequality algorithms using the following basic theorems.

1. Banach's fixed point theorem (Banach [16]). *Let T be a map, $T : K \to K$, defined on a closed and convex subset K of R^n. If T is a* contraction *map on K relative to the $\|.\|_G$ norm, then for every point $y \in K$, the map $T^k(y)$ converges to a fixed point of the map T.*

2. Baillon's theorem (Baillon [2]). *Let T be a map, $T : K \to K$, defined on a closed, bounded, and convex subset K of R^n. If T is a* nonexpansive *map on K relative to the $\|.\|_G$ norm, then for every point $y \in K$ the map $S_k(y) = \frac{y + T(y) + \cdots + T^{k-1}(y)}{k}$ converges to a fixed point of the map T.*

3. Averaging theorem (Dunn [8]). *Let $T : K \to K$ be a map defined on a closed, convex subset K of R^n and suppose the fixed-point problem* (2) *it defines has a solution. Then if T is a* nonexpansive *map on K relative to the $\|.\|_G$ norm, the sequence*

$$x_k = \frac{a_1 x_1 + a_2 T(x_1) + \cdots + a_k T(x_{k-1})}{a_1 + \cdots + a_k}, \quad x_1 \in K,$$

converges to a fixed point of the map T whenever each $a_k > 0$, $a(k) = \frac{a_k}{a_1 + \cdots + a_k}$, and $\sum_{k=1}^{\infty} a(k)(1 - a(k)) = +\infty$. This fixed point is also the limit of the projection of the points x_k on the set of fixed points of map T.

4. Opial's lemma (Opial [27]). *Let T be a map, $T : K \to K$, defined on a closed and convex subset K of R^n. If T is a* nonexpansive *map on K relative to the $\|.\|_G$ norm and for every point $y \in K$, T is* asymptotically regular, *that is,* $\lim_{k\to+\infty} \|T^{k+1}(y) - T^k(y)\|_G = 0$, *then the map $T^k(y)$ converges to a fixed point of the map T.*

The following elementary observation, together with Proposition 1.1, shows one implication of Opial's lemma for solving variational inequalities.

Proposition 2.1 (see Rockafellar [32]). *Let ρ be a given constant, G some positive definite and symmetric matrix, and $f : K \subseteq R^n \to R^n$ a given function. Then the map $I - \rho G^{-1} f$ is nonexpansive relative to the G norm if and only if f is strongly f-monotone.*

Proof. The map $I - \rho G^{-1} f$ is nonexpansive if and only if for all $x, y \in K$,

$$\|(x-y) - \rho G^{-1}(f(x) - f(y))\|^2_G \leq \|x - y\|^2_G$$

or

$$\|x-y\|^2_G - 2\rho[f(x) - f(y)]^t[x-y] + \rho^2\|f(x) - f(y)\|^2_{G^{-1}} \leq \|x-y\|^2_G$$

or

$$[f(x) - f(y)]^t[x-y] \geq \frac{\rho}{2}\|f(x) - f(y)\|^2_{G^{-1}}.$$

Let g_{max} be the largest eigenvalue of matrix G and g_{min} be the smallest eigenvalue of matrix G. The last expression implies that f is strongly f-monotone with monotonicity constant $\frac{\rho}{2g_{max}}$, and, conversely, whenever f is strongly f-monotone with monotonicity constant $\frac{\rho}{2g_{min}}$, then the map $I - \rho G^{-1} f$ is nonexpansive. □

Opial's lemma for variational inequalities. *Suppose the function $f : K \subseteq R^n \to R^n$ in the variational inequality problem* (1) *is strongly f-monotone, with a monotonicity constant $\frac{\rho}{2}$. Let $T = Pr^G_K(I - \rho G^{-1} f)$. Suppose $y \in K$ and that $\|T^{k+1}(y) - T^k(y)\|_G \longrightarrow_{k\to+\infty} 0$. Then the map $T^k(y)$ converges to a variational inequality solution.*

Proof. Since the projection operator Pr^G_K is nonexpansive and the composition of nonexpansive maps is nonexpansive, by Proposition 2.1 T is nonexpansive. Therefore, Opial's lemma and Proposition 1.1 imply this result. □

Note. The previous lemma provides a general framework for applying Opial's lemma to variational inequalities. Frequently, it is common to view solutions to variational inequality algorithms as fixed points obtained in other ways than through the map $Pr^G_K(I - \rho G^{-1} f)$. Therefore, there are alternate ways to apply Opial's lemma to variational inequality problems. We will use one such approach later in this section.

In order to obtain convergence results, researchers often use auxiliary potential functions. $L(x, x^*) = \|x - x^*\|^2_G$ for some positive definite and symmetric matrix G is an example. This potential (trivially) satisfies the properties $L(x^*, x^*) = \|x^* - x^*\|^2_G = 0$ and $|L(x, x^*)| \geq \|x - x^*\|^2_G$. We will consider other functions that satisfy these properties.

Definition 3. *A potential function $L : K \times K \to R$ is coercive if $L(x^*, x^*) = 0$ and $|L(x, x^*)| \geq d\|x - x^*\|^2_G$ for some positive constant d.*

Researchers have used the potential function $L(x, x^*) = \|x - x^*\|^2_G$ in convergence proofs for projection, linearization, and other algorithms (see, for example, [3], [29]). These proofs require that the map T be nonexpansive, relative to the $\|.\|_G$ norm, around every solution. Another example of a *coercive* potential is $L(x, x^*) = M(x^*) -$

$M(x) - M'(x)^t(x^* - x)$ for some strongly convex function $M : K \subseteq R^n \to R$. This potential appears in Cohen's auxiliary problem framework (see Remark 1(b) at the end of this section and [4], [24] for more details concerning this framework). The condition $|L(x, x^*)| \geq d\|x - x^*\|_G^2$ follows from the strong convexity of M. Another coercive potential often used in convergence proofs is $L(x, x^*) = F(x) - F(x^*)$ for some strongly convex function F. In many cases F is the objective function of a minimization problem corresponding to the variational inequality problem.

In [31] and in subsequent publications, we established the convergence of new [19], [21] and some classical [20], [21] algorithms by implicitly using a common proof technique, which we now state and prove.

The orthogonality theorem (see also [20]). *Let T be a mapping, $T : K \to K$, defined over a closed and convex subset K of R^n. Assume that the map $f : K \to K \subseteq R^n$ defining a variational inequality problem* (1) *is* strongly f-monotone *and that the map T satisfies the following* orthogonality *condition: along a subsequence $\{T^{k_j}(y)\} \subseteq \{T^k(y)\}$ for a given point $y \in K$,*

$$f(T^{k_j}(y))^t(T^{k_j}(y) - x^*) \longrightarrow_{k_j \to \infty} 0$$

for some variational inequality solution x^*. The condition assumes that variational inequality problem VI(f,K) has at least one solution.

I. *Then every accumulation point of the subsequence $T^{k_j}(y)$ is a variational inequality solution.*

II. *Suppose that for every variational inequality solution x^* and some real-valued* coercive *potential function $L(x, x^*)$, the map T is* nonexpansive *relative to L around x^* in the sense that*

$$|L(T^{k+1}(y), x^*)| \leq |L(T^k(y), x^*)|.$$

Then the entire subsequence $\{T^{k_j}(y)\}_{k=0}^{\infty}$ converges to a variational inequality solution.

Proof. I. We show, under the assumptions of this theorem, that for all $x \in K$, $\lim_{k_j \to \infty} f(T^{k_j}(y))^t(x - T^{k_j}(y))$ exists and $\lim_{k_j \to \infty} f(T^{k_j}(y))^t(x - T^{k_j}(y)) \geq 0$. Let x^* be a variational inequality solution for which the *orthogonality condition* holds. The definition of x^* and the strong-f-monotonicity condition imply that for a constant $a > 0$,

$$f(T^{k_j}(y))^t(T^{k_j}(y) - x^*) = [f(T^{k_j}(y)) - f(x^*)]^t(T^{k_j}(y) - x^*) + f(x^*)^t(T^{k_j}(y) - x^*)$$

$$\geq [f(T^{k_j}(y)) - f(x^*)]^t(T^{k_j}(y) - x^*) \geq a\|f(x^*) - f(T^{k_j}(y))\|^2 \geq 0.$$

The orthogonality condition implies that the left-hand side of these inequalities approaches zero as $k_j \to \infty$. Therefore,

$$(3) \qquad \lim_{k_j \to +\infty} \|f(x^*) - f(T^{k_j}(y))\|^2 = 0 \text{ and so } \lim_{k_j \to +\infty} f(T^{k_j}(y)) = f(x^*).$$

This result, together with the orthogonality condition, implies that

$$f(T^{k_j}(y))^t T^{k_j}(y) \longrightarrow_{k_j \to \infty} f(x^*)^t x^*.$$

Let $\bar{x}$ be an accumulation point of the algorithm subsequence $\{T^{k_j}(y)\}$. Then for all $x \in K$,

$$(4) \qquad f(\bar{x})^t(x - \bar{x}) = \lim_{k_j \to \infty} f(T^{k_j}(y))^t(x - T^{k_j}(y)) = f(x^*)^t(x - x^*) \geq 0$$

since x^* is a variational inequality solution. Therefore, $\bar{x}$ is a variational inequality solution.

II. If some potential $L(x, x^*)$ satisfies the condition $|L(x, x^*)| \geq d\|x - x^*\|_G^2$, $d > 0$, with $L(x^*, x^*) = 0$, and the map T is nonexpansive relative to L around every variational inequality solution x^*, then $|L(T^{k_j+1}(y), x^*)| \leq |L(T^{k_j}(y), x^*)|$, and so the sequence $\{|L(T^{k_j}(y), x^*)|\}$ is nonincreasing and, therefore, convergent for every solution x^*. Moreover, since $|L(T^{k_j}(y), x^*)| \geq d\|T^{k_j}(y) - x^*\|_G^2$, the entire sequence $\{T^{k_j}(y)\}$ is bounded and therefore it has at least one accumulation point. We have just shown that every accumulation point $\bar{x}$ of the subsequence $\{T^{k_j}(y)\}$ is a variational inequality solution. Therefore, $|L(\bar{x}, \bar{x})| = 0$. If we set $x^* = \bar{x}$, then the nonexpansiveness of $|L(T^{k_j}(y), x^*)|$ implies that

$$0 \leq d\|T^{k_j}(y) - \bar{x}\|_G^2 \leq |L(T^{k_j}(y), \bar{x})| \longrightarrow_{k_j \to \infty} 0,$$

which implies that the entire subsequence $T^{k_j}(y)$ converges to the solution $\bar{x}$. □

In contrast to Opial's lemma, the orthogonality theorem applies to situations when the algorithm sequence has multiple accumulation points. Later, in Figure 1, we present an example with two accumulation points, which are both variational inequality solutions.

One significant limitation of the orthogonality theorem, as stated, is that the orthogonality condition requires a variational inequality solution x^*. Can we use the theorem without having a solution in hand? Several references that we cited earlier [3], [4], [19], [24], [29] have shown how to use coercive potential functions to establish algorithmic convergence by *implicitly* considering solutions x^*.

The orthogonality theorem suggests a possible new potential function, the potential $P(x, z) = f(x)^t(x - z)$. In particular, if $x = T^k(y)$ and $z = x^*$, then $P(T^k(y), x^*) = f(T^k(y))^t(T^k(y) - x^*)$. This is the potential used in the orthogonality condition. This potential relates to the gap function

$$h(x) = f(x)^t x - \min_{w \in K} f(x)^t w = P(x, \text{argmin}_{w \in K} f(x)^t w) \quad \text{for all } x \in K.$$

Hearn [15] introduced this gap function in the context of nonlinear programming problems (see section 4 in [28]). A point x^* is a solution of the variational inequality problem if and only if x^* is a global optimal solution of the problem $\min_{x \in K} h(x)$ and $h(x^*) = 0$. Observe that in general $h(x) \geq P(x, x^*)$. In [19], [21], and [31] we have used this potential and the orthogonality theorem (implicitly or explicitly) in convergence proofs, again by considering the solution x^* only implicitly. In particular, in [21] we have used this potential to establish the convergence of a descent framework for solving variational inequalities, which includes as special cases the steepest descent method, the Frank–Wolfe method (symmetric and asymmetric) linearization schemes [29], and a generalized contracting ellipsoid method [13].

The following result shows another way to avoid explicit knowledge of a variational inequality solution.

Proposition 2.2. *Let VI(f,K) be a variational inequality problem with a monotone problem function f. Consider any continuous mapping $T : K \to K$ satisfying the property that every fixed point of this mapping is a variational inequality solution (e.g., see Proposition* 2.1*). Then for every $y \in K$, the asymptotic regularity condition of T, i.e.,* $\lim_{k \to +\infty} \|T^{k+1}(y) - T^k(y)\|_G = 0$, *implies the orthogonality condition along any convergent subsequence $\{T^{k_j}(y)\} \subseteq \{T^k(y)\}$, i.e.,*

$$\lim_{k_j \to +\infty} f(T^{k_j}(y))^t(T^{k_j}(y) - x^*) = 0.$$

Proof. When $\lim_{k\to+\infty}\|T^{k+1}(y)-T^k(y)\|_G=0$ for some $y\in K$, the continuity of the map T implies that every accumulation point $\bar{x}$ of the sequence $\{T^k(y)\}$ satisfies the condition $\|T(\bar{x})-\bar{x}\|_G=0$ and is therefore a fixed point of the map T. By assumption, $\bar{x}$ is also a variational inequality solution. But since $\bar{x}$ solves the variational inequality problem, the monotonicity of f implies that for any variational inequality solution x^*, $0\geq f(\bar{x})^t(\bar{x}-x^*)\geq 0$, and, therefore, for any convergent subsequence $\{T^{k_j}(y)\}$, $\lim_{k_j\to+\infty} f(T^{k_j}(y))^t(T^{k_j}(y)-x^*)=0$. □

This proposition shows that for variational inequalities, Opial's asymptotic regularity condition, which does not require knowledge of a variational inequality solution, implies the orthogonality condition. Therefore, in the context of variational inequalities, the orthogonality theorem conditions are more general than those imposed in Opial's lemma.

Is the orthogonality condition more general or is this condition and asymptotic regularity equivalent (when applied to variational inequalities)? Under what circumstances are these conditions equivalent? The following example shows that, in general, the orthogonality condition is more general.

Example 1. Consider the symmetric Frank–Wolfe algorithm with problem iterates x^k: at each step $k=1,2,\ldots$, the algorithm solves the linear program $y^k=\text{argmin}_{x\in K} f(x^{k-1})^t x$ and then solves the following one-dimensional variational inequality problem, i.e., find $x^k\in[y^k;x^{k-1}]$ satisfying $f(x^k)^t(x-x^k)\geq 0$ for all $x\in[y^k;x^{k-1}]$.

For more details on this algorithm see, for example, [12] and [25].

As shown by the following data, the algorithm's iterates need not always satisfy the asymptotic regularity condition.

Let $K=\{x=(x_1,x_2)\in R^2:\ 0\leq x_1\leq 1,\ 0\leq x_2\leq 1\}$ be the feasible set and let

$$f(x)=\left\{\begin{array}{ll}(x_1-\frac{1}{4},1) & \text{if } 0\leq x_1\leq\frac{1}{4}\\ (0,1) & \text{if } \frac{1}{4}\leq x_1\leq\frac{3}{4}\\ (x_1-\frac{3}{4},1) & \text{if } \frac{3}{4}\leq x_1\leq 1\end{array}\right\}$$

be the problem function f.

It is easy to see that f is a Lipschitz continuous and a strongly f-monotone function (but not strictly or strongly monotone) with a symmetric Jacobian matrix. For problems that satisfy these conditions, the orthogonality theorem (see [20]) and several other proof techniques show that every accumulation point is a variational inequality solution. The solutions of this problem are all the points $x^*=(x_1^*,0)$ with $\frac{1}{4}\leq x_1^*\leq\frac{3}{4}$. If we initiate the algorithm at the point $x^0=(0,\frac{1}{8})$, then **step k=1** solves at $y^1=(1,0)$ and $x^1=(\frac{7}{8},\frac{1}{64})$. **Step k=2** solves at $y^2=(0,0)$ and $x^2=\frac{13}{49}(\frac{7}{8},\frac{1}{64})$. Starting at the point $x^0=(0,\frac{1}{8})$ (or any point $(0,z)$ with $z\leq\frac{1}{8}$), the algorithm induces two subsequences, $\{x^{2l-1}\}_{l=0}^{\infty}$ and $\{x^{2l}\}_{l=0}^{\infty}$ with two accumulation points $x^*=(\frac{3}{4},0)$ and $x^{**}=(\frac{1}{4},0)$ that are both variational inequality solutions. The asymptotic regularity condition and, therefore, Opial's lemma does not hold since $\lim_{k\to+\infty}\|x^k-x^{k+1}\|=\frac{1}{2}$. As is easy to check, the orthogonality condition holds for both subsequences, so in this example the orthogonality theorem, but not Opial's lemma, applies. Figure 1 illustrates this example.

This example shows that in one algorithmic setting (the Frank–Wolfe algorithm) for variational inequalities, the orthogonality condition is more general than asymptotic regularity. We now show that for certain classes of variational inequalities, for a rather large class of algorithms, the two conditions are equivalent.

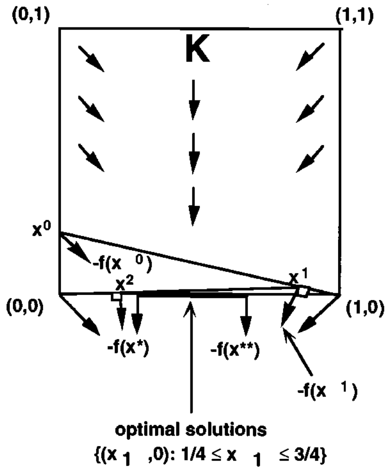


FIG. 1. *The orthogonality theorem.*

General iterative scheme. This class of algorithms (see [7]) determines the point x^{k+1} from the previous iterate x^k by solving the variational inequality,

$$\text{find } x^{k+1} \in K \text{ satisfying } g(x^{k+1}, x^k)^t(z - x^{k+1}) \geq 0 \quad \text{for all } z \in K. \tag{5}$$

We assume that the underlying map $g(x, y)$ satisfies two assumptions.

A1. $g(x, y)$ is strongly f-monotone with respect x.

A2. $g(x, x) = \rho f(x)$ for some constant $\rho > 0$.

Examples. ($\rho > 0$ is a given constant)

(a) $g(x, y) = \rho f(y) + G(x - y)$ for a positive semidefinite, symmetric matrix G.

(b) $g(x, y) = \rho f(y) + [G(x) - G(y)]$, with $G(x) = \nabla K(x)$ for some convex function K.

Note that $x^{k+1} = x^k$ solves the variational inequality (5) if and only if x^k solves the variational inequality (1). Therefore, in the context of this algorithm, fixed points are the same as variational inequality solutions.

PROPOSITION 2.3. *Let VI(f,K) be a variational inequality problem. Consider the general iterative scheme* (5). *Let $T : K \to K$ be a function that maps a point $y = x^k$ into a point $T(y) = x^{k+1}$ that solves* (5). *If the problem function f is monotone, K is a bounded set and some constant $C > 0$ satisfies the condition $\|\nabla_y g(x, y)\| \leq C$ for all $x, y \in K$, then the asymptotic regularity condition on the map T implies the orthogonality condition across the* entire *sequence. Conversely, if*

1. *the problem function f is strongly f-monotone with constant a,*

2. *the scheme's function $g(x, y)$ is strongly monotone relative to its x component, i.e., for some constant $b > 0$, $[g(x_1, y) - g(x_2, y)]^t[x_1 - x_2] \geq b\|x_1 - x_2\|^2$ for all $x_1, x_2 \in K$,*

3. $0 < \rho < 4ab$ *(often $\rho = 1$, then this condition requires $1 < 4ab$),*

then the orthogonality condition along some subsequence implies the asymptotic regularity along that subsequence.

3'. *Replacing* 3 *with the assumption that the orthogonality condition holds along the* entire *sequence $\{T^k(y)\}$ (with no restrictions on ρ) also implies the asymptotic regularity condition.*

Proof. " $\Rightarrow$ " Set $T^k(y) = x^k$ and $T^{k+1}(y) = x^{k+1}$. If T is asymptotically regular, then $\lim_{k\to+\infty} \|x^k - x^{k+1}\| = 0$. The fact that $g(x,x) = \rho f(x)$, f is monotone, and $x^* \in K$ is a variational inequality solution implies that

$$0 \leq g(x^{k+1},x^k)^t(x^* - x^{k+1}) \leq [g(x^{k+1},x^k) - g(x^{k+1},x^{k+1})]^t[x^* - x^{k+1}]$$

(an application of the mean value and Cauchy's inequality imply that)

$$\leq \lim_{k\to+\infty} \|x^k - x^{k+1}\| \cdot \|\nabla_y g(x^{k+1},y)\| \cdot \|x^* - x^{k+1}\| \leq C \lim_{k\to+\infty} \|x^k - x^{k+1}\| \cdot \|x^* - x^{k+1}\| = 0.$$

Therefore, $\lim_{k\to+\infty} g(x^{k+1},x^k)^t(x^* - x^{k+1}) = 0$ and

$$\lim_{k\to+\infty} [g(x^{k+1},x^k) - g(x^{k+1},x^{k+1})]^t[x^* - x^{k+1}] = 0.$$

So $\rho \lim_{k\to+\infty} f(x^{k+1})^t(x^* - x^{k+1}) = \lim_{k\to+\infty}[g(x^{k+1},x^{k+1})]^t[x^* - x^{k+1}]$

$$= \lim_{k\to+\infty}[g(x^{k+1},x^k)]^t[x^* - x^{k+1}] - [g(x^{k+1},x^k) - g(x^{k+1},x^{k+1})]^t[x^* - x^{k+1}] = 0.$$

Consequently, the orthogonality condition holds, i.e., $\lim_{k\to+\infty} f(x^k)^t(x^* - x^k) = 0$.

" $\Leftarrow$ " Conversely, (a) assume that the orthogonality condition holds along some subsequence, i.e., $\lim_{k_j\to+\infty} f(x^{k_j})^t(x^* - x^{k_j}) = 0$. Then the general iterative scheme and the fact that $x^{k_j} \in K$ imply that

$$0 \leq g(x^{k_j+1},x^{k_j})^t(x^{k_j} - x^{k_j+1}) = [g(x^{k_j+1},x^{k_j}) - g(x^{k_j},x^{k_j})]^t(x^{k_j} - x^{k_j+1})$$

$$+ g(x^{k_j},x^{k_j})^t(x^{k_j} - x^{k_j+1})$$

(the strong monotonicity of $g(x,y)$ relative to x, and the fact that $g(x,x) = \rho f(x)$ implies that)

$$\leq -b\|x^{k_j+1} - x^{k_j}\|^2 + \rho f(x^{k_j})^t(x^{k_j} - x^{k_j+1}) = -b\|x^{k_j+1} - x^{k_j}\|^2$$

$$+ \rho f(x^{k_j})^t(x^{k_j} - x^*) + \rho f(x^{k_j})^t(x^* - x^{k_j+1})$$

(from the definition of a VI(f,K) solution x^*)

$$\leq -b\|x^{k_j+1} - x^{k_j}\|^2 + \rho f(x^{k_j})^t(x^{k_j} - x^*)$$

$$+ \rho[f(x^{k_j}) - f(x^*)]^t(x^* - x^{k_j+1}) = -b\|x^{k_j+1} - x^{k_j}\|^2 + \rho f(x^{k_j})^t(x^{k_j} - x^*)$$

$$+ \rho[f(x^{k_j}) - f(x^*)]^t(x^* - x^{k_j}) + \rho[f(x^{k_j}) - f(x^*)]^t(x^{k_j} - x^{k_j+1})$$

(strong-f-monotonicity implies that)

$$\leq -b\|x^{k_j+1} - x^{k_j}\|^2 + \rho f(x^{k_j})^t(x^{k_j} - x^*) - a\rho\|f(x^{k_j}) - f(x^*)\|^2$$
$$+ \rho[f(x^{k_j}) - f(x^*)]^t(x^{k_j} - x^{k_j+1})$$

(by expanding $\|\sqrt{a}(f(x^{k_j}) - f(x^*)) - \frac{1}{2\sqrt{a}}(x^{k_j} - x^{k_j+1})\|^2$ and rearranging terms, we obtain)

$$\leq \left[-b + \frac{\rho}{4a}\right] \|x^{k_j+1} - x^{k_j}\|^2 + \rho f(x^{k_j})^t(x^{k_j} - x^*).$$

If $0 < \rho < 4ab$, then $0 < [b - \frac{\rho}{4a}]\|x^{k_j+1} - x^{k_j}\|^2 \leq \rho f(x^{k_j})^t(x^{k_j} - x^*)$. Therefore, the orthogonality condition along the subsequence x^{k_j}, i.e.,

$$\lim_{k_j \to +\infty} f(x^{k_j})^t(x^{k_j} - x^*) = 0,$$

implies the asymptotic regularity along that subsequence, i.e.,

$$\lim_{k_j \to +\infty} \|T^{k_j+1}(x^0) - T^{k_j}(x^0)\| = 0.$$

(b) Assume that there are no restrictions on ρ and the orthogonality condition holds along the entire sequence, i.e., $\lim_{k\to+\infty} f(x^k)^t(x^* - x^k) = 0$. The argument that led us to (3) in part (I) of the orthogonality theorem implies that $\lim_{k\to+\infty} f(x^k) = f(x^*)$. Furthermore, as in part (a), we find that

$$\begin{aligned} 0 \leq -b\|x^{k+1} - x^k\|^2 + \rho f(x^k)^t(x^k - x^*) + \rho[f(x^k) - f(x^{k+1})]^t(x^* - x^{k+1}) \\ + \rho f(x^{k+1})^t(x^* - x^{k+1}). \end{aligned}$$

Therefore,

$$b\|x^{k+1} - x^k\|^2 \leq \rho f(x^k)^t(x^k - x^*) - \rho f(x^{k+1})^t(x^{k+1} - x^*) + \rho[f(x^k) - f(x^{k+1})]^t(x^* - x^{k+1}).$$

Cauchy's inequality and the fact that K is a bounded set implies that

$$\begin{aligned} \lim_{k\to+\infty} b\|x^{k+1} - x^k\|^2 \leq \rho f(x^k)^t(x^k - x^*) - \rho f(x^{k+1})^t(x^{k+1} - x^*) \\ + \rho\|f(x^k) - f(x^{k+1})\|.\|x^* - x^{k+1}\| = 0. \end{aligned}$$

Therefore, asymptotic regularity holds: $\lim_{k\to+\infty} \|x^{k+1} - x^k\| = 0$. Notice that part (b) holds regardless of the choice of ρ. □

Remarks.

1. If we impose some conditions on the underlying data, several classical methods for solving the variational inequality problem which are special cases of the general iterative scheme satisfy the strong monotonicity condition on the scheme's function g.

(a) *Linear approximation methods* (see [29]), with $g(x, y) = f(y) + A(y)^t(x - y)$. In this case, the matrix $A(y)$ should be uniformly positive definite for all y and the problem function f strongly f-monotone. The following examples are special cases of this class of algorithms.

The linearized Jacobi method, with $A(y) = \text{diag}(\nabla f(y))$, which should have positive elements that are bounded away from zero as y varies.

The projection method, with $A(y) = G$, a positive definite matrix.

Newton's method, with $A(y) = \nabla f(y)$, a uniformly positive definite matrix.

The quasi-Newton method, with $A(y) = approx(\nabla f(y))$, which we require to be a uniformly positive definite matrix. The notation $approxM$ represents an approximation of the matrix M.

The linearized Gauss–Seidel method, with $A(y) = L(y) + D(y)$ or $A(y) = U(y) + D(y)$. ($L(y)$ and $U(y)$ are the lower and upper diagonal parts of the matrix $\nabla f(y)$.) $A(y)$ should be a uniformly positive definite matrix.

(b) *Cohen's auxiliary problem framework* (see [4], [24]), with $g(x, y) = \rho f(y) + G(x) - G(y)$. In this case, the problem function f should be strongly f-monotone

Table 1
Convergence approaches.

What Theorems	Which Algorithms	What Conditions
Banach	Projection [29], [7], [5], [3]	strong monot., $0 < \rho < 2a/L^2 g$
Banach	Relaxation [1], [6]	$\sup_{x,y \in K} \ \|g_y(x,y)\| \leq \lambda\alpha$, $0 < \lambda < 1$
Banach	Original Steepest Descent [13]	$Df(x)$ p.d., $Df(x)^2$ p.d.
Banach	Cohen's Aux. Probl. Fram. [4]	strong monot., $0 < \rho < 2ab/L^2$
Banach	Forw. and backw. step alg. [11]	strong monot., $0 < \rho < 2a/g$
Baillon	Averages of Steepest Descent [13]	$Df(x)$ p.d., $Df(x)^2$ p.s.d.
Baillon	Averages of Short Step Steepest Descent [21]	strong-f-monot., $0 < \rho \leq 2a$
Baillon	Averages of Constr. Short Step Steepest Descent [21]	strong-f-monot., $0 < \rho \leq 2a$
Baillon	Averages of Projection [22]	strong-f-monot., $0 < \rho \leq 2a/g$
Baillon	Averages of Relaxation [22]	$\sup_{x,y \in K} \ \|g_y(x,y)\| \leq \alpha$
Opial	Forw.-backw. oper. splitting alg. [11]	strong-f-monot., $0 < \rho < 2a/g$
Opial	Projection [11]	strong-f-monot., $0 < \rho < 2a/g$
Opial	Cohen's Aux. Probl. Framew. [24]	strong-f-monot., $0 < \rho < 2ab$
Opial	Short Step Steepest Descent [21]	strong-f-monot., $0 < \rho < 2a$
Opial	Constr. Short Step Steepest Descent [21]	strong-f-monot., $0 < \rho < 2a/g$
Opial	Asymmetric Projection [35]	G asym., p.d., $G'^{-1/2}[f(G'^{-1/2}y) - (G - G')G'^{-1/2}y]$ strong-f-mon., cnst. $\geq 1/2$
Opial	Modified Aux. Probl. Framew. [24]	$f - M$ strong mon., K' strong mon., some ρ
Orthogonality	Short Step Steepest Descent [21]	strong-f-monot., $0 < \rho < 2a$
Orthogonality	Constr. Short Step Steepest Descent [21]	strong-f-monot., $0 < \rho < 2a/g$
Orthogonality	Projection [22]	strong-f-monot., $0 < \rho < 2a/g$
Orthogonality	Accum. pts. of Sym. Frank-Wolfe [22]	strong-f-monot., symmetry, $K \neq$, comp., conv.
Orthogonality	Accum. pts. of Affine Asym. Frank-Wolfe [21]	affine, strong monot., near-square-symmetry
Orthogonality	Accum. pts. of Affine descent Fram. [21]	affine, strong monot., near-square-symmetry
Orthogonality	Accum. pts. of Geometric Framework [19]	strong-f-monot., $K \neq$, convex, compact
Orthogonality	Cohen's Aux. Probl. Framew. [4]	strong-f-monot., $0 < \rho < 2ab$
Orthogonality	Asymmetric Projection [35]	G asym., p.d., $G'^{-1/2}[f(G'^{-1/2}y) - (G - G')G'^{-1/2}y]$ strong-f-mon, cnst. $\geq 1/2$
Orthogonality	Modified Aux. Probl. Framew. [24]	$f - M$ strong mon., K' strong mon., some ρ

and the function G should be strongly monotone. (In fact, Cohen assumes that $G(y) = M'(y)$, that is a gradient matrix for a strongly convex function M.)

2. Table 1 illustrates the use of various convergence conditions for solving variational inequalities. As indicated in this table, the use of the orthogonality theorem establishes the convergence of several algorithms whose convergence has not been established using Opial's lemma. These algorithms include the general geometric framework [19], the Frank–Wolfe algorithm [20], and a descent framework [21].

Notes. In Table 1, G' denotes the symmetric part of the matrix G involved in the projection method, i.e., $G' = \frac{1}{2}(G + G^t)$. The constants involved are $g = \min(\textit{eigenvalue of } G)$, a is the strong-f-monotonicity constant, L is the Lipschitz continuity constant, $\alpha = \inf_{x,y \in K} \ (\min(\textit{eigenvalue } g_x(x,y))) > 0$, and b is the constant involved in the strong convexity of the function K. Finally, the map M is the monotone part of the function G_ϵ that is involved in the modified auxiliary problem framework (see [24] for more details).

Proposition 2.4. *Let VI(f,K) be a variational inequality problem. Consider the general iterative scheme* (5) *for maps $g(x,y)$ that satisfy the assumptions* A1 *and* A2*. Let $T : K \rightarrow K$ be a function that maps a point $y = x^k$ into a point $T(y) = x^{k+1}$ that solves* (5)*. Suppose K is a convex, compact set and f satisfies the following* incremental orthogonality condition*:*

$$\lim_{k \rightarrow +\infty} f(x^k)^t (x^k - x^{k+1}) = 0.$$

Then every accumulation point of the sequence $\{x_k\}$ is a solution.

Proof. (The proof is essentially the same as the proof of Proposition 2.3, with $g(x^{k+1}, x^k)$ playing the role of $f(x^k)$.) Suppose $\lim x^{k_j} = \bar{x}$. Since $g(x^k, x^k) = \rho f(x^k)$, the definition of x^{k+1} and the strong-f-monotonicity of $g(x,y)$ with respect to x implies that

$$\rho \lim_{k \rightarrow +\infty} f(x^k)^t [x^{k+1} - x^k]$$

$$= \lim_{k\to+\infty}[\rho f(x^k) - g(x^{k+1}, x^k)]^t[x^k - x^{k+1}] + \lim_{k\to+\infty} g(x^{k+1}, x^k)^t(x^k - x^{k+1})$$

$$\geq \lim_{k\to+\infty}[g(x^k, x^k) - g(x^{k+1}, x^k)]^t[x^k - x^{k+1}] \geq B \lim_{k\to+\infty} \|g(x^k, x^k) - g(x^{k+1}, x^k)\|^2.$$

Therefore, $\lim_{k\to+\infty} \|g(x^k, x^k) - g(x^{k+1}, x^k)\|^2 = 0$. This result together with the conditions $\lim_{k\to+\infty} f(x^k)^t(x^k - x^{k+1}) = 0$ and $g(x^k, x^k) = \rho f(x^k)$ implies that

$$\lim_{k\to+\infty} \rho f(x^k)^t x^k = \lim_{k\to+\infty} g(x^{k+1}, x^k)^t x^{k+1}.$$

But then for all $x \in K$, the definition of x^{k+1} implies that

$$\rho f(\bar{x})^t(x - \bar{x}) = \lim \rho f(x^k)^t(x - x^k) = \lim_{k\to+\infty} g(x^{k+1}, x^k)^t(x - x^{k+1}) \geq 0.$$

That is, $\bar{x}$ is a variational inequality solution. □

Proposition 2.5. (a) *The asymptotic regularity condition on the map T implies the* incremental orthogonality condition. (b) *If* (i) *the problem function f is strongly f-monotone with a monotonicity constant a, and if* (ii) *the orthogonality condition holds, then the* incremental orthogonality condition *holds.*

Proof. (a) It is easy to see this result. (b) If x^* is a variational inequality solution, then the strong-f-monotonicity condition implies that

$$f(x^k)^t(x^k - x^*) \geq (f(x^k) - f(x^*))^t(x^k - x^*) \geq a\|f(x^k) - f(x^*)\|^2.$$

Therefore, the orthogonality condition

$$\lim_{k\to+\infty} f(x^k)^t(x^k - x^*) = 0$$

implies that $\lim_{k\to+\infty} \|f(x^k) - f(x^*)\|^2 = 0$, which in turn implies that $\lim_{k\to+\infty} \|f(x^k) - f(x^{k+1})\|^2 = 0$. Moreover, since

$$\lim_{k\to+\infty} f(x^k)^t(x^k - x^{k+1})$$

$$= \lim_{k\to+\infty} f(x^k)^t(x^k - x^*) - f(x^{k+1})^t(x^{k+1} - x^*) + [f(x^{k+1}) - f(x^k)]^t[x^{k+1} - x^*]$$

$$\leq \lim_{k\to+\infty} f(x^k)^t(x^k - x^*) - \lim_{k\to+\infty} f(x^{k+1})^t(x^{k+1} - x^*) + \lim_{k\to+\infty} \|f(x^{k+1}) - f(x^k)\|.\|x^{k+1} - x^*\|$$

$$= 0. \qquad □$$

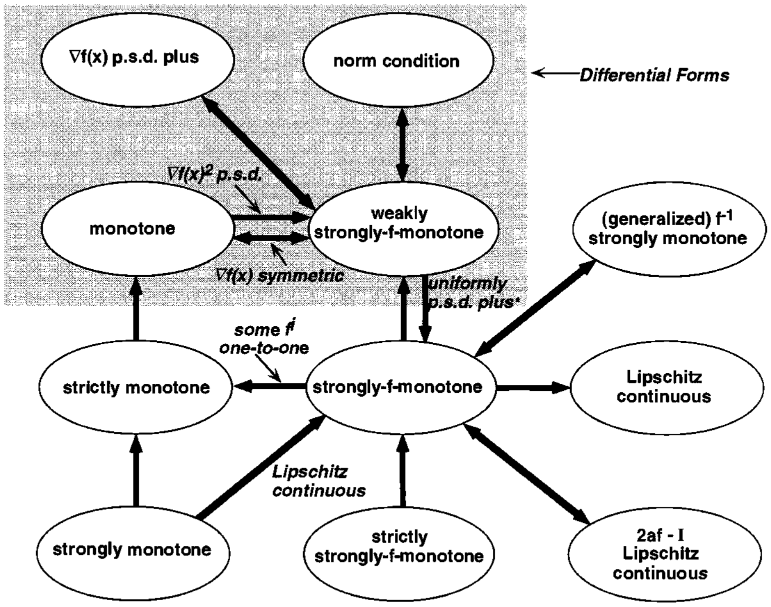
Remarks.

1. The condition

$$\lim_{k\to+\infty} f(x^k)^t(x^k - x^{k+1}) = 0$$

also applies to schemes that include line searches. Let $x^{k+1} = x^k + a(k)(y^{k+1} - x^k)$, with $a(k)$ found through a line search procedure, satisfying some assumptions (see [23] for more details), and y^{k+1} found through the solution of a scheme (5). Then, if

$$\lim_{k\to+\infty} f(x^k)^t(x^k - y^{k+1}) = 0,$$



* Implies differential form of strongly-f-monotone

Relationships between different types of monotonicity

FIG. 2.

a proof similar to the previous proposition again shows that every accumulation point of the sequence x^k is a VIP solution.

2. As we have noted, *the incremental orthogonality condition* is more general than the asymptotic regularity condition and the orthogonality condition. Moreover, as shown in Proposition 2.2, the orthogonality condition for a general scheme with an algorithm function $g(x, y)$ that is strongly monotone with respect to x is equivalent to the asymptotic regularity condition. Therefore, by Opial's lemma for any algorithm satisfying these conditions, the entire sequence of iterates must converge to a variational inequality solution. In contrast, the incremental orthogonality condition permits the algorithm to have multiple accumulation points.

3. For fixed-point problems FP(T,K), $f(x^k) = x^k - T(x^k)$ and so $y^{k+1} = T(x^k)$ and $g(y^{k+1}, x^k) = y^{k+1} - T(x^k)$. Therefore,

$$\lim_{k \to +\infty} f(x^k)^t (x^k - y^{k+1}) = \lim_{k \to +\infty} \|x^k - T(x^k)\|^2 = 0$$

implies that every accumulation point of x^k is a solution.

3. On the strong-f-monotonicity condition. As shown in the previous section, and particularly as summarized in Table 1, the strong-f-monotonicity condition plays an important role as an underlying condition for establishing the convergence of several variational inequality algorithms. Moreover, as we have seen, it is a key assumption in the orthogonality theorem. In this section we study this condition. In particular, we examine the relationship between convexity and strong-f-monotonicity. Is strong-f-monotonicity the "natural" generalization of convexity for asymmetric variational inequality problem functions? Figure 2 summarizes the results of this section.

TABLE 2
Several types of monotonicity.

Type of monotonicity imposed upon f	**Definition***	**Differential condition***
monotone on K	$[f(x) - f(y)](x - y) \geq 0$	$\nabla f(x)$ p.s.d.$^{+}$
strongly f-monotone on K	$\exists a > 0$, $[f(x) - f(y)](x - y) \geq a \parallel f(x) - f(y) \parallel_2^2$,	$\exists a > 0$ $[\nabla f(x)^t - a\nabla f(x)^t \nabla f(x_2)]$ p.s.d.$^{+}$
strictly strongly f-monotone on K^{***}	$\exists a > 0$, $[f(x) - f(y)](x - y) > a \parallel f(x) - f(y) \parallel_2^2$	$\exists a > 0$, $[\nabla f(x)^t - a\nabla f(x)^t \nabla f(y)]$ p.d.$^{++}$
strictly monotone on K^{**}	$[f(x) - f(y)](x - y) > 0$	$\nabla f(x)$ p.d.$^{++}$
strongly monotone on K^{**}	$\exists a > 0$, $[f(x) - f(y)](x - y) \geq a \parallel x - y \parallel_2^2$	$\nabla f(x)$ uniformly p.d.$^{++}$
* Definition holds for all $x, y \in K$ or all $x \in K$ ** Condition holds for $x \neq y$ *** Condition holds for $f(x) \neq f(y)$		$^{+}$ p.s.d. means positive semidefinite $^{++}$ p.d. means positive definite

3.1. Convexity and the weak differential form of strong-f-monotonicity. First, consider the symmetric case in which $f = \nabla F$ for some twice differentiable function F, and so $\nabla f(x)$ is symmetric for all $x \in K$. In this case, since strong-f-monotonicity implies monotonicity, $\nabla f(x)$ is positive semidefinite for all $x \in K$ (see Table 2), and so strong-f-monotonicity implies that F is a convex function. *Is the converse true?* That is, does convexity imply strong-f-monotonicity and, if not, how far from convexity can we stray and yet ensure that the function f is strongly f-monotone?

The following results give a partial answer to these questions. We begin by giving two examples, one showing that convexity of F does not imply that f is strongly f-monotone and a second showing that even when F is convex over a compact set, it still might not be strongly f-monotone. In the second example, we use the fact (see [19]) that a differentiable function $f(x)$ from R^n to R^n is strongly f-monotone if and only if it satisfies a differential condition: for some $a > 0$,

$$w^t \nabla f(x)^t w \geq a w^t \nabla f(x)^t \nabla f(y) w \quad \text{for all } x, y, w \in R^n. \tag{6}$$

Example 2. Consider the variational inequality with the feasible noncompact set $K = \{x = (x_1, x_2) \in R^2 : \; x_1 \geq 0, \; x_2 \geq 0\}$ and the problem function $f(x) = (x_1^2, 1)$. $f(x) = \nabla F(x)$ with $F(x) = \frac{x_1^3}{3} + x_2$, which is a convex function over K. In this case, f is not strongly f-monotone on K since for $y = (y_1, y_2)$ and $x = (x_1, x_2)$, with $x_1 = y_1 + 2$, *no* constant $a > 0$ satisfies the condition

$$[f(x) - f(y)]^t [x - y] = 2[2y_1 + 4] \geq a \|f(x) - f(y)\|^2 = a[2y_1 + 4]^2 \quad \text{for all} \quad y \in K.$$

Example 3. Consider the variational inequality with the feasible set $K = \{x = (x_1, x_2) \in R^2 : \; x_1 \leq 1, \; x_2 \leq 1, \, x_1 + x_2 \geq 1, \; x_1 \leq x_2\}$ and problem function

$$f(x) = \left(\frac{x_1^2}{2} - \frac{(1 - x_2)^2}{2}, -(1 - x_1)(1 - x_2) \right).$$

$\nabla f(x) = [\begin{smallmatrix} x_1 & 1-x_2 \\ 1-x_2 & 1-x_1 \end{smallmatrix}]$ is a symmetric, positive semidefinite matrix over K.

$$F(x) = \frac{x_1^3}{6} + \frac{(1 - x_1)(1 - x_2)^2}{2}$$

is convex in K, since its Hessian matrix is $\nabla^2 F(x) = \nabla f(x)$. For the points $x = (1, 1)$ and $y = (\frac{1}{2}, \frac{1}{2})$, $\nabla f(x) = [\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}]$ and $\nabla f(y) = \frac{1}{2}[\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}]$. Then for all $w = (w_1, w_2) \in R^2$,

$$w^t \nabla f(x)^t w = w_1^2 \quad \text{and} \quad w^t \nabla f(x)^t \nabla f(y) w = \frac{1}{2}(w_1^2 + w_1 w_2),$$

which implies that *no* constant $a > 0$ satisfies the condition

$$w^t \nabla f(x)^t w \geq a w^t \nabla f(x)^t \nabla f(y) w \text{ for all } w = (w_1, w_2) \in R^2,$$

since for $\lim w_1 = 0$ and $w_2 = 1$,

$$\lim \frac{w^t \nabla f(x)^t \nabla f(y) w}{w^t \nabla f(x)^t w} = \lim \frac{1}{2} + \lim \frac{w_2}{2w_1} = +\infty.$$

In the following development, we use a weak differential form of the strong-f-monotonicity condition (obtained by setting $x = y$ in the differential form).

DEFINITION 4. *A function satisfies the weak differential form of strong-f-monotonicity if for all $w \in R^n$ and $x \in K$, some constant $a > 0$ satisfies the condition*

$$w^t \nabla f(x) w \geq a w^t \nabla f(x)^t \nabla f(x) w. \tag{7}$$

For notational convenience we will say that f satisfies the strong form if it satisfies (6) and satisfies the weak form if it satisfies (7). Note then that $w^t \nabla f(x) w = 0$ for any x and w implies that $w^t \nabla f(x)^t \nabla f(x) w = 0$ and $\nabla f(x) w = 0$. Luo and Tseng [18] have studied a class of matrices $B = \nabla f(x)$ that satisfy this property.

DEFINITION 5 (see Luo and Tseng [18]). *A matrix B is positive semidefinite plus (p.s.d. plus) if it is positive semidefinite and*

$$\text{if } x^t B x = 0 \text{ implies that } Bx = 0.$$

Luo and Tseng [18] have shown that the class of p.s.d. plus matrices B is equivalent to the class of matrices that can be decomposed into the product $B = P^t P_1 P$ for some P_1 positive definite matrix and some (possibly nonsquare) matrix P. Note that every symmetric, positive semidefinite matrix B is p.s.d. plus, since in this case $B = H^t H$ for some matrix H and so $x^t B x = 0$ implies $x^t H^t H x = 0$, which implies that $Hx = 0$ and therefore $Bx = 0$.

After stating two elementary matrix results, we then establish a relationship between the convexity of F and the weak form for f.

LEMMA 3.1. Suppose that M is a symmetric, positive semidefinite $n \times n$ matrix and that $d > 0$ is the largest eigenvalue of M. Then if $a \geq \frac{1}{d}$,

$$w^t M w \geq a w^t M^t M w \text{ for all } w \in R^n.$$

Proof. Any symmetric, positive semidefinite, nonzero matrix has an orthogonal representation (see [33]) $M = Q^t D Q$ for some orthogonal matrix Q and D, a diagonal positive semidefinite matrix whose elements are the eigenvalues of M. Then

$$w^t M w = w^t Q^t D Q w$$

$$\text{while } w^t M^t M w = w^t Q D^t D Q w.$$

Requiring $w^t M w \geq a w^t M^t M w$ for some constant $a > 0$ is equivalent to requiring

$$\sum_i d_i (Qw)_i^2 \geq a \sum_i d_i^2 (Qw)_i^2.$$

Since $\max_i d_i \leq d$ and the matrix M is positive semidefinite, setting $a = \frac{1}{d}$ gives $d_i \geq a d_i^2$ for all i, which implies the inequality. □

We also make an observation about a general asymmetric matrix M.

LEMMA 3.2. *If the matrix M^2 is positive semidefinite, then $\|\frac{M+M^t}{2}w\|^2 \geq \frac{\|Mw\|^2}{4}$.*

Proof. When M^2 is positive semidefinite,

$$\left\|\frac{M+M^t}{2}w\right\|^2 = w^t\left(\frac{M+M^t}{2}\right)^t\left(\frac{M+M^t}{2}\right)w$$

$$= w^t\frac{M^2+(M^2)^t+M^tM+MM^t}{4}w \geq \frac{\|Mw\|^2}{4}. \qquad \square$$

PROPOSITION 3.1 (see [20]). *Suppose that $F : K \subseteq R^n \to R$ is a twice continuously differentiable function and that the maximum eigenvalue of the Hessian matrix $\nabla^2 F(x) = \nabla f(x)$ is uniformly bounded on K; i.e., if $d_i(x)$ is the ith eigenvalue of $\nabla^2 F(x)$, then $\sup_{x\in K}[\max_{\{i=1,\ldots,n\}} d_i(x)] \leq d$ for some positive constant d. Then if $\nabla^2 F(x)$ is positive semidefinite for all $x \in K$, F is convex if and only if for all $x \in K$ and for all $w \in R^n$, $w^t\nabla f(x)w \geq aw^t\nabla f(x)^t\nabla f(x)w$ for some constant $a > 0$.*

Proof. "⇒" The proof follows from Lemma 3.1 since the Hessian matrix $\nabla^2 F(x) = \nabla f(x)$ is positive semidefinite for all $x \in K$.

"⇐" The converse is easy to see. When the strong form (6) holds for all $x = y$, then the Jacobian matrix is positive semidefinite and, therefore, the function F is convex. $\square$

COROLLARY 3.1. *Let $F : K \subseteq R^n \to R$ be a continuous function and ∇F a Lipschitz continuous function in the sense that $||\nabla^2 F(x)|| \leq L$ for some positive constant L. Then F is a convex function if and only if ∇F is a weakly strong-f-monotonicity function.*

Remark. Proposition 3.1 and results in [19] imply that on a compact set (or a problem function F with bounded eigenvalues of its Hessian matrix over K) F is convex if and only if $\|I - a\nabla^2 F(x)\| \leq 1$ for all $x \in K$. In applying the convergence of the general iterative scheme (5), researchers have imposed a norm condition (see [7], [29]).

Norm condition (inequality form):

$$\|g_x^{-1/2}(x,x)g_y(x,x)g_x^{-1/2}(x,x)\| \leq 1 \quad \text{for all } x \in K. \tag{8}$$

Convergence results impose this condition as a strict inequality. Our previous observation implies the following result.

COROLLARY 3.2. *Let $f(x) = \nabla F(x)$ and $\nabla f(x)$ be a symmetric matrix. If $g_x(x,x)$ is a positive definite and symmetric matrix for all $x \in K$ and the function $g(x,y)$ satisfies the conditions* A1 *and* A2*, then on a compact set (or a problem function F with bounded eigenvalues of its Hessian matrix over K), F is convex if and only if $g(x,y)$ satisfies the norm condition* (8).

Proof. The proof follows from Proposition 3.1 and Theorem 6 in [22]. $\square$

Corollary 3.2 shows that the less than or equal form of the norm condition (on a compact set, or a problem function F where the Hessian matrix has bounded eigenvalues over K) together with the symmetry of $\nabla f(x) = \nabla^2 F(x)$, is equivalent to convexity. Therefore, we can view the norm condition as a form of "generalization" of convexity and, therefore, as a "natural" condition to assume for asymmetric problems.

Although Proposition 3.1 shows a connection between convexity and strong-f-monotonicity, it does not show that convexity implies strong-f-monotonicity since it requires $x = y$ in the differential condition. As our previous examples show, we need to impose additional structure on f or on K to ensure that the convexity of F implies regular strong-f-monotonicity. In the following discussion, we address this issue, considering two questions:

"What is the analog of Proposition 3.1 *for the general asymmetric case?"*

"How asymmetric can the Jacobian matrix be?"

PROPOSITION 3.2. *Suppose that f is a Lipschitz continuous function in the sense $\|\nabla f(x)\| \leq L$ for some positive constant L. If the Jacobian matrix $\nabla f(x)$ and the squared Jacobian matrix $(\nabla f(x))^2$ of the problem function f are positive semidefinite, then f satisfies the weak form* (7).

Moreover, the weak form (7) for f implies the positive semidefiniteness of the Jacobian matrix of the problem function f, i.e., ordinary monotonicity.

Proof. Since $\nabla f(x)$ is positive semidefinite, so is the symmetric matrix $\frac{\nabla f(x)^t + \nabla f(x)}{2}$, which equals $\nabla^2 F(x)$ for some convex function F. Therefore, Corollary 3.1 as applied to the symmetric matrix $\frac{\nabla f(x)^t + \nabla f(x)}{2}$ implies that for some constant $a > 0$ and for all x and w,

$$w^t \nabla f(x) w = w^t \frac{\nabla f(x) + \nabla f(x)^t}{2} w \geq a \left\| \frac{\nabla f(x) + \nabla f(x)^t}{2} w \right\|^2 .$$

Lemma 3.2 then implies that

$$a \left\| \frac{\nabla f(x) + \nabla f(x)^t}{2} w \right\|^2 \geq a \left\| \frac{\nabla f(x)}{2} w \right\|^2 \geq \frac{a}{4} w^t \nabla f(x)^t \nabla f(x) w,$$

which is the weak form (7). Furthermore, the weak form (7) implies the positive semidefiniteness of the Jacobian matrix. □

Hammond and Magnanti [13] originally introduced the condition of positive definiteness of the squared Jacobian matrix while establishing the convergence of the steepest descent method for variational inequalities. This condition implies that the Jacobian matrix cannot be "very" asymmetric. In fact, the squared Jacobian matrix is positive definite when the angle between $\nabla f(x) w$ and $\nabla f(x)^t w$ is less than 90 degrees for all $w \in R^n$.

PROPOSITION 3.3. *The converse of the statements in Proposition* 3.2 *are not valid.*

Proof. To establish this result, we will provide counterexamples.

Example 4. The weak form (7) does not imply that the square of the Jacobian matrix is positive semidefinite. Consider the function $f(x) = Mx$ with the Jacobian matrix $M = \left[\begin{smallmatrix} c & b \\ -b & c \end{smallmatrix}\right]$ and let $0 < c < b$. Then $M^2 = \left[\begin{smallmatrix} c^2-b^2 & 2cb \\ -2cb & c^2-b^2 \end{smallmatrix}\right]$ is a negative definite matrix, which means that M^2 is not positive semidefinite. Nevertheless the function $f(x) = Mx$ is strongly f-monotone (and therefore satisfies the weak form (7)) with a strong-f-monotonicity constant $a = \frac{c}{b^2+c^2} > 0$ since

$$w^t M w = c\|w\|^2 = \frac{c}{b^2+c^2}(b^2+c^2)\|w\|^2 = a\|Mw\|^2.$$

Example 5. The differential form of monotonicity does not imply the weak form (7). Consider the function $f(x) = Mx$ with the Jacobian matrix $M = \left[\begin{smallmatrix} 0 & -b \\ b & 0 \end{smallmatrix}\right]$ and $b \neq 0$. M is a positive semidefinite matrix. $M^t - aM^tM = \left[\begin{smallmatrix} -ab^2 & b \\ -b & -ab^2 \end{smallmatrix}\right]$ and since $b \neq 0$, there is no value of the constant $a > 0$ for which $M^t - aM^tM$ is a positive semidefinite matrix, since when $b \neq 0$ for all values of $a > 0$, $M^t - aM^tM$ is negative definite. Therefore, $f(x) = Mx$ is not a strongly f-monotone function (which in this case coincides with the weak form (7)). □

3.2. The weak and strong forms of strong-f-monotonicity. To this point, we have shown the relationship between convexity (monotonicity of f) and the weak form (7) in the symmetric case. In the asymmetric case, we have shown that the positive semidefiniteness of the squared Jacobian matrix together with the positive semidefiniteness of the Jacobian matrix imply the weak form (7) and the monotonicity condition. In the following discussion, we carry this analysis further to characterize the strong-f-monotonicity condition for general nonlinear problem maps. We show with a suitable boundedness assumption that

(i) the weak form is equivalent to ∇f being p.s.d. plus,

(ii) if ∇f satisfies a uniform version of p.s.d. plus, then it satisfies the strong form.

In stating the following result, we assume that $\nabla f(x)$ is a p.s.d. plus matrix and therefore we can rewrite it as

$$\nabla f(x) = P(x)^t \begin{bmatrix} 0 & 0 & 0 \\ 0 & P_0(x) & 0 \\ 0 & 0 & 0 \end{bmatrix} P(x).$$

First, we make the following observation.

Lemma 3.3. *Every p.s.d. plus matrix $M(x)$ can be rewritten as*

$$M(x) = P(x)^t \begin{bmatrix} 0 & 0 & 0 \\ 0 & P_0(x) & 0 \\ 0 & 0 & 0 \end{bmatrix} P(x)$$

for some $n_1(x) \times n_1(x)$ positive definite matrix $P_0(x)$ and some square matrix $P(x)$. Conversely, any matrix $M(x)$ that is of this form is also p.s.d. plus.

Proof. "$\Leftarrow$" Luo and Tseng [18] have shown that it suffices to show that whenever $w^t M(x) w = 0$, then $M(x)w = 0$. Suppose $M(x)$ can be written as

$$M(x) = P(x)^t \begin{bmatrix} 0 & 0 & 0 \\ 0 & P_0(x) & 0 \\ 0 & 0 & 0 \end{bmatrix} P(x).$$

Then

$$w^t M(x) w = w^t P(x)^t \begin{bmatrix} 0 & 0 & 0 \\ 0 & P_0(x) & 0 \\ 0 & 0 & 0 \end{bmatrix} P(x) w = [v^t, y^t, z^t] \begin{bmatrix} 0 & 0 & 0 \\ 0 & P_0(x) & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ y \\ z \end{bmatrix}$$

$$= y^t P_0(x) y = 0, \text{ with } P(x)w = \begin{bmatrix} v \\ y \\ z \end{bmatrix},$$

and y is a vector that has the same dimension as $P_0(x)$. But since $P_0(x)$ is a positive definite matrix, $y = 0$ and, therefore,

$$P(x)^t \begin{bmatrix} 0 \\ P_0(x)y \\ 0 \end{bmatrix} = P(x)^t \begin{bmatrix} 0 & 0 & 0 \\ 0 & P_0(x) & 0 \\ 0 & 0 & 0 \end{bmatrix} P(x)w = M(x)w = 0.$$

Therefore, $M(x)$ is also a p.s.d. plus matrix.

"$\Rightarrow$" Conversely, if an $n \times n$ matrix $M(x)$ is p.s.d. plus, then for some $n_1(x) \times n_1(x)$ positive definite matrix $P_0(x)$, we can rewrite $M(x) = P''(x)^t P_0(x) P''(x)$, with $P''(x)$ an $n_1 \times n$ matrix. Then setting $P(x)^t = [P'(x)^t, P''(x)^t, P'''(x)^t]$ for any matrices $P'(x)$ and $P'''(x)$, with appropriate dimensions, we can conclude that $M(x)$ can be rewritten as

$$P(x)^t \begin{bmatrix} 0 & 0 & 0 \\ 0 & P_0(x) & 0 \\ 0 & 0 & 0 \end{bmatrix} P(x). \qquad \square$$

We should observe that in this representation, the matrix $P_0(x)$ is not necessarily unique. We let $Q(x)$ be a submatrix of $P(x)^t P(x)$ defined as follows: let I be an identity matrix with the same dimension as $P_0(x)$; then

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & Q(x) & 0 \\ 0 & 0 & 0 \end{bmatrix} = P(x) \begin{bmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix} P(x)^t,$$

so $Q(x) = D(x)D(x)^t + E(x)E(x)^t + F(x)F(x)^t$ when

$$P(x) = \begin{bmatrix} A(x) & B(x) & C(x) \\ D(x) & E(x) & F(x) \\ G(x) & H(x) & J(x) \end{bmatrix}.$$

PROPOSITION 3.4. *Suppose that the matrix $\nabla f(x)$ is p.s.d. plus for all $x \in K$, then the weak form* (7) *holds whenever the maximum eigenvalue of the matrix*

$$B(x) = \left[\frac{P_0(x) + P_0(x)^t}{2}\right]^{-\frac{1}{2}} P_0(x)^t Q(x) P_0(x) \left[\frac{P_0(x) + P_0(x)^t}{2}\right]^{-\frac{1}{2}}$$

is bounded over the feasible set K by a constant d.

Conversely, if the weak form (7) holds, then the matrix $\nabla f(x)$ is a p.s.d. plus.

Proof. Suppose $\nabla f(x)$ is a p.s.d. plus matrix. Then

$$w^t \nabla f(x) w = w^t P(x)^t \begin{bmatrix} 0 & 0 & 0 \\ 0 & P_0(x) & 0 \\ 0 & 0 & 0 \end{bmatrix} P(x) w = w^t P(x)^t \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{P_0(x)+P_0(x)^t}{2} & 0 \\ 0 & 0 & 0 \end{bmatrix} P(x) w$$

$$= [v^t, y^t, z^t] \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{P_0(x)+P_0(x)^t}{2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ y \\ z \end{bmatrix}, \text{ with } P(x) w = \begin{bmatrix} v \\ y \\ z \end{bmatrix}$$

and y a vector with the same dimension as $P_0(x)$. Then since $P_0(x)$ is positive definite,

$$w^t \nabla f(x) w = y^t \left[\frac{P_0(x) + P_0(x)^t}{2}\right] y = y^t \left[\frac{P_0(x) + P_0(x)^t}{2}\right]^{\frac{1}{2}} \cdot \left[\frac{P_0(x) + P_0(x)^t}{2}\right]^{\frac{1}{2}} y.$$

Furthermore,

$$w^t \nabla f(x)^t \nabla f(x) w = w^t P(x)^t \begin{bmatrix} 0 & 0 & 0 \\ 0 & P_0(x)^t & 0 \\ 0 & 0 & 0 \end{bmatrix} P(x) P(x)^t \begin{bmatrix} 0 & 0 & 0 \\ 0 & P_0(x) & 0 \\ 0 & 0 & 0 \end{bmatrix} P(x) w$$

$$= [0, y^t P_0(x)^t, 0] P(x) P(x)^t \begin{bmatrix} 0 \\ P_0(x) y \\ 0 \end{bmatrix} = y^t P_0(x)^t Q(x) P_0(x) y$$

$$= y^t \left[\frac{P_0(x) + P_0(x)^t}{2}\right]^{\frac{1}{2}} \left[\frac{P_0(x) + P_0(x)^t}{2}\right]^{-\frac{1}{2}} P_0(x)^t Q(x) P_0(x)$$

$$\cdot \left[\frac{P_0(x) + P_0(x)^t}{2}\right]^{-\frac{1}{2}} \left[\frac{P_0(x) + P_0(x)^t}{2}\right]^{\frac{1}{2}} y.$$

Therefore, if $b = [\frac{P_0(x)+P_0(x)^t}{2}]^{\frac{1}{2}} y$, then $w^t \nabla f(x) w = b^t b$ and $w^t \nabla f(x)^t \nabla f(x) w = b^t [\frac{P_0(x)+P_0(x)^t}{2}]^{-\frac{1}{2}} P_0(x)^t Q(x) P_0(x) [\frac{P_0(x)+P_0(x)^t}{2}]^{-\frac{1}{2}} b$. Since the maximum eigenvalue of $B(x)$ is bounded over the feasible set K by a constant d, $\frac{b^t B(x) b}{b^t b} \leq d$ and so for $a = \frac{1}{d}$ we have $w^t \nabla f(x) w \geq a w^t \nabla f(x)^t \nabla f(x) w$ for all $w \in R^n$ and $x \in K$.

Conversely, if for some constant $a > 0$, $w^t \nabla f(x) w \geq a w^t \nabla f(x)^t \nabla f(x) w$ for all $x \in K$ and $w \in R^n$ then, as we have observed previously, $\nabla f(x)$ is a p.s.d. matrix and therefore p.s.d. plus. □

Remark. In the symmetric case, the matrix $B(x) = [\frac{P_0(x)+P_0(x)^t}{2}]^{-\frac{1}{2}} P_0(x)^t Q(x) \cdot P_0(x) [\frac{P_0(x)+P_0(x)^t}{2}]^{-\frac{1}{2}}$ becomes $B(x) = ([P_0(x)]^{\frac{1}{2}})^t Q(x) [P_0(x)]^{\frac{1}{2}}$. Furthermore, $B(x) = D(x)$ since $P_0(x) = D(x)$ is a diagonal matrix whose diagonal elements are the positive eigenvalues $d_i(x)$ of $\nabla f(x)$ and $Q(x) = I$. Therefore, requiring the maximum eigenvalue of $B(x)$ to be bounded over the feasible set K coincides with the assumption of Proposition 3.1, i.e., $\sup_{x \in K} [\max_i d_i(x)] \leq d$. So Proposition 3.4 is a natural generalization of Proposition 3.1.

COROLLARY 3.3 (see [24]). *Suppose a variational inequality problem is affine with $f(x) = Mx - c$. Then the matrix M is p.s.d. plus if and only if its problem function f is strongly f-monotone.*

The proof of this result follows directly from Proposition 3.4 since in the affine case the weak form (7) coincides with the strong form (6).

Proposition 3.1 showed the relationship between convexity and the weak form (7). For the asymmetric case, the analog of convexity is monotonicity. Therefore, we might wish to address the following question. *What is the relationship between monotonicity and strong-f-monotonicity for the general asymmetric case?* Example 4 in the proof of Proposition 3.3 shows that monotonicity does not imply strong-f-monotonicity. What additional conditions do we need to impose on the feasible set K and the problem function f other than compactness to ensure that monotonicity implies strong-f-monotonicity? Example 3 suggests that even in the symmetric case, compactness and convexity are not enough. We need to impose additional assumptions. For this development, we use the following definition which applies to general asymmetric matrices.

DEFINITION 6. *A matrix $M(x)$ is uniformly p.s.d. plus if for every point $x \in K$, we can express $M(x)$ as*

$$M(x) = P^t \begin{bmatrix} 0 & 0 & 0 \\ 0 & P_0(x) & 0 \\ 0 & 0 & 0 \end{bmatrix} P,$$

with P independent of x. $P_0(x)$ is a positive definite or zero matrix of fixed dimension $n_1 \times n_1$ and is always in the same location in the bracketed matrix.

Remark. As our nomenclature shows, every uniformly p.s.d. plus matrix $M(x)$ is also p.s.d. plus.

Before continuing, we state a preliminary result about uniformly p.s.d. plus matrices. We first set some notation. In the representation of a p.s.d. plus matrix as specified in Definition 6, suppose we partition P compatibly with

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & P_0(x) & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ as } P = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix}.$$

PROPOSITION 3.5. *If $\nabla f(x)$ is a uniformly p.s.d. plus matrix, then f is strongly f-monotone whenever, for the values of x_1 for which the matrix $P_0(x_1)$ is positive definite, the maximum eigenvalue of the matrix $B(x_1,x_2)^t B(x_1,x_2)$ is bounded over the feasible set K by some constant d^2, with*

$$B(x_1,x_2) = \left[\frac{P_0(x_1)+P_0(x_1)^t}{2}\right]^{-\frac{1}{2}} P_0(x_1)^t Q P_0(x_2) \left[\frac{P_0(x_1)+P_0(x_1)^t}{2}\right]^{-\frac{1}{2}}.$$

Proof. First, we observe that if $x_1 \in K$ and $P_0(x_1) = 0$, then $\nabla f(x_1) = 0$ and, therefore, for all $a > 0$ and $x_2 \in K$, $aw^t \nabla f(x_1)^t \nabla f(x_2) w \leq w^t \nabla f(x_1)^t w$, which is the strong form (6). Now suppose that $x_1 \in K$ and $P_0(x_1)$ is positive definite. The uniform p.s.d. plus property implies that

$$w^t \nabla f(x_1)^t \nabla f(x_2) w = w^t P^t \begin{bmatrix} 0 & 0 & 0 \\ 0 & P_0(x_1)^t & 0 \\ 0 & 0 & 0 \end{bmatrix} P P^t \begin{bmatrix} 0 & 0 & 0 \\ 0 & P_0(x_2) & 0 \\ 0 & 0 & 0 \end{bmatrix} Pw.$$

Then

$$w^t \nabla f(x_1)^t \nabla f(x_2) w = w^t P^t \begin{bmatrix} 0 & 0 & 0 \\ P_0(x_1)^t D & P_0(x_1)^t E & P_0(x_1)^t F \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & D^t P_0(x_2) & 0 \\ 0 & E^t P_0(x_2) & 0 \\ 0 & F^t P_0(x_2) & 0 \end{bmatrix} Pw$$

$$= [v^t, y^t, z^t] \begin{bmatrix} 0 & 0 & 0 \\ P_0(x_1)^t D & P_0(x_1)^t E & P_0(x_1)^t F \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & D^t P_0(x_2) & 0 \\ 0 & E^t P_0(x_2) & 0 \\ 0 & F^t P_0(x_2) & 0 \end{bmatrix} \begin{bmatrix} v \\ y \\ z \end{bmatrix},$$

$$\text{with } Pw = \begin{bmatrix} v \\ y \\ z \end{bmatrix}$$

and y a vector with the same dimension as $P_0(x_i)$, $i = 1, 2$. Let $Q = EE^t + DD^t + FF^t$. Therefore, since $\bar{P}_0(x_1) = \frac{P_0(x_1)+P_0(x_1)^t}{2}$ is a positive definite and symmetric matrix,

$$w^t \nabla f(x_1)^t \nabla f(x_2) w = y^t P_0(x_1)^t Q P_0(x_2) y$$

$$= y^t [\bar{P}_0(x_1)]^{\frac{1}{2}} [\bar{P}_0(x_1)]^{-\frac{1}{2}} P_0(x_1)^t Q P_0(x_2) [\bar{P}_0(x_1)]^{-\frac{1}{2}} [\bar{P}_0(x_1)]^{\frac{1}{2}} y.$$

If $b = [\bar{P}_0(x_1)]^{\frac{1}{2}} y$, then $w^t \nabla f(x_1)^t w = b^t b$ and

$$w^t \nabla f(x_1)^t \nabla f(x_2) w = b^t [\bar{P}_0(x_1)]^{-\frac{1}{2}} P_0(x_1)^t Q P_0(x_2) [\bar{P}_0(x_1)]^{-\frac{1}{2}} b$$

$$= b^t B(x_1, x_2) b \leq b^t b \|B(x_1, x_2)\| \leq d b^t b = d w^t \nabla f(x_1)^t w,$$

since the maximum eigenvalue of $B(x_1, x_2)^t B(x_1, x_2)$ is bounded over the feasible set K by a constant d^2. This inequality shows that for all $x_1, x_2 \in K$ and $w \in R^n$, the constant $a = \min\{\frac{1}{d}, 1\} > 0$ satisfies the condition

$$a w^t \nabla f(x_1)^t \nabla f(x_2) w \leq w^t \nabla f(x_1)^t w,$$

which is the strong form (6) (see Table 2). Therefore, f is strongly f-monotone. $\square$

Remarks.

1. In Proposition 3.4 we show that when $\nabla f(x)$ is uniform p.s.d. plus, the weak form (7) holds. The proof of Proposition 3.5 permits us to show that when $\nabla f(x)$ is uniform p.s.d. plus and K is compact, the weak and strong forms of strong-f-monotonicity are equivalent. To establish this result, we note that the steps of Proposition 3.5 and the fact that the matrix $[P_0(x_1)^t Q P_0(x_1)]$ is positive definite and symmetric imply the following result. Let $B(x_1, x_2)$ be defined as in Proposition 3.5 and let d^2 be the maximum eigenvalue of $B(x_1, x_2)^t B(x_1, x_2)$ over the compact set K.

$$w^t \nabla f(x_1)^t \nabla f(x_2) w = y^t P_0(x_1)^t Q P_0(x_2) y$$

$$= y^t [P_0(x_1)^t Q P_0(x_1)]^{\frac{1}{2}} [P_0(x_1)^t Q P_0(x_1)]^{-\frac{1}{2}} P_0(x_1)^t Q P_0(x_2)$$

$$\cdot [P_0(x_1)^t Q P_0(x_1)]^{\frac{1}{2}} [P_0(x_1)^t Q P_0(x_1)]^{\frac{1}{2}} y$$

$$= b^t B(x_1, x_2) b \leq \|B(x_1, x_2)\| b^t b \leq d b^t b$$

$$= y^t P_0(x_1)^t Q P_0(x_1) y = w^t \nabla f(x_1)^t \nabla f(x_1) w,$$

with $b = [P_0(x_1)^t Q P_0(x_1)]^{\frac{1}{2}} y$. Therefore, $w^t \nabla f(x_1)^t \nabla f(x_2) w \leq d w^t \nabla f(x_1)^t \nabla f(x_1) w$, which implies that the weak (7) and the strong forms (6) are equivalent.

2. Remark (1) implies that if $\nabla f(x)$ is uniformly p.s.d. plus and K is compact, then for the general iterative scheme (5) (see [7], [29]), the strong form (6) is *equivalent to* the norm condition (8) in a less than or equal form.

We now show how to check the uniform p.s.d. plus condition.

Definition 7 (see Sun [34]). *A matrix $M(x)$ satisfies the Hessian similarity property over the set K if* (i) *$M(x)$ is a positive semidefinite matrix for all $x \in K$ and* (ii) *for all $w \in R^n$ and $y, z \in K$ and for some constant $r \geq 1$, $M(x)$ satisfies the condition*

$$r w^t M(z) w \geq w^t M(y) w \geq \frac{1}{r} w^t M(z) w.$$

Matrices that do not depend on x, i.e., $M = M(x)$ for all x, and positive definite matrices on compact sets K satisfy this property. In the latter case, we can choose

r as the ratio of the maximum eigenvalue of $M(x)$ over K divided by the minimum eigenvalue of $M(x)$ over K.

Sun [34] has established the following result.

LEMMA 3.4. *If a matrix is positive semidefinite and symmetric and satisfies the Hessian similarity property, then it also satisfies the uniform p.s.d. plus property.*

COROLLARY 3.4. *If for a variational inequality problem VI(f,K), $\nabla f(x)$ is a symmetric, positive definite matrix and the set K is compact, then $\nabla f(x)$ satisfies the uniform p.s.d. plus property and the problem function f is strongly f-monotone.*

Proof. When $\nabla f(x)$ is a symmetric, positive definite matrix and the set K is compact, $\nabla f(x)$ satisfies the Hessian similarity condition. Lemma 3.4 implies the uniform p.s.d. plus property. Therefore, Proposition 3.5 implies that f is a strongly f-monotone problem function. □

COROLLARY 3.5. *If the Jacobian matrix $\nabla f(x)$ of a variational inequality problem is symmetric and positive semidefinite and satisfies the Hessian similarity condition and the set K is compact, then the problem function f is strongly f-monotone.*

Proof. By Lemma 3.4, the Jacobian matrix $\nabla f(x)$ satisfies the uniform p.s.d. plus condition and so the result follows from Proposition 3.5. The following result provides a generalization of Proposition 3.5. □

PROPOSITION 3.6. *Suppose that $\nabla f(x)$ can be written as*

$$\nabla f(x) = P^t \begin{bmatrix} P_1(x) & ... & 0 & ... & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & ... & P_i(x) & ... & 0 \\ \vdots & ... & \vdots & \ddots & \vdots \\ 0 & ... & 0 & ... & P_m(x) \end{bmatrix} P.$$

The matrices $P_i(x)$ for $i = 1, 2, \ldots, m$ are either positive definite or zero and for all $i = 1, 2, \ldots, m$, they have the same dimension $n_1 \times n_1$ for all x; moreover, $PP^t = I$. Let

$$B_i(x_1, x_2) = \left[\frac{P_i(x_1) + P_i(x_1)^t}{2}\right]^{-\frac{1}{2}} P_i(x_1)^t Q P_i(x_2) \left[\frac{P_i(x_1) + P_i(x_1)^t}{2}\right]^{-\frac{1}{2}};$$

then f is a strongly f-monotone function whenever for $i = 1, \ldots, m$ and for the values of x_1 for which the matrix $P_i(x_1)$ is positive definite, the matrix $B_i(x_1, x_2)^t B_i(x_1, x_2)$ has maximum eigenvalue that is bounded over the feasible set K by a constant d_i^2.

Proof. We will first define the matrix

$$D_i(x) = P^t \begin{bmatrix} 0 & ... & 0 & ... & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & ... & P_i(x) & ... & 0 \\ \vdots & ... & \vdots & \ddots & \vdots \\ 0 & ... & 0 & ... & 0 \end{bmatrix} P.$$

Then $\nabla f(x) = \sum_{i=1}^m D_i(x)$. Observe that since $PP^t = I$, $D_i(x_1) D_j(x_2) = 0$ for $i \neq j$. Therefore, Proposition 3.5 permits us to conclude that

$$aw^t \nabla f(x_1)^t \nabla f(x_2) w = aw^t \sum_{i,j=1}^m (D_i(x_1)^t D_j(x_2)) w = aw^t \sum_{i=1}^m (D_i(x_1)^t D_i(x_2)) w$$

$$= a \sum_{i=1}^{m} [w^t D_i(x_1)^t D_i(x_2) w] \leq \sum_{i=1}^{m} w^t D_i(x_1)^t w = w^t \nabla f(x_1)^t w$$

for $a = 1/d$ and $d = \max_{\{i=1,\ldots,m\}} d_i$. □

Corollary 3.6. *For a variational inequality problem VI(f,K) if the Jacobian matrix $\nabla f(x)$ is a diagonal positive semidefinite matrix and the set K is compact, then the problem function f is strongly f-monotone.*

Proof. The proof of this result follows directly from Proposition 3.6 since the diagonal positive semidefinite matrix $\nabla f(x)$ is the sum of uniform p.s.d. plus matrices, with $P_i(x)$ as 1×1 matrices that are zero or positive definite (zero or positive scalars in this case) and with $P = I$. □

Remarks.

(i) In Proposition 3.6 we could have made a more general “orthogonality” assumption that $D_i(x_1)D_j(x_2) = 0$ for $i \neq j$ and for all x_1, x_2, which is the central observation in its proof. Then Corollaries 3.3 through 3.6 and Proposition 3.7 would become special cases of Proposition 3.6.

(ii) The condition on $\nabla f(x)$ in Proposition 3.6 requires that the matrices $P_i(x)$ have fixed dimensions and occupy a fixed location in the block diagonal matrix of the $P_i(x)$s. Can these conditions be relaxed in any sense? Doing so would permit us to define a broader class for which a p.s.d. plus type of condition would imply strong-f-monotonicity.

Finally, we note that strong-f-monotonicity is related to the condition of firm nonexpansiveness (used, for example, by Lions and Mercier [17], Eckstein and Bertsekas [9]).

Definition 8. *A mapping $T : K \to K$ is firmly nonexpansive (or pseudocontractive) over the set K if*

$$\|T(x) - T(y)\|^2 \leq \|x - y\|^2 - \|[x - T(x)] - [y - T(y)]\|^2 \quad \textit{for all } x, y \in K.$$

Expanding $\|[x - T(x)] - [y - T(y)]\|^2$ as $\|x - y\|^2 + \|T(x) - T(y)\|^2 - 2[T(x) - T(y)]^t[x - y]$ and rearranging shows the following.

Proposition 3.7. *If a problem function f is strongly f-monotone for a constant $a \geq 1$, then it is firmly nonexpansive. Conversely, if a problem function f is firmly nonexpansive, then it is strongly f-monotone for the constant $a = 1$.*

Remark. To conclude this discussion, we note that most of the results in this paper, including the orthogonality theorem, can be easily extended to a more general form of variational inequality:

$$\text{find } \; x^* \in K : \quad f(x^*)^t(x - x^*) + F(x) - F(x^*) \geq 0 \quad \text{for all } x \in K,$$

with $f : K \to R^n$ a continuous function, $F : K \to R$ a continuous and convex function, and K a closed and convex subset of R^n.

Acknowledgment. We are grateful to the referees whose suggestions have led to considerable improvements to this paper.

REFERENCES

[1] B. H. Ahn and W. W. Hogan (1982), *On the convergence of the PIES algorithm for computing equilibria*, Oper. Res., 30, pp. 281–300.

[2] J. B. Baillon (1975), *Un théorème de type ergodique pour les contractions non linéaires dans un espace de Hilbert*, C.R. Acad. Sci. Paris Sér. A, t.28, pp. 1511–1514.

[3] D. P. Bertsekas and E. M. Gafni (1982), *Projection methods for variational inequalities with application to the traffic assignment problem*, Math. Programming, 17, pp. 139–159.

[4] G. Cohen (1988), *Auxiliary problem principle extended to variational inequalities*, J. Optim. Theory Appl., 59, pp. 325–333.

[5] S. Dafermos (1980), *Traffic equilibria and variational inequalities*, Transportation Sci., 14, pp. 42–54.

[6] S. Dafermos (1982), *Relaxation algorithms for the general asymmetric traffic equilibrium problem*, Transportation Sci., 16, pp. 231–240.

[7] S. Dafermos (1983), *An iterative scheme for variational inequalities*, Math. Programming, 26, pp. 40–47.

[8] J. C. Dunn (1973), *On recursive averaging processes and Hilbert space extensions of the contraction mapping principle*, J. Franklin Inst., 295, pp. 117–133.

[9] J. Eckstein and D. P. Bertsekas (1992), *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Math. Programming, 55, pp. 293–318.

[10] M. Florian and D. Hearn (1995), *Network equilibria*, in Handbook of Operations Research and Management Science: Vol. 8, Network Routing, M. Ball, T. Magnanti, C. Monma, and G. Neumhauser, eds., North-Holland, Amsterdam.

[11] D. Gabay (1983), *Applications of the method of multipliers to variational inequalities*, in Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems, M. Fortin and R Glowinski, eds., North-Holland, Amsterdam, pp. 299–332.

[12] J. H. Hammond (1984), *Solving Asymmetric Variational Inequality Problems and Systems of Equations with Generalized Nonlinear Programming Algorithms*, Ph.D. dissertation, Department of Mathematics, M.I.T., Cambridge, MA.

[13] J. H. Hammond and T. L. Magnanti (1987), *Generalized descent methods for asymmetric systems of equations*, Math. Operations Res., 12, pp. 678–699.

[14] P. T. Harker and J. S. Pang (1990), *Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications*, Math. Programming, 48, pp. 161–220.

[15] D. W. Hearn (1982), *The gap function of a convex program*, Oper. Res. Lett., 1, pp. 67–71.

[16] A. N. Kolmogorov and S. V. Fomin (1970), *Introduction to Real Analysis*, Dover Publications, New York.

[17] P. L. Lions and B. Mercier (1979), *Splitting algorithms for the sum of two nonlinear operators*, SIAM J. Numer. Anal., 16, pp. 964–979.

[18] Z. Luo and P. Tseng (1991), *A decomposition property for a class of square matrices*, Appl. Math. Lett., 4, pp. 67–69.

[19] T. L. Magnanti and G. Perakis (1995), *A Unifying Geometric Framework for Solving Variational Inequality Problems*, Math. Programming, 71, pp. 327–352.

[20] T. L. Magnanti and G. Perakis (1993), *On the Convergence of Classical Variational Inequality Algorithms*, presentation at the ORSA-TIMS national meeting, Chicago, May 1993.

[21] T. L. Magnanti and G. Perakis (1993), *From Frank-Wolfe to Steepest Descent*; *A Descent Framework for Solving $VIPs$*, forthcoming / ORSA-TIMS national meeting, Boston, May 1994.

[22] T. L. Magnanti and G. Perakis (1994), *Averaging paper for solving fixed point and variational inequality problems*, Math. Oper. Res., to appear.

[23] T. L. Magnanti and G. Perakis (1995), *Best Recursive Averaging Schemes for Solving Fixed Point Problems*, forthcoming/ INFORMS national meeting, Los Angeles, May 1995.

[24] P. Marcotte and D. Zhu (1993), *Co-coercivity and Its Role in the Convergence of Iterative Schemes for Solving Variational Inequalities*, Centre de Recherche sur les Transports, Université de Montréal, preprint.

[25] B. Martos (1975), *Nonlinear Programming: Theory and Methods*, North-Holland, Amsterdam.

[26] A. Nagurney (1992), *Network Economics: A Variational Inequality Approach*, Kluwer Academic Publishers, Norwell, MA.

[27] Z. Opial (1967), *Weak convergence of the successive approximation for non-expansive mappings in Banach spaces*, Bull. Amer. Math. Soc., 73, pp. 123–135.

[28] J. S. Pang (1994), *Complementarity problems*, in Handbook of global optimization, R. Horst and P. Pardalos, eds., Kluwer Academic Publishers, Norwell, MA.

[29] J. S. Pang and D. Chan (1982), *Iterative methods for variational and complementarity problems*, Math. Programming, 24, pp. 284–313.

[30] G. B. Passty (1979), *Ergodic convergence to a zero of the sum of monotone operators in Hilbert space*, J. Math. Anal. Appl., 72, pp. 383–390.

[31] G. Perakis (1992), *Geometric, Interior Point, and Classical Methods for Solving Finite Dimensional Variational Inequality Problems*, Ph.D. dissertation, Department of Applied Mathematics, Brown University, Providence, RI.

[32] R. T. Rockafellar (1976), *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14, pp. 877–898.

[33] G. Strang (1988), *Linear Algebra and its Applications*, 3rd ed., Harcourt Brace Jovanovich, New York, San Diego.

[34] J. Sun (1990), *An Affine Scaling Method for Linearly Constrained Convex Programs*, Department of Industrial Engineering and Management Sciences, Northwestern University, preprint.

[35] P. Tseng (1990), *Further applications of a matrix splitting algorithm to decomposition in variational inequalities and convex programming*, Math. Programming, 48, pp. 249–264.

SIAM J. OPTIM.
Vol. 7, No. 1, pp. 274–279, February 1997

COMPUTABLE ERROR BOUNDS FOR CONVEX INEQUALITY SYSTEMS IN REFLEXIVE BANACH SPACES[*]

SIEN DENG[†]

Abstract. In 1952, A. J. Hoffman proved a fundamental result of an error bound on the distance from any point to the solution set of a linear system in $\mathbb{R}^n$. In *SIAM J. Control*, 13 (1975), pp. 271–273, Robinson extended Hoffman's theorem to any system of convex inequalities in a normed linear space which satisfies the Slater constraint qualification and has a bounded solution set. This paper studies any system of convex inequalities in a reflexive Banach space which has an unbounded solution set. It is shown that Hoffman's error bound holds for such a system when a related convex system, which defines the recession cone of the solution set for the system, satisfies the Slater constraint qualification.

Key words. error bounds, recession cones, recession functions

AMS subject classifications. 90C25, 90C31, 49J52

PII. S1052623495284832

1. Introduction.

Consider the convex inequality system

$$x \in C \subset X, \quad F(x) \le 0, \tag{1}$$

where X is a real reflexive Banach space, C is a nonempty closed convex set, $F(x) = (f_1(x), \ldots, f_m(x))$ is a vector-valued function from X to $\mathbb{R}^m$, and each f_i is a continuous convex function on X. Let S be the set of all solutions to (1). We assume throughout that S is nonempty. For a given p with $1 \le p \le \infty$, Hoffman's error bound holds for the convex system (1) if there exists a positive constant τ such that

$$d(x, S) \le \tau \|[F(x)]_+\|_p \quad \text{for all } x \in C \subset X, \tag{2}$$

where $d(x,S) = \inf_{y \in S} \|x - y\|$, $[\cdot]_+$ is the positive part of a vector, and $\|\cdot\|$ and $\|\cdot\|_p$ denote the norm on X and the p-norm on $\mathbb{R}^m$, respectively.

For $X = \mathbb{R}^n$, problems of Hoffman's error bounds have been studied by many authors (see [5, 12, 1, 18, 6, 8, 9, 17]). For other related error bound results, see [4, 7, 13].

For X being an infinite-dimensional space, Robinson [14] proved that the error bound (2) holds when S is bounded and (1) satisfies the Slater constraint qualification; Ioffe [11] obtained the same bound when each f_i is a continuous linear function. But no one has demonstrated that (2) still holds under appropriate conditions when S is unbounded, and each f_i is not necessarily linear. This task will be undertaken here.

In this paper, we study how the information of the recession cone of S can be used to provide a *computable* constant τ such that (2) holds. Specifically, we show that if a related convex system (see (4) of section 2), which has the recession cone of S as its solution set, satisfies the Slater constraint qualification, then the error bound (2) holds. Our results are complementary to Robinson's.

We now briefly give the notation and some of the basic concepts used below. We denote the dual space of X by X^*. The spaces X and X^* are paired in duality by

[*] Received by the editors April 17, 1995; accepted for publication September 27, 1995.
http://www.siam.org/journals/siopt/7-1/28483.html

[†] Department of Mathematical Sciences, Northern Illinois University, DeKalb, IL 60115 (deng@math.niu.edu).

the continuous bilinear form

$$\langle x^*, x\rangle = x^*(x),$$

defined on $X^* \times X$. We denote the norms on X and X^* by $\|\cdot\|$ and $\|\cdot\|_*$, respectively.

For a nonempty closed convex set U in X, we denote the indicator function of U by $\delta_U(\cdot)$.

For a nonempty closed convex set U in X and $\tilde{x} \in U$, we define the normal cone to the set U at $\tilde{x}$, denoted by $N_U(\tilde{x})$, as follows [10, p. 47]:

$$N_U(\tilde{x}) = \{x^* \in X^* | \langle x^*, x - \tilde{x}\rangle \leq 0 \text{ for any } x \in U\}.$$

For a nonempty closed convex set U in X, we define the recession cone of U, denoted by U^∞, as follows [16, Theorem 2A (e)]:

$$U^\infty = \{u \in X| \text{ there exist sequences of scalars } \mu_i > 0 \text{ and } x_i \in U \text{ such that } \lim_i \mu_i = 0 \text{ and } \lim_i \mu_i x_i = u\}.$$

According to [16, Theorem 2A (c)], U^∞ can also be defined algebraically as

$$U^\infty = \{u \in X | U + u \subset U\}. \tag{3}$$

A proper lower semicontinuous (l.s.c.) convex function g on X is an everywhere-defined function with values in $(-\infty, +\infty]$, not identically $+\infty$, such that epi g is a closed convex set in $X \times \mathbb{R}$, where epi g denotes the epigraph of g. Its *effective* domain is the nonempty convex set

$$\operatorname{dom} g = \{x \in X| \ g(x) < +\infty\}.$$

For a proper l.s.c. convex function g, we use $\partial g(x)$ to denote the subdifferential of g at x ($\in \operatorname{dom} g$).

For a proper l.s.c. convex function g, we use the recession cone of the epigraph of g to define the recession function of g, denoted by g^∞; that is,

$$\operatorname{epi}(g^\infty) = (\operatorname{epi} g)^\infty.$$

2. The main theorem. We begin with a proposition on the recession cone of the intersection of convex sets in X. The proof of the proposition is similar to that of [15, Corollary 8.3.3] when $X = \mathbb{R}^n$. For the completeness, we give a proof.

PROPOSITION 2.1. *Suppose that $\{U_i | i \in I\}$, where I is an arbitrary index set, is a family of nonempty closed convex sets in X and $U = \cap_{i\in I} U_i$ is nonempty. Then $U^\infty = \cap_{i\in I} {U_i}^\infty$.*

Proof. The recession cone $U^\infty \subset \cap_{i\in I} {U_i}^\infty$ follows from the definition of recession cone. On the other hand, let $u \in U_i^\infty$ for all $i \in I$. Then, for any $x \in U$, by (3), $u + x \in U_i$ for all $i \in I$. It follows that $u + x \in U$, and $u \in U^\infty$. □

For a proper l.s.c. convex function g, we have the following proposition to compute g^∞.

PROPOSITION 2.2 (see [16, Corollary 3C]). *If g is a proper l.s.c. convex function, then g^∞ can be determined from any of the following formulas:*

(a) $g^\infty(u) = \sup_{x\in(\operatorname{dom} g)} \{g(x + u) - g(x)\}$,

(b) $g^\infty(u) = \sup_{\lambda>0}[g(x + \lambda u) - g(x)]/\lambda$ *for any $x \in dom\, g$,*

(c) $g^\infty(u) = \sup_{v\in \operatorname{cl}(\operatorname{dom} g^*)} \langle v, u\rangle$, *where g^* is the convex conjugate function of g.*

From Proposition 2.2(c), g^∞ is the support function of the nonempty closed convex set cl(domg^*). Thus, g^∞ is a proper l.s.c. convex function. For more about recession cones and recession functions, see [15, 16].

Using Propositions 2.1 and 2.2(b), one can easily verify that the recession cone S^∞ of S given by (1) is the set of all solutions to the following convex inequality system:

$$u \in C^\infty, \quad f_i^\infty(u) \leq 0 \text{ for } i = 1, 2, \ldots, m. \tag{4}$$

If system (1) satisfies Robinson's boundedness condition [14], then $S^\infty = \{0\}$. For S being unbounded, asymptotic constraint qualification assumptions [12, 1] are not applicable to the infinite-dimensional case. In this paper, we introduce the following Slater constraint qualification regarding the convex system (4).

Assumption 1. There exist a unit vector $\hat{u} \in C^\infty$ and a constant $\tau > 0$ such that $f_i^\infty(\hat{u}) \leq -\tau^{-1}$ for $i = 1, 2, \ldots, m$.

Remark 2.1. Under Assumption 1, S must be an unbounded set.

Now we are in a position to state the main theorem of this paper.

THEOREM 2.3. *Suppose that S is the set of all solutions to* (1) *and Assumption* 1 *holds. Then for any p with $1 \leq p \leq \infty$,*

$$d(z, S) \leq \tau \|[F(z)]_+\|_p \quad \textit{for all } z \in C \subset X,$$

where τ is given by Assumption 1.

Proof. Let $f(x) = \max_{1 \leq i \leq m}\{f_i(x)\}$. Then f is a continuous convex function on X, and $S = \{x \in C | f(x) \leq 0\}$. Since X is reflexive and $\|\cdot\|$ is weak* l.s.c. (see [3]), for any $z \in C$ but not in S, there exists an $\bar{x}$ in S such that $\|\bar{x} - z\| = d(z, S)$. Thus,

$$0 \in \partial\|\bar{x} - z\| + N_S(\bar{x}).$$

That is, there exist $v_1 \in \partial\|\bar{x} - z\|$ and $v_2 \in N_S(\bar{x})$ with

$$0 = v_1 + v_2.$$

It follows from [10, p. 46] that $\|\bar{x} - z\| = \langle -v_1, z - \bar{x}\rangle$ and $\|v_1\|_* = 1$.

It is easy to see that $f(\bar{x}) = 0$; otherwise, we could find a "better" point in S by the convexity of S and the continuity of f. An immediate consequence of Assumption 1 is that there is an $x_0 \in C$ such that $f_i(x_0) < 0$ for $i = 1, 2, ..., m$. Therefore $f(x_0) < 0$ and $0 \notin \partial f(\bar{x})$. By the continuity of f, we have that $\delta_{\{x|f(x) \leq 0\}}(\cdot)$ is continuous at x_0. By applying [10, Theorem 0.3.3, p. 47] to $\delta_S(\cdot) = \delta_{\{x|f(x)\leq 0\}}(\cdot) + \delta_C(\cdot)$, we get

$$N_S(\bar{x}) = \cup_{\lambda \geq 0} \lambda \partial f(\bar{x}) + N_C(\bar{x}).$$

Hence,

$$-v_1 = v_2 = \lambda u_1 + u_2, \tag{5}$$

where $\lambda \geq 0$, $u_1 \in \partial f(\bar{x})$, and $u_2 \in N_C(\bar{x})$. Since $\partial f(\bar{x})$=co$\{\partial f_i(\bar{x}) | i \in I(\bar{x})\}$,

$$u_1 = \sum_{i \in I(\bar{x})} \mu_i v_i \quad \text{with } \mu_i \geq 0, \textstyle\sum_{i \in I(\bar{x})} \mu_i = 1, \text{ and } v_i \in \partial f_i(\bar{x}) \quad \text{for } i \in I(\bar{x}), \tag{6}$$

where $I(\bar{x})$ denotes the set of indices i for which $f_i(\bar{x}) = f(\bar{x})$. By (3), $\bar{x} + \hat{u} \in C$. Thus $\langle u_2, \hat{u}\rangle = \langle u_2, \bar{x} + \hat{u} - \bar{x}\rangle \le 0$. Since $1 = \| - v_1\|_* \ge \langle -v_1, -\hat{u}\rangle$, it follows from (5) and (6) that

$$1 \ge \lambda\langle u_1, -\hat{u}\rangle + \langle u_2, -\hat{u}\rangle \ge \lambda\langle u_1, -\hat{u}\rangle \ge \lambda \left(\sum_{i\in I(\bar{x})} \mu_i \langle v_i, -\hat{u}\rangle \right). \tag{7}$$

By Proposition 2.2(a), for each i with $1 \le i \le m$, one has

$$\begin{aligned} -\tau^{-1} \ge f_i^{\infty}(\hat{u}) &= \sup_{x\in \mathrm{dom} f_i} \{f_i(x+\hat{u}) - f_i(x)\} \\ &\ge \sup_{x\in X} \sup_{v\in\partial f_i(x)} \langle v, \hat{u}\rangle \qquad \text{by the convexity of } f_i, \\ &\ge \sup_{v\in\partial f_i(\bar{x})} \langle v, \hat{u}\rangle \ge \langle v_i, \hat{u}\rangle, \end{aligned}$$

which, combining with (7), yields $\lambda \le \tau$. Therefore,

$$\begin{aligned} \|\bar{x} - z\| &= \langle -v_1, z - \bar{x}\rangle \\ &= \langle \lambda u_1 + u_2, z - \bar{x}\rangle \\ &\le \lambda\langle u_1, z - \bar{x}\rangle \quad \text{since } \langle u_2, z - \bar{x}\rangle \le 0, \\ &= \lambda \sum_{i\in I(\bar{x})} \mu_i \langle v_i, z - \bar{x}\rangle \\ &\le \lambda \sum_{i\in I(\bar{x})} \mu_i (f_i(z) - f_i(\bar{x})) \quad \text{by the convexity of } f_i, \\ &\le \lambda \sum_{i\in I(\bar{x})} \mu_i \max\{f_i(z), 0\} \quad \text{since } f_i(\bar{x}) = 0 \text{ for all } i \in I(\bar{x}), \\ &\le \tau \left(\sum_{i\in I(\bar{x})} \mu_i^q\right)^{1/q} \left(\sum_{i\in I(\bar{x})} (\max\{f_i(z), 0\})^p\right)^{1/p} \qquad \text{where } 1/p + 1/q = 1, \\ &\le \tau\|[F(z)]_+\|_p. \end{aligned}$$

This completes the proof. □

Remark 2.2. When $X = \mathbb{R}^n$, Assumption 1 implies the asymptotic constraint qualification assumptions introduced in [1, 12] for S being unbounded. Thus, Assumption 1 can be viewed as a constraint qualification condition when X is an infinite-dimensional reflexive Banach space and S is unbounded. One important feature of Assumption 1 is that this assumption provides a verifiable condition such that (2) holds with a *computable* constant τ.

The following proposition will be needed for Corollary 2.5.

PROPOSITION 2.4. *Let Y be a compact metric space. Suppose that g is continuous on $X \times Y$ and $g(\cdot, y)$ is convex for each $y \in Y$. If $f(x) = \sup_{y\in Y} g(x, y)$, then f is continuous convex on X.*

Proof. It is evident that f is convex. To show that f is continuous, let $x_n \to \bar{x}$. For each x_n, there is a $y_n \in Y$ such that $f(x_n) = g(x_n, y_n)$ since Y is compact. Without loss of generality, by the compactness of Y, we can assume that $\{y_n\}$ converges to some $\bar{y} \in Y$. Since $g(x_n, y_n) \ge g(x_n, y)$ for all $y \in Y$, $g(\bar{x}, \bar{y}) = \sup_{y\in Y} g(\bar{x}, y)$ follows from the continuity of g at $(\bar{x},\bar{y})$. Hence,

$$\begin{aligned}\lim_{n\to\infty} f(x_n) &= \lim_{n\to\infty} g(x_n,y_n) = g(\bar{x},\bar{y})\\ &= \sup_{y\in Y} g(\bar{x},y) = f(\bar{x}).\end{aligned}$$

The result follows. $\square$

In view of Propositions 2.1, 2.4, and Theorem 2.3, we have the following corollary.

COROLLARY 2.5. *Suppose that f is given in Proposition* 2.4. *For each $y \in Y$, let $g^\infty(\cdot,y)$ be the recession function of $g(\cdot,y)$. Let $C \subset X$ be a nonempty closed convex set. Suppose that $\tilde{S} = \{x \in C | f(x) \le 0\}$ is nonempty and there exist a unit vector $\hat{u} \in C^\infty$ and a constant $\tau > 0$ such that $g^\infty(\hat{u},y) \le -\tau^{-1}$ for all $y \in Y$. Then,*

$$d(z,\tilde{S}) \le \tau[f(x)]_+ \quad \text{for all } z \in C \subset X.$$

3. Examples of recession functions. To apply Theorem 2.3, one needs to know recession functions of given continuous functions f_i. In this section, we give several examples of recession functions for some important convex functions.

Example 1. Suppose that X is a Hilbert space and L is a self-conjugate continuous linear operator [2, p. 23] from X to X satisfying (a) $\langle Lx, x\rangle \ge 0$ for all $x \in X$ (positive semidefinite), and (b) $\operatorname{Im} L$ is closed in X, where $\operatorname{Im} L$ denotes the image of L. Then $f(x) = 1/2\langle Lx,x\rangle + \langle b,x\rangle + c$ is a continuous convex function on X, where $b \in X$ and c is a constant. Since $\operatorname{Im} L = (\operatorname{Ker} L)^\perp$, where $\operatorname{Ker} L$ denotes the kernel of L, $\operatorname{dom} f^* = (\operatorname{Ker} L)^\perp + b$ [2, Proposition 3.7]. It follows from Proposition 2.2(c) that

$$f^\infty(u) = \langle b,u\rangle + \delta_{Ker\,L}(u).$$

Example 2 (see [15, p. 68]). Suppose that $X = \mathbb{R}^n$. Let $f(x) = \log(e^{x_1} + e^{x_2} + \cdots + e^{x_n})$. Then

$$f^\infty(u) = \max\{u_1,\ldots,u_n\}.$$

Example 3 (see [15, Theorem 9.3.]). Suppose that $X = \mathbb{R}^n$. Let $f_1,\ldots,f_m$ be continuous convex functions on $\mathbb{R}^n$. Then

$$(f_1 + \cdots + f_m)^\infty = f_1^\infty + \cdots + f_m^\infty.$$

Example 4 (see [15, Theorem 9.4]). Suppose that $X = \mathbb{R}^n$. For an arbitrary index set I, let $f(x) = \sup_{i\in I}\{f_i(x)\}$, where each f_i is a continuous convex function on $\mathbb{R}^n$. Then

$$f^\infty(u) = \sup_{i\in I}\{f_i^\infty(u)\}.$$

Acknowledgments. The author thanks Professor R. F. Wheeler for several stimulating discussions that led to an improved presentation of Theorem 2.3 and two anonymous referees for their useful comments.

REFERENCES

[1] A. A. AUSLENDER AND J.-P. CROUZEIX, *Global regularity theorems,* Math. Oper. Res., 13 (1988), pp. 243–253.

[2] J. P. AUBIN, *Optima and Equilibria,* Springer-Verlag, Berlin, New York, 1993.

[3] M. M. DAY, *Normed Linear Spaces,* Springer-Verlag, Berlin, New York, 1973.

[4] M. S. GOWDA, *An analysis of zero set and global error bound properties of a piecewise affine function via its recession function*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 594–609.

[5] A. J. HOFFMAN, *On approximate solutions of systems of linear inequalities,* J. Res. Nat. Bur. Standards, 49 (1952), pp. 263–265.

[6] H. HU AND Q. WANG, *On approximate solutions of infinite systems of linear inequalities,* Linear Algebra Appl., 114/115 (1989), pp. 429–438.

[7] W. LI, *Error bounds for piecewise convex quadratic programs and applications,* SIAM J. Control Optim., 33 (1995), pp. 1510–1529.

[8] X.-D. LUO AND Z.-Q. LUO, *Extension of Hoffman's error bound to polynomial systems,* SIAM J. Optim., 4 (1994), pp. 383–392.

[9] Z. Q. LUO AND J. S. PANG, *Error bounds for analytic systems and their applications,* Math. Programming 67 (1995), pp. 1–28.

[10] A. D. IOFFE AND V. M. TIHOMIROV, *Theory of Extremal Problems,* Nauka, Moscow, 1974 (in Russian).

[11] A. D. IOFFE, *Regular points of Lipschitz functions,* Trans. Amer. Math. Soc., 251 (1979), pp. 61–69.

[12] O. L. MANGASARIAN, *A condition number for differentiable convex inequalities,* Math. Oper. Res., 10 (1985), pp. 175–179.

[13] R. MATHIAS AND J. S. PANG, *Error bounds for the linear complementarity problem with a P-matrix,* Linear Algebra Appl., 132 (1990), pp. 123–136.

[14] S. M. ROBINSON, *An application of error bounds for convex programming in a linear space,* SIAM J. Control, 13 (1975), pp. 271–273.

[15] R. T. ROCKAFELLAR, *Convex Analysis,* Princeton University Press, Princeton, NJ, 1970.

[16] R. T. ROCKAFELLAR, *Level sets and continuity of conjugate convex functions,* Trans. Amer. Math. Soc., 123 (1966), pp. 46–63.

[17] T. WANG AND J. S. PANG, *Global error bounds for convex quadratic inequality systems*, Optimization, 31 (1994), pp. 1–12.

[18] J. V. BURKE AND P. TSENG, *A unified analysis of Hoffman's bound via Fenchel duality*, SIAM J. Optim., 6 (1996), pp. 265–282.

SIAM J. OPTIM.
Vol. 7, No. 1, pp. 280–294, February 1997

IMPLEMENTATION OF A VARIANCE REDUCTION-BASED LOWER BOUND IN A BRANCH-AND-BOUND ALGORITHM FOR THE QUADRATIC ASSIGNMENT PROBLEM*

P. M. PARDALOS[†], K. G. RAMAKRISHNAN[‡], M. G. C. RESENDE[§], AND Y. LI[¶]

Abstract. The efficient implementation of a branch-and-bound algorithm for the quadratic assignment problem (QAP), incorporating the lower bound based on variance reduction of Li, Pardalos, Ramakrishnan, and Resende (1994), is presented. A new data structure for efficient implementation of branch-and-bound algorithms for the QAP is introduced. Computational experiments with the branch-and-bound algorithm on different classes of QAP test problems are reported. The branch-and-bound algorithm using the new lower bounds is compared with the same algorithm utilizing the commonly applied Gilmore–Lawler lower bound. Both implementations use a greedy randomized adaptive search procedure for obtaining initial upper bounds. The algorithms report all optimal permutations. Optimal solutions for previously unsolved instances from the literature, of dimensions $n = 16$ and $n = 20$, have been found with the new algorithm. In addition, the new algorithm has been tested on a class of large data variance problems, requiring the examination of much fewer nodes of the branch-and-bound tree than the same algorithm using the Gilmore–Lawler lower bound.

Key words. combinatorial optimization, quadratic assignment problem, branch-and-bound, GRASP, computer implementation, data structures, hashing function, hash table, lower bound, test problems

AMS subject classifications. 90B80, 90C20, 90C35, 90C27, 65H20, 65K05

PII. S1052623494273393

1. Introduction.

The quadratic assignment problem (QAP) can be stated as

$$\min_{p \in \Pi} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{p(i)p(j)},$$

where Π is the set of all permutations of $\{1, 2, \ldots, n\}$, $A = (a_{ij}) \in \mathcal{R}^{n \times n}$, and $B = (b_{ij}) \in \mathcal{R}^{n \times n}$. The QAP was first proposed by Koopmans and Beckmann in 1957 as a mathematical model for a set of indivisible economical activities [29]. A typical example of the QAP is the facility location problem, in which a set of n facilities is to be assigned to an equal number of locations. Between each pair of facilities there is a given amount of flow, contributing a cost equal to the product of the flow and the distance between the locations to which the facilities are assigned. Applications of the QAP are abundant and can be found in [6, 20, 25, 30, 33, 38, 43]. Many classical combinatorial optimization problems, such as the traveling salesman problem and the graph partitioning problem, are special cases of the QAP.

A wide range of heuristics have been applied to find approximate solutions to the QAP [10, 19, 33, 37, 39, 48, 49, 50]. Exact solution approaches have been limited to instances of dimension $n \leq 15$ and are based mostly on branch-and-bound. General QAPs of dimension $n > 15$ are considered difficult large-scale problems.

[*] Received by the editors August 17, 1994; accepted for publication (in revised form) September 12, 1995.

http://www.siam.org/journals/siopt/7-1/27339.html

[†] Center for Applied Optimization and Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611 (pardalos@ufl.edu).

[‡] Bell Laboratories, Murray Hill, NJ 07974 (kgr@research.bell-labs.com).

[§] AT&T Research, Murray Hill, NJ 07974 (mgcr@research.att.com).

[¶] Sanwa Financial Products Company, 55 East 52nd Street, 26th Floor, New York, NY 10055 (yong@sanwaBGK.com).

Branch-and-bound is an enumerative technique for solving combinatorial optimization problems. Branching usually refers to a successive partitioning of the feasible domain while bounding refers to the determination of lower and upper bounds for the global optimal solution. Recently, Li, Pardalos, Ramakrishnan, and Resende [32] proposed new lower bounds, based on reduction techniques, for the QAP. In this paper, we show how to efficiently implement these bounds in a branch-and-bound algorithm for the QAP. We report on computational experiments with a branch-and-bound algorithm using the new bounds, as well as the Gilmore–Lawler lower bounds, on a large set of test problems.

Before we conclude the introduction, let us define some notation and state some assumptions used in this paper. Matrix A is referred to as the *flow matrix*, while B is the *distance matrix*. For convenience of discussion, an instance of the QAP with flow and distance matrices A and B is denoted as QAP(A, B). Without loss of generality, it is assumed that the entries of matrices A and B are nonnegative [43]. We further assume that the diagonal entries of matrices A and B are zero.

The paper is organized as follows. In section 2 we discuss issues related to branch-and-bound algorithms. A specialized branch-and-bound approach for the QAP is given in section 3. In section 4, an efficient implementation of the branch-and-bound algorithm is considered. Computational results are summarized in section 5 and concluding remarks are made in section 6.

2. Branch-and-bound algorithms.

The underlying idea of a branch-and-bound algorithm is to partition a given initial problem into a number of intermediate partial problems of smaller sizes. Every subproblem is characterized by the inclusion of one or more constraints. The decomposition is repeatedly applied to the generated subproblems until each unexamined subproblem is decomposed, solved, or shown not to lead to an optimal solution to the original problem. Branch-and-bound is essentially a variant, or refinement, of backtracking that can take advantage of information about the optimality of partial solutions to avoid considering solutions that cannot be optimal—hence, to reduce the search space significantly.

The notation of Ibaraki [26] is employed to formally define a branch-and-bound algorithm that will be needed in the sequel. Let P_0 denote an optimization problem and f denote the objective function to be minimized. The decomposition process applied to P_0 is represented by a rooted tree $\mathcal{R} = (\mathcal{P}, \mathcal{E})$, where $\mathcal{P}$ is a set of nodes and $\mathcal{E}$ is a set of arcs. The root of $\mathcal{R}$, denoted P_0, corresponds to the given problem P_0, and other nodes P_i correspond to partial problems P_i. The arc $(P_i, P_j) \in \mathcal{E}$ if and only if P_j is generated from P_i by a decomposition. The set of terminal nodes of $\mathcal{R}$, denoted $\mathcal{T}$, are those partial problems that are solved without further decomposition. The level of $P_i \in \mathcal{R}$, denoted $L(P_i)$, is the length of the path from P_0 to P_i in $\mathcal{R}$. P_0 has level 0. $\mathcal{R}$ is assumed to be a finite graph.

A branch-and-bound algorithm attempts to solve P_0 by examining only a small portion of $\mathcal{R}$. This is accomplished by no longer proceeding along the branches rooted at those nodes P_i that are either solved or found by test not to yield an optimal solution of P_0 (i.e., $F(P_i) > F(P_0)$, where $F(P) = \min_{x \in P} f(x)$). A lower bound function g is calculated for each subproblem as it is created, to help eliminate unnecessary search. This lower bound function represents a smallest possible cost of a solution to that subproblem, given the subproblem's constraints. Its values satisfy the following conditions:

- $g(P_i) \leq F(P_i)$ for $P_i \in \mathcal{P}$,
- $g(P_i) = F(P_i)$ for $P_i \in \mathcal{T}$,

- $g(P_j) \geq g(P_i)$ if P_j is a descendant of P_i.

A typical branch-and-bound algorithm consists of four major procedures: selection, branching, elimination, and termination test.

- *Selection.* At any step during the execution of the algorithm, there exists a set $\mathcal{A}$ of problems that have been generated but not yet examined. The selection procedure selects a single subproblem from the set $\mathcal{A}$, based on a selection heuristic function h. The set $\mathcal{A}$ is maintained in an ordered list by increasing values of h. The following three heuristic searching strategies are commonly used:
  - best-bound search: $h \equiv g$,
  - depth-first search: $h \equiv -L$, or
  - breadth-first search: $h \equiv L$, where L is the node level in $\mathcal{R}$.
- *Branching.* A *branching rule* related to a given problem is used to generate new smaller subproblems from the one selected by the selection procedure. Lower bounds for the newly generated subproblems are calculated accordingly.
- *Elimination.* A newly created subproblem is deleted if its lower bound is greater than or equal to that of the incumbent (the best feasible solution discovered up to that point of the search).
- *Termination Test.* In some cases, with restrictive constraints, it may be possible to define a number of auxiliary rules that help identify infeasible partial solutions.

In the selection procedure, the best-bound and depth-first search strategies are used in most situations. Best-bound search minimizes the number of partial problems decomposed prior to termination. However, it tends to consume an amount of memory that is an exponential function of the problem size. On the other hand, depth-first search consumes an amount of space that is only a linear function of the problem size, and its implementation is relatively easy. The branch-and-bound algorithm terminates when the list of active subproblems is empty, and the incumbent is the optimal solution of the original problem.

3. Branch-and-bound algorithms for the QAP. Three classes of methods have been used to find globally optimal solutions to the QAP. These methods include cutting plane techniques, branch-and-bound methods, and dynamic programming.

Exact cutting plane algorithms have not succeeded to generate optimal solutions for problems with dimension as small as $n = 10$ [4, 28]. They have, however, been successfully applied to obtain good suboptimal permutations [7].

Branch-and-bound algorithms have been the most successful methods for proving optimality of QAPs. Lower bounds are key to the computational performance of these branch-and-bound algorithms. Lower bounds for the QAP can be categorized into three groups. The first category includes the classical Gilmore–Lawler bound (GLB) [22, 31] and related bounds. The second category consists of eigenvalue-based bounds [18, 24, 23, 44]. The rest of the bounds are mostly based on reformulations of the QAP and generally involve solving a number of linear assignment problems (e.g., [3, 11, 13, 16, 21]). A new class of lower bounds that belongs to the first category was proposed by Li, Pardalos, Ramakrishnan, and Resende [32] and is described in section 3.1.

One of the first exact branch-and-bound algorithms for the QAP is described in [16], but no computational results are reported. In the book by Burkard and Derigs [8], the Fortran source code for solving exactly QAPs with a branch-and-bound algorithm is listed. Roucairol [47] proposed and implemented sequential and parallel branch-and-bound algorithms on a Cray X-MP/48 (four processors) and solved the Nugent-12 ($n = 12$) test problem [40] in about 5 minutes but was unable to solve the Nugent-15

($n = 15$) instance, due to insufficient memory. Pardalos and Crouse [41] developed another parallel implementation of a single assignment branch-and-bound algorithm on an IBM 3090/400E (four processors). That implementation solved the Nugent-12 problem in half a minute and partially solved (examined 95% of the nodes of the branch-and-bound tree) in about 30 minutes. More recently, Mautor and Roucairol [36, 35] considered new approaches to reduce the size of the search tree in an exact branch-and-bound algorithm and report computational results for some problems of size up to $n = 20$. The long standing problem Nugent-20 ($n = 20$) was reportedly solved in 1994 by Clausen [14]. In addition, QAPs of size up to $n = 30$, in which the flow matrix is the weighted adjacency matrix of a tree, have been solved exactly, using dynamic programming approaches [12]. Other exact approaches are described in [43].

In this paper, we discuss an exact branch-and-bound algorithm that incorporates the new lower bound using efficient data structure techniques and the GRASP heuristic [17] to find the initial upper bound.

3.1. A new class of lower bounds. A class of lower bounds based on optimal reduction schemes for the QAP was proposed in [32]. For a given QAP(A, B), consider a partition of A into the two matrices $A_1 = (a_{ij}^{(1)})$ and $A_2 = (a_{ij}^{(2)})$ such that $A = A_1 + A_2$ and a partition of B into two matrices $B_1 = (b_{ij}^{(1)})$ and $B_2 = (b_{ij}^{(2)})$ such that $B = B_1 + B_2$. For each pair $\{i, j\}$, $i, j = 1, \ldots, n$, let

$$l_{ij} = \min_{p \in \pi,\ p(i)=p(j)} \left\{ \sum_{k=1}^{n} a_{ik}^{(1)} b_{jp(k)}^{(1)} + \sum_{k=1}^{n} a_{ki}^{(2)} b_{p(k)j} + \sum_{k=1}^{n} a_{ki} b_{p(k)j}^{(2)} - \sum_{k=1}^{n} a_{ki}^{(2)} b_{p(k)j}^{(2)} \right\}, \tag{3.1}$$

where π is the set of all permutations of $\{1, 2, \ldots, n\}$. Let $L = (l_{ij})$ be an $n \times n$ matrix. The following theorem defines a new lower bound [32, Theorem 4.1].

THEOREM 3.1. *Let the matrix L be defined as above. The solution of the linear assignment problem with cost matrix L is a lower bound for the corresponding QAP.*

The classical Gilmore–Lawler bound [22, 31] (denoted here by GLB(A, B)) is a special case in which neither matrix A nor B are partitioned. Different ways of partitioning the matrices A and B (also referred to as *reduction*) yield different lower bounds. The common reduction techniques used in the literature choose A_2 and B_2 with constant column sums (often called constant columns). We refer to such techniques as constant column reductions.

Let $M = (m_{ij})$ be a matrix in $R^{n \times n}$. We treat a row vector m_i, $1 \leq i \leq n$, of M as a $1 \times n$ matrix and a column vector m_j^T, $1 \leq j \leq n$, as an $n \times 1$ matrix. For convenience of discussion, we use the following notation for average $\gamma(M)$, variance $V(M)$, and total variance $T(M, \lambda)$ of M:

$$\gamma(M) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} m_{ij},$$

$$V(M) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\gamma(M) - m_{ij})^2,$$

$$T(M, \lambda) = \lambda \sum_{i=1}^{n} V(m_i) + (1 - \lambda) V(M) \text{ for } 0 \leq \lambda \leq 1.$$

In our reduction scheme, we considered the partition $A = A_1 + A_2$, where $A_1 = A + \Delta$ and $A_2 = -\Delta$, such that the variances of A_1 and A_2, the sum of variances of the rows of A_1, and the sum of variances of the rows of A_2 are minimized. This minimization problem has been formulated as

$$\min\theta\, T(A+\Delta,\lambda) + (1-\theta)\, T(-\Delta^\top,\lambda) \tag{3.2}$$

$$\text{such that}\Delta \in R^{n\times n}, \tag{3.3}$$

where λ and θ $(0 \le \lambda, \theta \le 1)$ are input parameters.

Motivated by the observation that for QAP(A,B) the tightness of GLB is inversely proportional to the variances of A and B, the following reduction schemes were proposed in [32]:

- $\mathcal{R}$-1: $a_{ij}^{(1)} = a_{ij} - \theta(a_{nn} - a_{ij})$ and $a_{ij}^{(2)} = \theta(a_{nn} - a_{ij}),\ i,j = 1,\ldots,n,$
- $\mathcal{R}$-2: $a_{ij}^{(1)} = a_{ij} - \theta(\gamma(a_n^T) - \gamma(a_j^T))$ and $a_{ij}^{(2)} = \theta(\gamma(a_n^T) - \gamma(a_j^T)),\ i,j = 1,\ldots,n.$

Note that both reduction schemes are independent of the value of λ (c.f. [32]). One new lower bound proposed in [32] is to use reduction scheme $\mathcal{R}$-1. This lower bound is denoted by LB1(θ). The other new lower bound proposed is to use reduction scheme $\mathcal{R}$-2. This lower bound is denoted LB2(θ). Both new lower bounds depend on the parameter θ. Note that LB1(0.0) = GLB(A,B) and LB1(1.0) = GLB(A^T,B^T).

In [32], it was observed empirically that $\theta = 0.5$ and $\theta = 1.0$ are good choices for LB1(θ) and LB2(θ), respectively. Furthermore, in that study the bound LB2(1.0) was slightly tighter than LB1(0.5). Consequently, in this implementation we use LB2(1.0).

For LB2(1.0) the computation is simpler. The variance minimization problem (3.2–3.3) becomes

$$\min V(A+\Delta) \tag{3.4}$$

$$\text{such that}\Delta \in R^{n\times n}. \tag{3.5}$$

The solution of (3.4–3.5) is simply

$$\delta_{ij} = \gamma(a_j^\top) - \gamma(a_n^\top) \text{ for } i,j = 1,\ldots,n,$$

which is the constant column partitioning scheme.

The new lower bounds can be computed efficiently. Computing matrix Δ to partition matrices A and B takes only $O(n^2)$ time (c.f. [32]). By presorting the rows of the flow and distance matrices A and B, one can compute l_{ij} $(i,j = 1,\ldots,n)$ in $O(n^3)$. Hence the total running time is $O(n^3)$, which is the same as that for computing GLB. Furthermore, the constant factor is small. Later in this paper, we show how to efficiently incorporate these bounds into a branch-and-bound algorithm for the QAP.

Recently, Jansen [27] has derived an analytical closed form solution to (3.2–3.3). That solution is given by

$$\begin{aligned}\delta_{ij}(\theta) = {} & \theta\lambda\frac{1-\theta}{1-\theta\lambda}\gamma(a_i) + \frac{\theta(1-\lambda)+\theta\lambda^2(1-\theta)-\theta^2\lambda^2(1-\theta)}{(1-\theta\lambda)(1-\lambda+\theta\lambda)}\gamma(A) \\ & -\frac{\lambda\theta(1-\theta)}{1-\lambda+\theta\lambda}\gamma(a_j^\top) - \theta a_{ij}.\end{aligned}$$

Note that, as $\theta \to 1$, $\delta_{ij}(\theta) \to \gamma(A) - a_{ij}$, which is the constant column reduction partitioning scheme. Experimentally, we have observed that the constant column reduction partitioning scheme is more effective and is easier to implement than the closed form solution.

3.2. The new branch-and-bound algorithm for QAP. The exact algorithm presented in this section uses the branch-and-bound technique described in section 2. The terms *solution* and *permutation* are used interchangeably in the discussion. The algorithm consists of three steps.

In the first step, an initial upper bound is computed and an initial branch-and-bound search tree is set up. Our branch-and-bound tree is a forest of n binary trees (not necessarily a complete binary tree). Each node of the tree has a left and a right child. For the purpose of describing the branching process, let us denote (at any node of the branch-and-bound tree) S_A to be the set of assignments (of facilities to sites) that are always fixed at any node of the subtree rooted at this node (including this node) and S_E to be the set of excluded assignments, i.e., the assignments that are forever excluded in any node of the subtree rooted at the current node (including the current node). The sets S_A and S_E completely describe a node of the branch-and-bound tree. Let S^l_A, S^l_E and S^r_A, S^r_E be the corresponding sets for the left and right children of the current node. Currently unexplored nodes of the branch-and-bound tree are organized as a heap with a key that is equal to the lower bound on the solution to the original QAP obtainable by any node in the subtree rooted at this node. The heap is organized in maximum order; i.e., the node with the largest lower bound is first.

The initial best known upper bound is computed by the GRASP heuristic described in [33, 45]. Let $P = (p_1, p_2, \dots, p_n)$ denote the initial solution found by the GRASP heuristic; i.e., p_i is the site assigned to facility i in this solution. We use the notation $\{i \to s\}$ to indicate that facility i is assigned to site s. The initial search tree consists of a forest of n isolated nodes, where for $i = 1, \dots, n$, S_A of node i is $\{1 \to p_i\}$, $S_E = \emptyset$, and all n nodes have a key of 0.

In the second step, the four procedures of the branch-and-bound algorithm, as described in section 2, are used as follows:

- *Selection.* The selection procedure simply chooses the node at the root of the heap, i.e., the node with the maximum key.
- *Branching.* The branching procedure creates two children, the left and the right children, as follows: let i be the smallest index of a facility that is not in any assignment of S_A and s be the index of a site that is not in any assignment of S_A, such that the assignment $\{i \to s\}$ is not in S_E. Then,

$$\begin{aligned} &S^l_A = S_A \cup \{i \to s\},\\ &S^l_E = \emptyset,\\ &S^r_A = S_A,\\ &S^r_E = S_E \cup \{i \to s\}, \end{aligned}$$

and the key of the right child is the same as the key of the current node and the key of the left child is the newly computed lower bound.

- *Elimination.* The elimination procedure compares the newly computed lower bound of the left child to the incumbent and deletes the left child if its key is greater than the incumbent, thus pruning the entire subtree rooted at the left child.
- *Termination test.* The algorithm stops if and only if the heap is empty.

In the final step, a best permutation found is taken as the global optimal permutation.

The binary search tree has many interesting properties. First, observe that S^l_E is set to $\emptyset$. This is a consequence of the relationship between S_A and S_E at every node of the branch-and-bound tree (as enumerated below). The S^l_A implicitly captures the excluded assignment in S^l_E and so S^l_E can be set to a $\emptyset$. Other interesting properties are listed below. All these properties enable us to derive the result on the maximum depth of the branch-and-bound tree and the maximum number of nodes in the branch-and-bound tree.

We denote by L the level of the binary tree, counting the root of the branch-and-bound tree as level 1. The following properties hold for the branch-and-bound tree:

- For any node of the branch-and-bound tree, if $S_E \neq \emptyset$ then all assignments in S_E have exactly one facility index, and that index is one larger than the largest facility index in S_A.
- All site indices in $S_A \cup S_E$ are distinct.
- For any node, $|S_A| + |S_E| \leq n$.
- A node i is a right-ancestor of node j if node i is in the path from node j to the root of the branch-and-bound tree and i is the right child of its parent. This definition considers node i to be a right-ancestor of itself ($i = j$ in the definition) if i is a right child. Let r_i be the number of right-ancestors for node i. At any level L of the branch-and-bound tree and for any node i the following relation holds:

$$|S_A| + r_i = L.$$

- For any node i of the branch-and-bound tree, we have that $r_i \leq n^2$.
- The maximum depth of the branch-and-bound tree is n^2. This property gives a bound of at most 2^{n^2} branch-and-bound nodes.

4. Efficient computation of the new lower bound. To implement the new lower bound in the above branch-and-bound scheme, we exploit some properties of the bound. At each node of the branch-and-bound tree, the matrix L must be computed and the corresponding linear assignment problem solved.

At a particular node of the branch-and-bound tree, let n' denote the number of facilities already assigned to sites. Let q be the corresponding partial assignment vector. Let S_A and S_B denote the index sets of already assigned facilities and sites, respectively (corresponding to the partial assignment q). Note that $|S_A| = |S_B| = n'$. At the current node, a QAP of reduced size $n - n'$ remains to be solved. Theorem 3.1 can be used to obtain a lower bound for the reduced problem.

Let $n'' = n - n'$ be the size of the reduced problem, and let $\bar{A}$ and $\bar{B}$ be the corresponding flow and distance matrices for the reduced problem. Recall from Theorem 3.1 that for $i, j = 1, \ldots, n''$, the element l_{ij} of L is given by

$$\begin{aligned} l_{ij} = \min_{p\in\Pi, p(i)=j} & \sum_{k=1}^{n''} \bar{a}^{(1)}_{ik}\bar{b}^{(1)}_{jp(k)} + \sum_{k=1}^{n''} \bar{a}^{(2)}_{ki}\bar{b}_{p(k)j} \\ & + \sum_{k=1}^{n''} \bar{a}_{ki}\bar{b}^{(2)}_{p(k)j} - \sum_{k=1}^{n''} \bar{a}^{(2)}_{ki}\bar{b}^{(2)}_{p(k)j}. \end{aligned} \tag{4.1}$$

Equivalently, l_{ij} can be written as

$$\begin{aligned} l_{ij} = {} & \bar{a}^{(1)}_{ii}\bar{b}^{(1)}_{jj} + \bar{a}^{(2)}_{ii}\bar{b}_{jj} + \bar{a}_{ii}\bar{b}^{(2)}_{jj} - \bar{a}^{(2)}_{ii}\bar{b}^{(2)}_{jj} \\ & + \min_{p\in\Pi} \left\{ \sum_{k=1,k\neq i}^{n''} \bar{a}^{(1)}_{ik}\bar{b}^{(1)}_{jp(k)} + \sum_{k=1,k\neq i}^{n''} \bar{a}^{(2)}_{ki}\bar{b}_{p(k)j} \right. \end{aligned} \tag{4.2}$$

$$+ \sum_{k=1,k\neq i}^{n''} \bar{a}_{ki}\bar{b}^{(2)}_{p(k)j} - \sum_{k=1,k\neq i}^{n''} \bar{a}^{(2)}_{ki}\bar{b}^{(2)}_{p(k)j} \Bigg\} .$$

The minimization problem in (4.2) can be solved by using minimal products [43].

In order to prune the entire branch-and-bound tree rooted at this particular node, it is desirable to obtain a lower bound on any solution of the original problem with the restriction of fixed q, i.e., any solution obtainable from the branch-and-bound subtree rooted at the current node. If such a lower bound is available and is larger than the incumbent, then one can fathom the branch-and-bound subtree rooted at the current node. Let us call such a lower bound $lb(q, S_A, S_B)$. Observe that $lb(q, S_A, S_B)$ is not necessarily a lower bound for the original problem. Since a partial assignment q exists at this node, it can be advantageously combined with the lower bound available for the reduced problem (Theorem 3.1) to obtain $lb(q, S_A, S_B)$.

$$l'_{ij} = l_{ij} + \sum_{k\in S_A} a_{ik}b_{jq(k)} + \sum_{k\in S_A} a_{ki}b_{q(k)j}. \tag{4.3}$$

Let lb^* be the optimal solution to the linear assignment problem with costs l'_{ij}. $lb(q, S_A, S_B)$ is defined by

$$lb(q, S_A, S_B) = lb^* + \sum_{k\in S_A} a_{km}b_{q(k)q(m)} + a_{mk}b_{q(m)q(k)}. \tag{4.4}$$

Theorem 4.1. *$lb(q, S_A, S_B)$ is a lower bound on any solution obtained from the nodes of the branch-and-bound subtree rooted at the current node.*

The proof of this theorem follows along the same lines of the proof of Theorem 3.1 given in [32].

The following lemmas, whose proofs follow from the above discussion, characterize the properties of $lb(q, S_A, S_B)$ that are useful in the implementation.

Lemma 4.1. *A node of the branch-and-bound tree is uniquely determined by its descriptor, the tuple (q, S_A, S_B).*

Lemma 4.2. *The matrix L of the reduced subproblem is uniquely determined by S_A and S_B; i.e., two branch-and-bound nodes having the same S_A and S_B will have identical matrix L.*

Lemma 4.3. *In the complete branch-and-bound tree, there are $n'!$ nodes whose descriptor has identical index sets S_A and S_B.*

Note that for all $n'!$ branch-and-bound nodes, the values L are identical. The implementation of the branch-and-bound algorithm exploits this key property. To do this, we first need a definition. Define the *signature* of a node in a branch-and-bound tree to be a function of S_A and S_B of that corresponding node. As the branch-and-bound tree is traversed, the signature of each node is computed. In our implementation the signature is given by

$$\sigma(S_A, S_B) = 2^n \cdot \sum_{i\in S_A} 2^{i-1} + \sum_{j\in S_B} 2^{j-1},$$

i.e., a binary positional representation, where n is the dimension of the original QAP. With this signature we achieve uniqueness, in the sense that for every pair S_A and S_B there corresponds one, and only one, signature $\sigma(S_A, S_B)$. This is computationally efficient, since it avoids collisions in the hash table. However, note that uniqueness is not necessary.

A *hash table* [2, 15] is a data structure for implementing dictionaries (dynamic sets with the operations of insert, delete, and search). The expected time to search an element in a hash table is $O(1)$, which makes hash tables a computationally effective data structure.

If the signature of this node does not match the signature of any previously examined node, the matrix L of that node is computed and saved in a hash table. Otherwise, the computation of L is unnecessary, since its values can be retrieved from the hash table.

The use of signatures and the hash table, as prescribed above, does not avoid having to solve a linear assignment problem at each node. Nevertheless, it reduces substantially the bulk of the work of computing the entries of L.

The computational effort can further be reduced in case L needs to be computed. Observe that the critical computations are the minimal products. Since we use a constant column reduction scheme, we first determine the partitions $\bar{A}_1$, $\bar{A}_2$ and $\bar{B}_1$, $\bar{B}_2$. The columns of $\bar{A}$, $\bar{B}$, $\bar{A}_2$, $\bar{B}_2$, $\bar{A}_1^\top$, $\bar{B}_1^\top$ need to be sorted. Sorting dominates the computational effort at each step of the minimal product computation. Note that since $\bar{A}_2$ and $\bar{B}_2$ have constant columns, there is no need to sort all of the columns of $\bar{A}_2$ and $\bar{B}_2$. Furthermore, observe that since the columns of $\bar{A}$ and $\bar{B}$ are subvectors of the columns of the original A and B matrices, one can presort the original columns once and store the permutation vectors to be retrieved when the sorted columns of $\bar{A}$ and $\bar{B}$ (of the current node) are needed. We make use of two arrays of pointers for each matrix. Array of type `invf(j)` points to the column number of the original matrix of column j is a subvector. Array of type `preperm(i,k)` is the position of the ith element in the kth column in the sorted sequence. Collapsing the retrieved permutation vectors obviates the need for sorting the columns of $\bar{A}$ and $\bar{B}$. Unfortunately, we still are required to sort the columns of matrices $\bar{A}_1$ and $\bar{B}_1$ at each node. This is done with QuickSort [2]. Thus, the complexity of computing of entries of L is bounded by $\mathcal{O}(n''^2 \log n'')$.

5. Computational results. In this section, we present experimental results comparing the variance reduction-based branch-and-bound algorithm with a branch-and-bound algorithm that differs only in the way the lower bounds are computed. The former algorithm uses the LB2(1.0) variant of the new variance reduction bound, while the latter algorithm uses the Gilmore–Lawler lower bound without reduction. This differs from some other implementations [8, 47] of Gilmore–Lawler-based branch-and-bound algorithms where reductions are carried out. The linear assignment problems that need to be solved to compute both lower bounds are solved with the implementation of the auction algorithm [5]. Both algorithms use a GRASP heuristic to compute the initial upper bound. The GRASP was run for 100 iterations on each problem instance.

Two sets of test problems are used in the experiments. The first set is taken from the collection of test problems QAPLIB [9]. The second is a new class of test problems, called *corner*, designed to show effectiveness of the new lower bound on problems with high data variance. The instances in this model have names of the form `rpm`-n-m, where n indicated the dimension of the QAP, and m is the random seed used to generate the instance. The distance and flow matrices are generated as follows. Four squares of size 5×5 are placed in each corner of a 100×100 square and n points are uniformly generated in the small squares such that the number of points in the squares does not differ by more than one. The entries in the distance and flow matrices are the (truncated) Euclidean distances between the points. The instances

Table 5.1
Problem characteristics–corner model.

		A		B		Optimal sol'n	
Name	n	σ	σ/μ	σ	σ/μ	Value	Perm
rpm-7.1	7	47.03	0.64	46.90	0.64	209472	1
rpm-7.2	7	47.07	0.64	46.83	0.64	208822	1
rpm-7.3	7	47.35	0.63	47.05	0.63	212120	1
rpm-7.4	7	46.90	0.64	46.95	0.64	210140	1
rpm-7.5	7	46.57	0.64	47.18	0.64	209382	1
rpm-7.6	7	48.11	0.64	47.25	0.64	211810	2
rpm-7.7	7	47.33	0.64	47.03	0.64	208334	1
rpm-7.8	7	47.70	0.63	46.78	0.63	211252	1
rpm-7.9	7	47.23	0.64	46.99	0.64	210708	1
rpm-7.10	7	47.09	0.64	47.28	0.64	211808	1
rpm-9.1	9	48.70	0.63	48.95	0.63	382018	1
rpm-9.2	9	48.89	0.64	48.82	0.64	379976	1
rpm-9.3	9	49.04	0.63	48.55	0.63	383808	1
rpm-9.4	9	49.37	0.64	48.48	0.64	375596	1
rpm-9.5	9	48.96	0.64	49.15	0.64	383854	1
rpm-9.6	9	48.96	0.64	49.22	0.64	384638	1
rpm-9.7	9	49.05	0.63	48.96	0.63	383324	2
rpm-9.8	9	49.59	0.63	48.13	0.63	379664	1
rpm-9.9	9	49.26	0.64	49.42	0.64	386990	1
rpm-9.10	9	49.54	0.64	49.35	0.64	385426	1
rpm-11.1	11	52.18	0.65	51.21	0.65	635046	1
rpm-11.2	11	52.06	0.66	51.91	0.66	639678	2
rpm-11.3	11	52.17	0.66	51.56	0.66	638560	2
rpm-11.4	11	52.23	0.65	51.67	0.65	638064	1
rpm-11.5	11	51.36	0.66	51.91	0.66	633000	1
rpm-11.6	11	51.60	0.66	51.47	0.66	628034	1
rpm-11.7	11	51.79	0.66	51.92	0.66	632496	1
rpm-11.8	11	52.24	0.66	52.03	0.66	635824	2
rpm-11.9	11	51.70	0.66	50.62	0.66	618010	1
rpm-11.10	11	51.32	0.66	51.96	0.66	629016	1
rpm-13.1	13	51.29	0.68	50.87	0.68	831154	1
rpm-13.2	13	51.11	0.68	50.64	0.68	821550	1
rpm-13.3	13	51.63	0.68	51.69	0.68	844858	2
rpm-13.4	13	51.51	0.68	50.54	0.68	826696	2
rpm-13.5	13	50.94	0.68	51.41	0.68	824084	2

are available from QAPLIB.

Tables 5.1 and 5.2 summarize the data characteristics of the problems considered. In each of these tables, we list the problem name, dimension (n), standard deviation (σ) and coefficient of variability (σ/μ) of input matrices A and B, the value of the optimal solution (Value), and the number of optimal permutations (Perm).

The experiments were conducted on a Silicon Graphics (SGI) Challenge (150 MHz MIPS R4400 processor, 1526 Mbytes of main memory, 16 Kbytes of data cache, and 16 Kbytes of instruction cache). The algorithms were implemented in Fortran and compiled with the `f77` compiler using compiler flags `-O2 -Olimit 800` and times were measured with the system routine `times`.

An upper limit of 2 billion nodes in the search tree was imposed; i.e., all runs that reached 2 billion nodes were terminated. We limit our report to only instances solved within that range. There were no instances for which the algorithm using the Gilmore–Lawler lower bound solved the problem while the one using the new lower bound did not. On the other hand, on several instances, the algorithm with the new lower bound proved optimality of the solution, while the one with the Gilmore–

TABLE 5.2
Problem characteristics–QAPLIB.

		A		B		Optimal sol'n	
Name	n	σ	σ/μ	σ	σ/μ	Value	Perm
chr12a	12	19.702	3.091	28.577	0.634	9552	2
chr12b	12	19.702	3.091	28.577	0.634	9742	1
chr12c	12	19.702	3.091	28.577	0.634	11156	5
chr15a	15	18.437	3.277	31.634	0.699	9896	16
chr15b	15	18.437	3.277	31.634	0.699	7990	5
chr15c	15	18.437	3.277	31.634	0.699	9504	16
esc08a	8	0.294	3.134	0.701	1.122	2	17280
esc08b	8	0.710	1.623	0.701	1.122	8	960
esc08c	8	2.343	1.388	0.701	1.122	32	48
esc08d	8	0.797	1.594	0.701	1.122	6	48
esc08e	8	0.487	2.226	0.701	1.122	2	1344
esc08f	8	1.052	1.295	0.701	1.122	18	96
esc16a	16	0.652	1.704	0.901	0.848	68	13271040
esc16c	16	1.146	1.334	0.901	0.848	160	2064384
esc16e	16	0.526	2.495	0.901	0.848	28	30965760
esc16g	16	0.577	2.546	0.901	0.848	26	46448640
esc16i	16	0.557	2.969	0.901	0.848	14	710277120
lipa10a	10	0.522	0.580	2.753	0.525	473	1
lipa10b	10	3.153	0.713	2.753	0.525	2008	1
lipa20a	20	0.617	0.325	4.498	0.456	7366	22
lipa20b	20	11.788	0.688	4.498	0.456	54152	1
nug05	5	0.891	0.696	1.943	1.104	50	5
nug06	6	0.903	0.650	2.619	1.309	86	4
nug07	7	1.055	0.646	2.418	1.118	148	3
nug08	8	1.098	0.628	3.095	1.286	214	4
nug12	12	1.221	0.571	2.827	1.170	578	4
nug15	15	1.411	0.567	2.817	1.067	1150	6
rou10	10	32.806	0.693	30.696	0.714	174220	1
rou12	12	31.317	0.673	30.301	0.718	235528	3
rou15	15	30.613	0.689	30.263	0.692	354210	6
scr10	10	515.207	2.346	1.214	0.601	26992	1
scr12	12	455.316	2.574	1.221	0.571	31410	8
scr15	15	434.694	2.483	1.331	0.550	51140	2

Lawler lower bound scanned the maximum number of search nodes without verifying optimality.

Tables 5.3 and 5.4 summarize the experimental results on the two sets of test problems. All CPU times are given in seconds. For each instance, the tables list CPU time required by the GRASP and the initial upper bound obtained, and for each of the branch-and-bound algorithms (BB/NLB = branch-and-bound algorithms using the new lower bound, BB/GLB = branch-and-bound algorithm using the Gilmore–Lawler lower bound), the CPU time and number of search tree nodes processed.

We make the following remarks regarding the experiments.

• On all test problems having high data variance in the A and B matrices, the algorithm with the new lower bound consistently dominated the one with the Gilmore–Lawler lower bound. This becomes more evident with the increase in problem size. In the class of largest problem dimension, the code with the Gilmore–Lawler lower bound processed on average 1.8 times the number of nodes processed by the code with the new lower bound, while taking on average 1.4 times the CPU time. See Table 5.3 for details.

Table 5.3
Run statistics–Corner model.

	GRASP		BB/NLB		BB/GLB	
Name	Time	Up bnd	Time	Nodes	Time	Nodes
rpm-7.1	0.07	209472	0.2	885	0.2	1936
rpm-7.2	0.08	208822	0.2	1228	0.2	2031
rpm-7.3	0.08	212120	0.2	1221	0.2	2133
rpm-7.4	0.08	210140	0.2	1302	0.3	2067
rpm-7.5	0.08	209382	0.2	1372	0.2	2066
rpm-7.6	0.07	211810	0.2	1590	0.2	2010
rpm-7.7	0.07	208334	0.2	1279	0.2	2061
rpm-7.8	0.07	211252	0.2	1175	0.2	2136
rpm-7.9	0.08	210708	0.2	1237	0.2	2094
rpm-7.10	0.07	211808	0.2	1236	0.2	2067
rpm-9.1	0.19	382018	3.1	28288	5.0	54724
rpm-9.2	0.16	379976	3.8	35825	6.7	79991
rpm-9.3	0.18	383808	3.1	27705	6.2	68445
rpm-9.4	0.17	375596	2.8	23856	4.2	49064
rpm-9.5	0.20	383854	3.0	25958	5.0	54282
rpm-9.6	0.17	384638	2.9	28790	4.6	43366
rpm-9.7	0.18	383324	2.3	20965	3.7	41759
rpm-9.8	0.17	379664	3.3	31171	6.3	73613
rpm-9.9	0.17	386990	2.3	20218	3.5	39851
rpm-9.10	0.16	385426	2.5	23205	4.2	48824
rpm-11.1	0.37	635046	636.1	5207959	1206.6	7788836
rpm-11.2	0.36	639678	750.9	5488210	974.3	9895900
rpm-11.3	0.35	638560	471.0	3835941	841.6	8712916
rpm-11.4	0.40	638064	452.0	3649701	828.0	8412364
rpm-11.5	0.40	633000	622.8	5271786	781.7	8018040
rpm-11.6	0.34	628034	502.7	4165664	854.8	8779899
rpm-11.7	0.35	632496	682.6	5789411	941.3	10195879
rpm-11.8	0.37	635824	424.1	3367356	822.9	8327159
rpm-11.9	0.43	618010	400.4	3109865	774.9	7656470
rpm-11.10	0.36	629016	612.7	5070997	943.2	9814496
rpm-13.1	0.66	831154	104141.3	662260712	151791.3	1270116829
rpm-13.2	0.67	821550	100907.4	606049824	117561.9	902560201
rpm-13.3	0.67	844858	44756.9	261310480	83343.2	597555186
rpm-13.4	0.71	826696	94319.0	611865352	133840.2	1164849566
rpm-13.5	0.67	824084	94712.6	609694622	126023.5	1054197091

• For the QAPLIB problems, there was a single class having high data variance in both the A and B matrices: `rou`. For that class, the code with the new lower bound also dominated the one using the Gilmore–Lawler lower bound. See Table 5.4 for details.

• Several previously unsolved problems from the QAPLIB were solved to optimality. These were problems `esc16a`, `esc16c`, `esc16e`, `esc16g` and `esc16i` of dimension $n = 16$ and problems `lipa20a` and `lipa20b` of dimension $n = 20$ [34]. Problems `esc16c`, `esc16i`, `lipa20a`, and `lipa20b` were not solved with the code that uses the Gilmore–Lawler lower bound within the limit of 2 billion search tree nodes.

• For the corner model of test problems, the GRASP heuristic found an optimal permutation on all instances. On the QAPLIB suite of problems, the GRASP heuristic found optimal permutation on 23 of the 33 instances solved. This indicates that verification of optimality is the most expensive part of exact algorithms for the QAP.

6. Concluding remarks. In this paper, we presented implementation details and computational results of a new branch-and-bound algorithm for solving the QAP. The algorithm incorporates a new lower bound based on variance reduction techniques

Table 5.4
Run statistics–QAPLIB.

	GRASP		BB/NLB		BB/GLB	
Name	Time	Up bnd	Time	Nodes	Time	Nodes
chr12a	0.45	9916	12.2	36050	0.7	672
chr12b	0.47	9742	3.8	8586	0.6	318
chr12c	0.45	11894	14.7	55845	1.5	3214
chr15a	1.00	11090	594.0	1415685	235.5	413825
chr15b	1.01	9096	93.5	168900	217.8	396255
chr15c	0.98	11366	465.7	1146109	240.0	428722
esc08a	0.07	2	7.3	57464	7.0	57464
esc08b	0.08	8	1.1	6968	0.7	7352
esc08c	0.09	32	0.3	1580	0.3	2552
esc08d	0.08	6	0.3	1448	0.3	2216
esc08e	0.08	2	1.1	10376	1.0	10376
esc08f	0.08	18	0.3	1616	0.3	1520
esc16a	1.01	68	15786.8	58018200	15216.0	60244656
esc16c	1.01	160	133372.4	428754386	-	-
esc16e	0.95	28	18013.8	97558848	14811.1	99030192
esc16g	1.05	26	17844.9	127106352	14542.3	132664368
esc16i	0.98	14	265900.8	1932419536	-	-
lipa10a	0.22	473	0.3	90	0.3	181
lipa10b	0.23	2008	0.3	126	0.3	126
lipa20a	2.56	7506	3182.8	2772772	-	-
lipa20b	2.73	74152	486.0	551	-	-
nug05	0.02	52	0.0	44	0.0	44
nug06	0.04	86	0.1	86	0.1	82
nug07	0.06	148	0.1	127	0.1	115
nug08	0.10	214	0.3	980	0.2	895
nug12	0.40	578	15.7	52626	14.6	49063
nug15	0.97	1152	1012.3	2106172	912.4	1794507
rou10	0.22	174220	0.7	1529	0.8	2683
rou12	0.50	235852	6.5	16309	12.3	37982
rou15	0.96	362518	1276.8	2805138	2240.3	4846805
scr10	0.25	26992	2.9	16162	0.6	1494
scr12	0.46	31410	104.4	408048	4.8	12918
scr15	1.01	51140	2269.3	5609533	274.7	506360

and uses a GRASP heuristic to produce the initial upper bound. The algorithm computes all optimal permutations of the QAP.

The algorithm was compared with an implementation using the Gilmore–Lawler lower bound and was found to perform better in problems having high data variance in the A and B input matrices. The new algorithm produced optimal solutions for several previously unsolved instances from the QAPLIB.

The data structures incorporated in the branch-and-bound codes can be useful in other branch-and-bound approaches for solving QAPs. The algorithm can be implemented in parallel to reduce running time requirements [41, 42]. Finally, the branch-and-bound scheme proposed in this paper can be implemented with other lower bounds, such as the linear programming based lower bounds [1, 46] and lower bounds based on eigenvalues [24, 44].

Acknowledgment. The authors would like to thank the anonymous referees for extensive comments that substantially improved the readability of the paper.

REFERENCES

[1] W. Adams and T. Johnson, *Improved linear programming-based lower bounds for the quadratic assignment problem*, in Quadratic Assignment and Related Problems, P. Pardalos and H. Wolkowicz, eds., DIMACS Series on Discrete Mathematics and Theoretical Computer Science, Vol. 16, American Mathematical Society, Providence, RI, 1994, pp. 43–75.

[2] A. Aho, J. Hopcroft, and J. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

[3] A. Assad and W. Xu, *On lower bounds for a class of quadratic* $\{0, 1\}$ *programs*, Oper. Res. Lett., 4 (1985), pp. 175–180.

[4] M. Bazaraa and H. Sherali, *New approaches for solving the quadratic assignment problem*, Oper. Res. Verfahren, 32 (1979), pp. 29–46.

[5] D. Bertsekas, *Linear Network Optimization: Algorithms and Codes*, MIT Press, Cambridge, MA, 1991.

[6] S. Bokhari, *Assignment problems in parallel and distributed computing*, in The Kluwer International Series in Engineering and Computer Science, Kluwer Academic Publishers, Boston, Dordrecht, Lancaster, 1987.

[7] R. Burkard and T. Bonniger, *A heuristic for quadratic boolean programs with applications to quadratic assignment problems*, European J. Oper. Res., 13 (1983), pp. 374–386.

[8] R. Burkard and U. Derigs, *Assignment and matching problems: Solution methods with Fortran programs*, in Lecture Notes in Economics and Mathematical Systems, Vol. 184, Springer, Berlin, 1980.

[9] R. Burkard, S. Karisch, and F. Rendl, *QAPLIB – a quadratic assignment problem library*, European J. Oper. Res., 55 (1991), pp. 115–119.

[10] R. Burkard and F. Rendl, *A thermodynamically motivated simulation procedure for combinatorial optimization problems*, European J. Oper. Res., 17 (1984), pp. 169–174.

[11] P. Carraresi and F. Malucelli, *A new lower bound for the quadratic assignment problem*, Oper. Res., 40 (1992), pp. S22–S27.

[12] N. Christofides and E. Benavent, *An exact algorithm for the quadratic assignment problem*, Oper. Res., 37 (1989), pp. 760–768.

[13] N. Christofides and M. Gerrard, *A graph theoretic analysis of bounds for the quadratic assignment problem*, in Studies on Graphs and Discrete Programming, P. Hansen, ed., North–Holland, Amsterdam, 1981, pp. 61–68.

[14] J. Clausen, Announcement on Discrete Mathematics and Algorithms Network (DIMANET), 1994.

[15] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1990.

[16] C. Edwards, *A branch-and-bound algorithm for the Koopmans-Beckman quadratic assignment problem*, Math. Programming Study, 13 (1980), pp. 35–52.

[17] T. Feo and M. Resende, *Greedy randomized adaptive search procedures*, J. Global Optim., 6 (1995), pp. 109–133.

[18] G. Finke, R. Burkard, and F. Rendl, *Quadratic assignment problems*, Ann. Discrete Math., 31 (1987), pp. 61–82.

[19] C. Fleurent and J. Ferland, *Genetic hybrids for the quadratic assignment problem*, in Quadratic Assignment and Related Problems, P. Pardalos and H. Wolkowicz, eds., DIMACS Series on Discrete Mathematics and Theoretical Computer Science, Vol. 16, American Mathematical Society, Providence, RI, 1994, pp. 173–187.

[20] R. Francis and J. White, *Facility Layout and Location*, Prentice–Hall, Englewood Cliffs, NJ, 1974.

[21] A. Frieze and J. Yadegar, *On the quadratic assignment problem*, Discrete Appl. Math., 5 (1983), pp. 89–98.

[22] P. C. Gilmore, *Optimal and suboptimal algorithms for the quadratic assignment problem*, J. Soc. Indust. Appl. Math., 10 (1962), pp. 305–313.

[23] S. Hadley, F. Rendl, and H. Wolkowicz, *Bounds for the quadratic assignment problem using continuous optimization techniques*, in Integer Programming and Combinatorial Optimization, University of Waterloo Press, Waterloo, Ontario, Canada, 1990, pp. 237–248.

[24] S. Hadley, F. Rendl, and H. Wolkowicz, *A new lower bound via projection for the quadratic assignment problem*, Math. Oper. Res., 17 (1992), pp. 727–739.

[25] L. Hubert, *Assignment Methods in Combinatorial Data Analysis*, Marcel Dekker, Inc., New York, NY, 1987.

[26] T. Ibaraki, *Theoretical comparisons of search strategies in branch-and-bound algorithms*, Internat. J. Comput. Inform. Sci., 5 (1976), pp. 315–344.

[27] B. Jansen, *A Note on "Lower Bounds for the QAP"*, Tech. rep., Delft University of Technology, Mathematics and Computer Science, Delft, The Netherlands, December, 1993.

[28] T. Kaufman and F. Broeckx, *An algorithm for the quadratic assignment problem using Bender's decomposition*, European J. Oper. Res., 2 (1978), pp. 204–211.

[29] T. Koopmans and M. Beckmann, *Assignment problems and the location of economic activities*, Econometrica, 25 (1957), pp. 53–76.

[30] J. Krarup and P. Pruzan, *Computer-aided layout design*, Math. Programming Study, 9 (1978), pp. 75–94.

[31] E. Lawler, *The quadratic assignment problem*, Management Sci., 9 (1963), pp. 586–599.

[32] Y. Li, P. Pardalos, K. Ramakrishnan, and M. Resende, *Lower bounds for the quadratic assignment problem*, Ann. Oper. Res., 50 (1994), pp. 387–410.

[33] Y. Li, P. Pardalos, and M. Resende, *A greedy randomized adaptive search procedure for the quadratic assignment problem*, in Quadratic Assignment and Related Problems, P. Pardalos and H. Wolkowicz, eds., DIMACS Series on Discrete Mathematics and Theoretical Computer Science, Vol. 16, American Mathematical Society, Providence, RI, 1994, pp. 237–261.

[34] Y. Li and P. M. Pardalos, *Generating quadratic assignment test problems with known optimal permutations*, Comput. Optim. Appl., 1 (1992), pp. 163–184.

[35] T. Mautor and C. Roucairol, *Difficulties of exact methods for solving the quadratic assignment problem*, in Quadratic Assignment and Related Problems, P. Pardalos and H. Wolkowicz, eds., DIMACS Series on Discrete Mathematics and Theoretical Computer Science, Vol. 16, American Mathematical Society, Providence, RI, 1994, pp. 263–274.

[36] T. Mautor and C. Roucairol, *A new exact algorithm for the solution of quadratic assignment problems*, Discrete Appl. Math., 55 (1994), pp. 150–173.

[37] H. Mawengkang and B. Murtagh, *Solving nonlinear integer programs with large-scale optimization software*, Ann. Oper. Res., 5 (1985/6), pp. 425–437.

[38] E. McCormik, *Human Factors Engineering*, McGraw-Hill, New York, 1970.

[39] B. Murtagh, T. Jefferson, and V. Sornprasit, *A heuristic procedure for solving the quadratic assignment problem*, European J. Oper. Res., 9 (1982), pp. 71–76.

[40] C. Nugent, T. Vollmann, and J. Ruml, *An experimental comparison of techniques for the assignment of facilities to locations*, J. Oper. Res., 16 (1969), pp. 150–173.

[41] P. Pardalos and J. Crouse, *A parallel algorithm for the quadratic assignment problem*, in Proc. Supercomputing 1989 Conference, ACM Press, 1989, pp. 351–360.

[42] P. Pardalos, A. Phillips, and J. Rosen, *Topics in Parallel Computing in Mathematical Programming*, Science Press, New York, Beijing, 1993.

[43] P. Pardalos and H. Wolkowicz, eds., *Quadratic assignment and related problems*, DIMACS Series on Discrete Mathematics and Theoretical Computer Science, Vol. 16, American Mathematical Society, Providence, RI, 1994.

[44] F. Rendl and H. Wolkowicz, *Applications of parametric programming and eigenvalue maximization to the quadratic assignment problem*, Math. Programming, 53 (1992), pp. 63–78.

[45] M. Resende, P. Pardalos, and Y. Li, *Algorithm* 754*: FORTRAN subroutines for approximate solution of dense quadratic assignment problems using GRASP*, ACM Trans. Math. Software, 22 (1996), pp. 104–118.

[46] M. Resende, K. Ramakrishnan, and Z. Drezner, *Computing lower bounds for the quadratic assignment problem with an interior point algorithm for linear programming*, Oper. Res., 43 (1995), pp. 781–791.

[47] C. Roucairol, *A parallel branch and bound algorithm for the quadratic assignment problem*, Discrete Appl. Math., 18 (1987), pp. 211–225.

[48] J. Skorin-Kapov, *Tabu search applied to the quadratic assignment problem*, ORSA J. Comput., 2 (1990), pp. 33–45.

[49] E. Taillard, *Robust tabu search for the quadratic assignment problem*, Parallel Computing, 17 (1991), pp. 443–455.

[50] U. W. Thonemann and A. M. Bölte, *An Improved Simulated Annealing Algorithm for the Quadratic Assignment Problem*, Tech. rep., School of Business, Dept. of Production and Operations Research, University of Paderborn, Germany, 1994.

SIAM J. OPTIM.
Vol. 7, No. 2, pp. 295–303, May 1997

ON THE SELF-CONCORDANCE OF THE UNIVERSAL BARRIER FUNCTION [*]

OSMAN GÜLER[†]

Abstract. Let K be a regular convex cone in $\mathbb{R}^n$ and let $F(x)$ be its universal barrier function. Let $D^kF(x)[h,\dots,h]$ be the kth order directional derivative at the point $x \in K^0$ and direction $h \in \mathbb{R}^n$. We show that for every $m \geq 3$ there exists a constant $c(m) > 0$ depending only on m such that $|D^mF(x)[h,\dots,h]| \leq c(m)\, D^2F(x)[h,h]^{m/2}$. For $m = 3$, this is the self-concordance inequality of Nesterov and Nemirovskii. Our proof uses a powerful recent result of Bourgain.

Key words. universal barrier function, self-concordance, interior point methods

AMS subject classifications. Primary 90C25, 90C60, 52A41; Secondary 90C06, 52A40

PII. S105262349529180X

1. Introduction.

Interior point methods have occupied a prominent place in continuous optimization ever since Karmarkar [7] introduced his polynomial-time projective algorithm for linear programming in 1984. Although much of the early activities were in linear programming and monotone linear complementarity problems, Nesterov and Nemirovskii [11] have successfully developed a theory of interior point methods for general nonlinear convex programming problems and monotone variational inequalities. One of the key ideas of this theory is the notion of a self-concordant barrier function for a convex set.

We recall some relevant concepts from [11]. Let $Q \subseteq \mathbb{R}^n$ be an open convex set. A function $F : Q \to \mathbb{R}$ is called a *self-concordant barrier function* if it is at least three times differentiable, convex, and satisfies the properties

$$|D^3F(x)[h,h,h]| \leq 2(D^2F(x)[h,h])^{3/2}, \tag{1}$$

$$|DF(x)[h]|^2 \leq \vartheta D^2F(x)[h,h], \tag{2}$$

and

$$F(x) \to \infty \quad \text{as } x \to \partial Q.$$

Here $D^kF(x)[h,\dots,h]$ is the kth directional of F at x along the direction $h \in \mathbb{R}^n$, and the constant ϑ is called the parameter of the barrier function. The parameter ϑ determines, in theory, the speed of the underlying interior point method.

Let $K \subseteq \mathbb{R}^n$ be a *regular cone*, that is, a convex cone containing no lines and having a nonempty interior. (There is no essential loss of generality in restricting attention to regular cones.) A function F satisfying (1) is called a ϑ-logarithmically homogeneous barrier for K if it is a barrier function for K (that is, $F(x) \to \infty$ as $x \to \partial K$) and satisfies the property

$$F(tx) = F(x) - \vartheta \log t. \tag{3}$$

[*] Received by the editors September 13, 1995; accepted for publication January 15, 1996. This research was partially supported by the National Science Foundation under grant DMS-9306318.
http://www.siam.org/journals/siopt/7-2/29180.html

[†] Department of Mathematics and Statistics, University of Maryland Baltimore County, Baltimore, MD 21228 (guler@math.umbc.edu).

That is, the function $\varphi(x) = e^{F(x)}$ is $-\vartheta$ homogeneous; $\varphi(tx) = \varphi(x)/t^{\vartheta}$. The function F is called a *ϑ-normal barrier* for K if it is ϑ-logarithmically homogeneous. It is well known that (3) implies (2) Nesterov and Nemirovskii [11, Proposition 5.1.4] show that any self-concordant barrier function on a convex set with nonempty interior can be extended to a logarithmically homogeneous self-concordant barrier function on the cone $K(Q)$ fitted to Q (conic hull in the terminology of [11]). This is the cone

$$K(Q) := \{(tx, t) : x \in Q, t \geq 0\}.$$

One of the most important theoretical results in Nesterov and Nemirovskii [11] is the existence of a self-concordant barrier, called the *universal barrier* function, for any open convex set $Q \subseteq \mathbb{R}^n$. This function is logarithmically homogeneous if Q is a regular cone. Nesterov and Nemirovskii define the universal barrier function for Q as

$$F(x) = \log u(x); \qquad u(x) = vol_n(Q^*(x)) = |Q^*(x)|,$$

where vol_n stands for the n-dimensional Lebesgue measure and $Q^*(x)$ is the polar set of Q centered at x; that is,

$$Q^*(x) = \cap_{z \in Q}\{y \in \mathbb{R}^n : \langle z - x, y\rangle \leq 1\}.$$

We will use a somewhat different representation of the universal barrier function. It is shown in [4] that the universal barrier can be written (up to an additive constant) as the logarithm of the characteristic function $\varphi_{K(Q)}$ of the cone $K(Q)$ fitted to Q:

$$\varphi_{K(Q)}(x) = \int_{K(Q)^*} e^{-\langle x,y\rangle} dy, \tag{4}$$

where

$$K(Q)^* = \cap_{x \in K(Q)}\{y : \langle x, y\rangle \geq 0\}$$

is the dual cone of $K(Q)$. The same formula (4) holds true if $K \subseteq \mathbb{R}^n$ is a regular cone. Thus, for such a cone we have

$$F(x) = \text{const} + \log\varphi(x) = \text{const} + \log\int_{K^*} e^{-\langle x,y\rangle} dy.$$

In this paper we will, without loss of any generality, restrict our attention to regular cones and their universal barrier functions.

The existence of the universal barrier function is an important result since it implies that one can, in theory, design an interior point method to solve *any* convex programming problem in polynomial time. However, if one uses only the defining properties of the self-concordant barrier functions (namely, the inequalities (1) and (2)), then the resulting interior point methods are short-step methods that are not likely to be efficient in practice. There have been recent efforts to obtain more efficient long-step interior methods for special cones/barriers; see [13], [14], [12], and [5].

In this paper, we proceed towards the same goal but via a different direction. We show that the universal barrier function for any regular cone enjoys properties that may play a role towards designing long-step interior point methods. Our main result, Theorem 4.1, states that for every $m \geq 3$ there exists a constant $c(m) > 0$ that depends only on m such that

$$|D^m F(x)[h, h, \ldots, h]| \leq c(m)(D^2 F(x)[h, h])^{m/2}. \tag{5}$$

If $m = 3$, this is just the self-concordance inequality (1). Our strategy to proving (5) is similar to the one in Nesterov and Nemirovskii [11] in that we first try to express the derivatives $D^m F(x)[h, \ldots, h]$ in terms of the mean value and central moments of a suitable random variable and then try to obtain inequalities between different central moments. However, our method of obtaining these moment inequalities differs from the one in [11]. Essentially, we replace the most difficult parts of the proof of Theorem 2.5.1 in [11] (starting from 3^0 on p. 52) with, in our opinion, a more flexible recent result of Bourgain [1]. In fact, once one has

$$|\theta| \leq O(1)\sigma$$

(see [11], p. 52), Bourgain's theorem immediately implies (1).

For $m = 4$, the inequality (5) has been used in [6] in a line search procedure to minimize a logarithmic barrier function along a given direction. Also, the inequality

$$|D^4 F(x)[h, h, \ldots, h]| \leq \alpha(\alpha + 1)(D^2 F(x)[h, h]) \, ||h||^2_{Q,x} \tag{6}$$

is used in [12] to devise some long-step interior point methods. We note that any universal barrier satisfies (6), since

$$|D^4 F(x)[h, h, h, h]| \leq c(4) D^2 F(x)[h, h]^2 \leq c(4)(1 + 3\vartheta)^2 D^2 F(x)[h, h] \cdot ||h||^2_{K,x}, \tag{7}$$

where the first inequality follows from (5) and the second one follows from inequality (2.3.9) in [11]. In general, we have $c(4)(1 + 3\vartheta)^2 = O(n^2)$. However, in some cases $c(4) = O(n^{-2})$ so $c(4)(1 + 3\vartheta)^2 = O(1)$; see the examples in [12]. It is then possible to scale the universal barrier to obtain a barrier function with a smaller ϑ and still satisfy the inequality (6) with a constant $\alpha = O(1)$ as (6) is scale invariant.

The paper is organized as follows. In section 2, we state and describe a powerful recent result of Bourgain concerning the behavior of polynomials on convex bodies. In section 3, we evaluate the directional derivatives of the universal barrier function in a form suitable to prove (5). This section may be skipped at a first reading except for the statement of Lemma 3.1. In section 4, we prove our main result (5).

2. Bourgain's theorem.

The following important result of Bourgain [1] in geometric functional analysis will be very useful for us.

THEOREM 2.1. *For every positive integer d and every $p < \infty$, there exists a universal constant $c(d, p) > 0$ such that the following is true: if $Q \subseteq \mathbb{R}^n$ is a convex body and $f : \mathbb{R}^n \to \mathbb{R}$ is a polynomial of degree d, then*

$$\left(\frac{\int_Q |f(x)|^p dx}{\int_Q dx} \right)^{1/p} \leq c(d, p) \frac{\int_Q |f(x)| dx}{\int_Q dx}.$$

Defining the uniform probability measure $dP(x) = dx/|Q|$ on Q, Bourgain's theorem becomes

$$\left(\int_Q |f(x)|^p dP(x) \right)^{1/p} \leq c(d, p) \int_Q |f(x)| dP(x).$$

Bourgain makes use of the so-called Knothe mapping that was first used by Knothe [8] to prove the Brunn–Minkowski theorem. A special case of the theorem for $d = 1$ is proved earlier by Gromov and Milman [3] and is called a “concentration of measure” phenomenon. The reason for the terminology can be explained as follows. Since

$$|f(x)|^p = \int_0^{|f(x)|} pt^{p-1}dt,$$

we have

$$\begin{aligned}\int_Q |f(x)|^p dx = p\int_Q \left(\int_0^{|f(x)|} t^{p-1}dt\right) dx = p\int_{\{(x,t):x\in Q, 0\le t\le |f(x)|\}} t^{p-1}dxdt\\ = p\int_0^\infty t^{p-1} vol_n(\{x\in Q : |f(x)|\ge t\})dt = p\int_0^\infty t^{p-1}|Q_t|dt,\end{aligned}$$

where the second and third equalities follow from the Fubini theorem and

$$Q_t := \{x \in Q : |f(x)| \ge t\}.$$

Here $|Q_t|$ is the Lebesgue measure of Q_t and is called the distribution function of f. Assuming $|Q| = 1$ and $\int_Q |f(x)| = 1$, Bourgain proves his theorem by showing that $|Q_t|$ decreases exponentially; that is,

$$|Q_t| \le e^{-t^{cd}}$$

for some absolute constant $c > 0$. Similar concentration of measure ideas are used by Milman to prove the Dvoretzky theorem in geometric functional analysis; see [10].

3. Directional derivatives of the universal barrier function. Let $K \subseteq \mathbb{R}^n$ be a regular cone. Fix a point $x \in K^0$ and a direction $h \in \mathbb{R}^n$. Define $g(t) = \varphi(x+th)$, and consider the function

$$h(t) := F(x + th) = \log\varphi(x + th) = \log g(t) = f(g(t)),$$

where

$$f(s) = \log s.$$

In this section, we will obtain expressions for the derivatives $h^{(k)}(t)$, $k \ge 1$, that will be useful for proving our main result (5). By definition,

$$h^{(m)}(t) = D^m F(x + th)[h, \dots, h].$$

Since h is a composite function, its derivatives are given by the Faà di Bruno formula. This has some combinatorial aspects and states that if f and g are any two functions and $h(t) = f(g(t))$ their composition, then

$$\frac{h^{(m)}(t)}{m!} = \sum_\lambda \left(\begin{array}{c} k \\ k_1, k_2, \dots, k_m\end{array}\right) \frac{f^{(k)}(g(t))}{k!} \left(\frac{g'(t)}{1!}\right)^{k_1} \left(\frac{g''(t)}{2!}\right)^{k_2} \cdots \left(\frac{g^{(m)}(t)}{m!}\right)^{k_m};$$
(8)

see, for example, Comtet [2, p. 137], and Knuth [9, pp. 50 and 480–481]. Here $k = k_1 + k_2 + \cdots + k_m$,

$$\left(\begin{array}{c} k \\ k_1, k_2, \dots, k_m\end{array}\right) = \frac{k!}{k_1!\cdots k_m!}$$

is a multinomial coefficient, and $\lambda = (1^{k_1}, 2^{k_2}, \ldots, m^{k_m})$ is a *partition* of the number m in which 1 occurs k_1 times, 2 occurs k_2 times, etc. Thus, $k_i \geq 0$ are integers satisfying

$$m = k_1 + 2k_2 + \cdots + mk_m,$$

and k is the number of parts in the partition λ.

We begin by evaluating the derivatives $g^{(k)}(t)$. We assume, without losing generality (because of logarithmic homogeneity), that $||x|| = 1$. From equation (4), we have

$$\begin{aligned} g(t) = \varphi(x+th) = \int_{K^*} e^{-\langle x,y\rangle} e^{-t\langle h,y\rangle} dy = \int_0^\infty e^{-s} \int_{\{y\in K^* : \langle x,y\rangle = s\}} e^{-t\langle h,y\rangle} dy\, ds \\ = b^{1-n} \int_0^\infty s^{n-1} e^{-s} \int_{\{y\in K^* : \langle x,y\rangle = b\}} e^{-st\langle h/b,y\rangle} dy\, ds, \end{aligned}$$

where the third equation follows from the coarea formula (see [4], Theorem 4.1), and the last equation follows from the change of variables formula. Here, $b > 0$ is chosen such that the set

$$Q := \{y \in K^* : \langle x,y\rangle = b\}$$

has volume 1; that is, $vol_{n-1}(Q) = |Q| = 1$.

Consider the uniform probability distribution on Q and let

$$\alpha := \int_Q \langle h/b, y\rangle dy$$

be the mean value of the random variable $\langle h/b, y\rangle$ on Q. From the equation above for $g(t)$, we have

$$b^{n-1} g(t) = \int_0^\infty s^{n-1} e^{-s(1+t\alpha)} \left(\int_Q e^{-st(\langle h/b,y\rangle - \alpha)} dy \right) ds,$$

and the inner integral can be expanded as

$$\int_Q e^{-st(\langle h/b,y\rangle-\alpha)} dy = \sum_{k=0}^\infty \frac{(-1)^k}{k!} s^k t^k \int_Q (\langle h/b,y\rangle - \alpha)^k dy = \sum_{k=0}^\infty \frac{(-1)^k}{k!} s^k t^k \mu_k,$$

where μ_k is the kth central moment of the random variable $\langle h/b, y\rangle$ on Q; that is,

$$\mu_k := \int_Q (\langle h/b,y\rangle - \alpha)^k dy.$$

Note that $\mu_0 = 1$ and $\mu_1 = 0$.

The above formulas give

$$\begin{aligned} b^{n-1} g(t) = \sum_{k=0}^\infty \frac{(-1)^k \mu_k}{k!} t^k \int_0^\infty s^{n+k-1} e^{-s(1+t\alpha)} ds = \sum_{k=0}^\infty \frac{(-1)^k \mu_k}{k!} t^k \frac{(n+k-1)!}{(1+t\alpha)^{n+k}} \\ = \frac{(n-1)!}{(1+t\alpha)^n} \left[1 + \sum_{k=1}^\infty (-1)^k \left(\begin{array}{c} n+k-1 \\ k \end{array} \right) \mu_k \frac{t^k}{(1+t\alpha)^k} \right] \\ = (n-1)!(1+t\alpha)^{-n} v(t), \end{aligned}$$

where

$$v(t) := 1 + \sum_{k=1}^{\infty} (-1)^k \left(\begin{array}{c} n+k-1 \\ k \end{array} \right) \mu_k t^k (1+t\alpha)^{-k}.$$

Applying (8) to the composite function $f_1(g_1(t))$ with $f_1(s) = s^{-k}$ and $g_1(t) = 1 + t\alpha$, we obtain

$$(1+t\alpha)^{-k} = \sum_{l=0}^{\infty} (-1)^l \left(\begin{array}{c} k+l-1 \\ l \end{array} \right) \alpha^l t^l.$$

Therefore,

$$\begin{aligned} v(t) &= 1 + \sum_{k=1,l=0}^{\infty} (-1)^{k+l} \left(\begin{array}{c} k+l-1 \\ l \end{array} \right) \left(\begin{array}{c} n+k-1 \\ k \end{array} \right) \mu_k \alpha^l t^{k+l} \\ &= 1 + \sum_{i=1}^{\infty} (-1)^i \left[\sum_{j=1}^{i} \left(\begin{array}{c} i-1 \\ i-j \end{array} \right) \left(\begin{array}{c} n+j-1 \\ j \end{array} \right) \mu_j \alpha^{i-j} \right] t^i \\ &:= 1 + \sum_{i=1}^{\infty} (-1)^i c_i t^i = 1 + \sum_{i=2}^{\infty} (-1)^i c_i t^i, \end{aligned}$$

where c_i is the expression within the square brackets, and the last equation follows since $\mu_1 = 0$.

We are finally ready to calculate the derivatives $h^{(m)}(0)$. Since $g(t) = b^{1-n}(n-1)!(1+t\alpha)^{-n} v(t)$, defining

$$w(t) = \log v(t)$$

results in

$$h(t) = \text{const} - n \log(1 + \alpha t) + w(t).$$

Since $f(s) = \log s$, we have

$$\frac{f^{(k)}(s)}{k!} = \frac{(-1)^{k-1}}{k} \frac{1}{s^k}.$$

Note that $v(0) = 1$, $v^{(i)}(0)/i! = (-1)^i c_i$, and $m = k_1 + 2k_2 + \cdots + mk_m$. Using these in (8) gives

$$\frac{w^{(m)}(0)}{m!} = (-1)^m \sum_{\lambda} \frac{(-1)^{k-1}}{k} \left(\begin{array}{c} k \\ k_1, k_2, \ldots, k_m \end{array} \right) c_1^{k_1} c_2^{k_2} \cdots c_m^{k_m}.$$

Since

$$\frac{d^{(m)} \log(1+\alpha t)}{dt^m} = (-1)^{m-1} (m-1)! \alpha^m (1+\alpha t)^{-m},$$

we obtain the following expression for the directional derivatives $D^m F(x)[h, \ldots, h] = h^{(m)}(0)$.

Lemma 3.1. *Let $K \subseteq \mathbb{R}^n$ be a regular cone. Fix a point $x \in K^0$ and a direction $h \in \mathbb{R}^n$. Let*

$$Q = \{y \in K^* : \langle x, y\rangle = b||x||\}$$

be a cross section of the cone K^* where b is chosen such that $vol_{n-1}(Q) = 1$. Consider the uniform probability distribution and the random variable $\langle h/(b||x||), y\rangle$ on Q. Let α and μ_j be the mean value and the jth central moment of this random variable. Then, we have

$$\frac{(-1)^m}{m!} D^m F(x)[h, \ldots, h] = \frac{1}{m}\, n\, \alpha^m + \sum_{\lambda} \frac{(-1)^{k-1}}{k} \begin{pmatrix} k \\ k_2, k_3, \ldots, k_m \end{pmatrix} c_2^{k_2}\, c_3^{k_3} \cdots c_m^{k_m}, \tag{9}$$

where the sum is over all partitions $\lambda = (1^0, 2^{k_2}, \ldots, m^{k_m})$ of the number m,

$$c_i = \sum_{j=2}^{i} \begin{pmatrix} i-1 \\ i-j \end{pmatrix} \begin{pmatrix} n+j-1 \\ j \end{pmatrix} \mu_j \alpha^{i-j} = \sum_{j=2}^{i} \begin{pmatrix} i-1 \\ i-j \end{pmatrix} \frac{1}{j!} \left([n]^j \mu_j\right) \alpha^{i-j}, \tag{10}$$

and

$$[n]^j := n(n+1)\cdots(n+j-1).$$

Proof. The preceding calculations prove the lemma for $||x|| = 1$. The general case follows since (3) implies that

$$D^m F(tx)[th, \ldots, th] = D^m F(x)[h, \ldots, h]$$

for any $t > 0$. Thus,

$$D^m F(x)[h, \ldots, h] = D^m F(x/||x||)[h/||x||, \ldots, h/||x||],$$

and the lemma is proved. □

The first few directional derivatives easily can be calculated:

$$\begin{aligned} DF(x)[h] &= -n\alpha, \\ D^2 F(x)[h, h] &= [n]^2 \mu_2 + n\alpha^2, \\ D^3 F(x)[h, h, h] &= -[n]^3 \mu_3 - 6[n]^2 \mu_2 \alpha - 2n\alpha^3, \\ D^4 F(x)[h, h, h, h] &= [n]^4 \mu_4 - 3([n]^2 \mu_2)^2 + 12[n]^3 \mu_3 \alpha + 36[n]^2 \mu_2 \alpha^2 + 6n\alpha^4. \end{aligned} \tag{11}$$

Remark. Our approach to calculating the directional derivatives $D^m F(x)[h, \ldots, h]$ is similar to the one in [11] in that we express them in terms of the mean value and the central moments of a suitable random variable. Our first approach was to express $D^m F(x)[h, \ldots, h]$ in terms of the mean value and usual moments of a random variable and then to express these moments in terms of the central ones. This resulted in a seemingly hard combinatorial problem, but our computer evaluations of $D^m F(x)[h, \ldots, h]$ up to $m \leq 8$ led us to conjecture Lemma 3.1. The author thanks his colleague Peter Matthews for suggesting that we first express $g^m(t)$ in terms of α and μ_j.

4. Self-concordance of the universal barrier function. In this section we use Theorem 2.1 and Lemma 3.1 to prove our main result.

THEOREM 4.1. *Let $K \subseteq \mathbb{R}^n$ be a regular cone and $F(x)$ its universal barrier. Then for any $m \geq 3$,*

$$|D^m F(x)[h, h, \dots, h]| \leq c(m)(D^2 F(x)[h, h])^{m/2},$$

where $c(m)$ is a constant that depends only on m.

Proof. The crucial ingredient of the proof is the fact that each term μ_k is coupled with the term $[n]^k$. Expanding the products of the terms c_k in equation (9) of Lemma 3.1 allows us to write

$$\begin{aligned} D^m F(x)[h, \dots, h] = & \sum_\lambda d_{k_1,k_2,\dots,k_m} ([n]^2\mu_2)^{k_2} \, ([n]^3\mu_3)^{k_3} \cdots ([n]^m\mu_m)^{k_m} \alpha^{k_1} \\ & + (-1)^m (m-1)! \, n \, \alpha^m, \end{aligned} \tag{12}$$

where the summation is taken over all partitions λ of the number m except the partition $\lambda = (1^m, 2^0, \dots, 0)$ which corresponds to the term $(-1)^m(m-1)! \, n \, \alpha^m$, and where the constants $d_{k_1,k_2,\dots,k_m}$ only depend on m.

We compare each term on the right-hand side of (12) with the terms of

$$D^2 F(x)[h, h]^m = ([n]^2\mu_2 + n\alpha^2)^m;$$

see (11). First, comparing the last term in (12) with the term $(n\alpha^2)^m$ in $([n]^2\mu_2 + n\alpha^2)^m$ shows that

$$|(m-1)! \, n\alpha^m| \leq O(1) \, (n\alpha^2)^{m/2} \leq O(1) \, D^2 F(x)[h, h]^{m/2}.$$

Theorem 2.1 applied with $p = k/2$ to the function

$$f(y) = (\langle h/(b||x||), y\rangle - \alpha)^2$$

on the set Q defined in Lemma 3.1 gives

$$\begin{aligned} |\mu_k| &= \left| \int_Q (\langle h/(b||x||), y\rangle - \alpha)^k dy \right| \leq \int_Q |\langle h/(b||x||), y\rangle - \alpha|^k dy = \int_Q |f(y)|^{k/2} dy \\ &\leq c(k) \left(\int_Q |f(y)| dy \right)^{k/2} = c(k) \mu_2^{k/2} \end{aligned}$$

for some constant $c(k)$ depending only on k. Consequently, we have

$$\begin{aligned} |t_{k_1,k_2,\dots,k_m}| &:= \left| d_{k_1,k_2,\dots,k_m} \, ([n]^2\mu_2)^{k_2} \, ([n]^3\mu_3)^{k_3} \cdots ([n]^m\mu_m)^{k_m} \alpha^{k_1} \right| \\ &= O(n^{m-k_1}) \, \mu_2^{(m-k_1)/2} |\alpha|^{k_1}. \end{aligned}$$

Comparing $t_{k_1,k_2,\dots,k_m}$ with the term

$$D^2 F(x)[h, h]^m \geq ([n]^2\mu_2)^{m-k_1} (n\alpha^2)^{k_1} = O(n^{2m-k_1}) \, \mu_2^{m-k_1} \alpha^{2k_1}$$

in $([n]^2\mu_2 + n\alpha^2)^m$ shows that

$$\begin{aligned} (t_{k_1,k_2,\dots,k_m})^2 &= O(n^{2m-2k_1}) \, \mu_2^{m-k_1} \alpha^{2k_1} \leq O(n^{2m-k_1}) \, \mu_2^{m-k_1} \alpha^{2k_1} \\ &\leq O(1) \, D^2 F(x)[h, h]^m. \end{aligned}$$

This proves the theorem. □

REFERENCES

[1] J. Bourgain (1991), *On the distribution of polynomials on high dimensional convex sets*, in Lecture Notes in Mathematics 1469, Springer-Verlag, Berlin, New York, pp. 127–137.

[2] L. Comtet (1974), *Advanced Combinatorics*, D. Reidel, Boston, MA.

[3] M. Gromov and V. Milman (1983), *Brunn Theorem and a Concentration of Volume Phenomena for Symmetric Convex Bodies*, Geometric and Functional Analysis Seminar Notes, Tel Aviv University, Israel, 1983–1984.

[4] O. Güler (1996), *Barrier functions in interior point methods*, Math. Oper. Res., 21, pp. 860–885.

[5] O. Güler (1995), *Hyperbolic polynomials and interior–point methods for convex programming*, Math. Oper. Res., to appear.

[6] F. Jarre (1994), *A new line search step based on the Weierstrass $\wp$-function for minimizing a class of logarithmic barrier functions*, Numer. Math., 68, pp. 81–94.

[7] N. Karmarkar (1984), *A new polynomial–time algorithm for linear programming*, Combinatorica, 4, pp. 373–395.

[8] H. Knothe (1957), *Contributions to the theory of convex bodies*, Michigan Math. J., 4, pp. 39–52.

[9] D. Knuth (1968), *The Art of Computer Programming*, Addison–Wesley, Reading, MA.

[10] V. Milman and G. Schechtman (1986), *Asymptotic Theory of Finite Dimensional Normed Spaces*, in Lecture Notes in Mathematics 1200, Springer-Verlag, Berlin, New York.

[11] Yu. E. Nesterov and A. S. Nemirovskii (1994), *Interior Point Polynomial Methods in Convex Programming*, SIAM , Philadelphia, PA.

[12] Yu. E. Nesterov and A. S. Nemirovskii (1995), *Multiparameter Surfaces of Analytic Centers and Long–Step Surface–Following Interior Point Methods*, Optimization Laboratory Research report 3/95, Technion, Haifa, Israel.

[13] Yu. E. Nesterov and M. J. Todd (1994), *Self–scaled barriers and interior–point methods for convex programming*, Math. Oper. Res., to appear.

[14] Yu. E. Nesterov and M. J. Todd (1995), *Primal–Dual Interior Point Methods for Self–Scaled Cones*, Center for Operations Resarch and Econometrics Discussion paper 9462, Louvain–la–Neuve, Belgium.

SIAM J. OPTIM.
Vol. 7, No. 2, pp. 304–317, May 1997

A QUADRATICALLY CONVERGENT INFEASIBLE-INTERIOR-POINT ALGORITHM FOR LCP WITH POLYNOMIAL COMPLEXITY[*]

RONGQIN SHENG[†] AND FLORIAN A. POTRA[†]

Abstract. A predictor–corrector algorithm is proposed for solving monotone linear complementarity problems (LCPs) from infeasible starting points. The algorithm terminates in $O(nL)$ steps either by finding a solution or by determining that the problem has no solution of norm less than a given number. The complexity of the algorithm depends on the quality of the starting point. If the problem is solvable and if a certain measure of feasibility at the starting point is small enough, then the algorithm finds a solution in $O(\sqrt{n}L)$ iterations. The algorithm requires two matrix factorizations and two backsolves per iteration. If the problem has a strictly complementary solution, then the algorithm is quadratically convergent, and, therefore, its asymptotic efficiency index is $\sqrt{2}$.

Key words. linear complementarity problems, predictor–corrector, infeasible-interior-point algorithm, polynomiality, superlinear convergence

AMS subject classifications. 90C05, 90C33, 49M35, 49M40, 65K05

PII. S1052623494267826

1. Introduction.

The monotone linear complementarity problem (LCP) asks for the determination of a vector pair $(x, s) \in \mathbb{R}^{2n}$ which satisfies the conditions

$$s = Mx + q, \qquad x \geq 0, \qquad s \geq 0, \qquad x^T s = 0, \tag{1.1}$$

where $q \in \mathbb{R}^n$, and $M \in \mathbb{R}^{n\times n}$ is positive semidefinite.

Most interior-point methods for linear programming have been successfully extended to this problem. They require that the starting points satisfy exactly the equality constraints and are strictly positive; i.e., they lie in the interior of the region defined by the inequality constraints. All subsequent points generated by the interior-point algorithm will have the same properties. However, it may be very difficult to obtain feasible starting points in practice. Moreover, there are problems for which such points do not exist. In the latter category, we mention problems having unbounded primal or dual optimal sets and, of course, problems that are infeasible to start with. The existence of feasible interior starting points implies the existence of a solution, and, therefore, interior-point algorithms cannot be used to detect whether or not the problem is solvable.

Numerical experiments have shown that it is possible to obtain a good practical performance by using starting points that lie in the interior of the region defined by the inequality constraints but do not satisfy the equality constraints (cf. [3]). The points generated by the algorithm will remain in the interior of the region defined by the inequality constraints but will never exactly satisfy the equality constraints, although the measure of "feasibility" as well as "optimality" will improve at each step. This property is reflected in the name "infeasible-interior-point algorithm" which has been suggested for such methods. For problems that have a solution, both optimality and

[*] Received by the editors May 16, 1994; accepted for publication (in revised form) November 8, 1995. This work was supported in part by NSF grant DMS 9305760.

http://www.siam.org/journals/siopt/7-2/26782.html

[†] Department of Mathematics, The University of Iowa, Iowa City, IA 52242 (rsheng@math.uiowa.edu, potra@math.uiowa.edu).

feasibility can be achieved up to any desired accuracy. Moreover, infeasible-interior-point methods can be used to detect whether the problem has solutions in a given region (cf. [1]). For a recent survey of infeasible-interior-point algorithms for linear programming (LP) and LCP, we refer the reader to [8].

In the above cited paper [8], the second author proposed a new extension of the Mizuno–Todd–Ye algorithm [5] for solving monotone LCPs from infeasible starting points. Its computational complexity depends on the quality of the starting point. If the problem is solvable and if the starting points are large enough, then the algorithm has $O(nL)$-iteration complexity. If a certain measure of feasibility at the starting point is small enough, then the algorithm has $O(\sqrt{n}L)$-iteration complexity. At each iteration, both "feasibility" and "optimality" are reduced exactly at the same rate. The algorithm requires two matrix factorizations and, at most, three backsolves per iteration. It is quadratically convergent for problems having a strictly complementary solution. Therefore, its asymptotic index in the sense of Ostrowski [7] is $\sqrt{2}$. Moreover, the algorithm can be modified along the lines of the ideas of [1] so that it can detect if the problem has solutions of norm less than a given constant. For problems with integer data of length L we can theoretically take a constant of order $O(2^L)$, and then the algorithm will detect in a finite number of steps whether or not the problem is solvable. However, no polynomial complexity results have been obtained for determining the solvability of the LCP by infeasible-interior-point methods. This is in contrast with the situation in linear programming where Ye, Todd, and Mizuno [13] used a homogeneous self-dual approach to show that solvability can be detected in $O(\sqrt{n}L)$ iterations, which improved the $O(nL)$-iteration complexity result from [4].

In the present paper, we propose a predictor–corrector algorithm having the same convergence and complexity properties on solvable problems as the algorithm proposed in [8], but our algorithm requires two factorizations and only two backsolves per iteration. Moreover, the algorithm can detect nonexistence of a solution in $O(nL)$ iteration. More precisely, given a general monotone LCP, the algorithm terminates in at most $O(nL)$ steps either by finding a solution or by determining that the problem is not solvable. If the problem is solvable and if a certain measure of feasibility at the starting point is small enough, then the algorithm finds a solution in $O(\sqrt{n}L)$ iterations. If the problem has a strictly complementary solution, then the algorithm is quadratically convergent. To our knowledge these are the best complexity results for LCPs obtained so far in the literature. Mizuno, Kojima, and Todd [2] mentioned that the $O(nL)$ infeasible-interior-point algorithms for linear programming considered in that paper can be generalized for LCPs, but the superlinear convergence of the resulting algorithms has not yet been established. The infeasible-interior-point algorithms for LCP proposed by Zhang [14] and Wright [9, 10, 11] have only $O(n^2L)$-iteration complexity.

We note that the algorithm in [8] can also be modified so that it determines whether or not the problem has a solution of norm less than a given number in at most $O(nL)$ iterations. The algorithm to be presented in this paper has the advantage of using one backsolve less per iteration. However, while the algorithm in [8] improves optimality and feasibility at exactly the same rate, the new algorithm improves them "almost" at the same rate as shown in Theorem 2.4.

The notation used throughout the paper is rather standard: capital letters denote matrices, lowercase letters denote vectors, script capital letters denote sets, and Greek letters denote scalars. All vectors are considered column vectors. The components of a vector $u \in \mathbb{R}^n$ will be denoted by $[u]_i$ (and when there is no danger of confusion by u_i), $i = 1, \ldots, n$. The relation $u > 0$ is equivalent to $[u]_i > 0,\ i = 1, \ldots, n$, while $u \geq 0$ means $[u]_i \geq 0,\ i = 1, \ldots, n$. If $u \in \mathbb{R}^n,\ w \in \mathbb{R}^m$ then (u, w) denotes the column vector formed by the components of u and w, i.e., $(u, w) \in \mathbb{R}^{n+m},\ [(u, w)]_i = [u]_i$ for $1 \leq i \leq n$, and $[(u, w)]_{n+j} = [w]_j$ for $1 \leq j \leq m$. We denote $\mathbb{R}^n_+ = \{u \in \mathbb{R}^n\ :\ u \geq 0\}$, $\mathbb{R}^n_{++} = \{u \in \mathbb{R}^n\ :\ u > 0\}$. If $u \in \mathbb{R}^n$ then $U := \operatorname{diag}(u)$ denotes the diagonal matrix having the components of u as diagonal entries. The most used norm is the l_2-norm, so we write $\| \cdot \|$ instead of $\| \cdot \|_2$ both for vector norms and the corresponding matrix norms $\|A\| = \max\{\|Ax\|\ :\ \|x\| = 1\}$. Whenever we need other norms like $\| \cdot \|_1$ or $\| \cdot \|_\infty$, we use the corresponding symbol. In particular, if $X = \operatorname{diag}(x)$ then $\|X\| = \max\{|\ x_i\ |\ :\ i = 1, \ldots, n\} = \|x\|_\infty \neq \|x\| = (\sum_{i=1}^n x_i^2)^{1/2}$.

2. The predictor–corrector algorithm. We denote the feasible set of the problem (1.1) by

$$\mathcal{F} = \{(x, s) \in \mathbb{R}^{2n}_+\ :\ s = Mx + q\}$$

and its solution set by

$$\mathcal{F}^* = \{(x^*, s^*) \in \mathcal{F}\ :\ {x^*}^T s^* = 0\}.$$

It is easily seen that $(x^*, s^*) \in \mathcal{F}^*$ if and only if (x^*, s^*) is the solution of the following nonlinear system:

$$F(x, s) := \begin{pmatrix} Xs \\ Mx - s + q \end{pmatrix} = 0. \tag{2.1}$$

For any given $\epsilon > 0$, we define the set of *ϵ-approximate solutions* of (1.1) as

$$\mathcal{F}_\epsilon = \{(x,\ s) \in \mathbb{R}^{2n}_+\ :\ x^T s \leq \epsilon, \quad \|Mx - s + q\| \leq \epsilon\}.$$

In what follows, we will present an algorithm that finds a point in this set in a finite number of steps, provided our problem has a solution (i.e., $\mathcal{F}^*$ is not empty). The algorithm depends on two parameters α, β satisfying the inequalities

$$\frac{\beta^2}{\sqrt{8}(1 - \beta)} \leq \alpha < \beta < 1. \tag{2.2}$$

For example, $\alpha = 0.25,\ \beta = 0.5$ verify (2.2). The starting point of the algorithm can be any pair of strictly positive vectors $(x^0, s^0) \in \mathbb{R}^{2n}_{++}$, that is, (α, τ)-centered in the sense that it belongs to the following set:

$$\mathcal{N}_{\alpha,\tau} = \{(x, s) \in \mathbb{R}^{2n}_{++} : \|Xs - \tau e\| \leq \alpha\tau\},$$

where $\tau > 0$. Throughout this paper, we will denote $\mu = \frac{1}{n} x^T s$.

At a typical step of our algorithm, we are given a pair $(x, s) \in \mathcal{N}_{\alpha,\tau}$ and obtain a predictor direction (u, v) by solving the linear system

$$Su + Xv = -Xs, \tag{2.3a}$$

$$Mu - v = r, \tag{2.3b}$$

where r is the residual $r = s - Mx - q$. Notice that this is just the Newton direction for the nonlinear system (2.1), whose Jacobian

$$F'(x, s) := \begin{pmatrix} S & X \\ M & -I \end{pmatrix}$$

is nonsingular whenever $x > 0$ and $s > 0$. If we take a steplength θ along this direction, we obtain the points

$$x(\theta) = x + \theta u, \quad s(\theta) = s + \theta v.$$

We define $\overline{\theta}$ as the largest steplength for which

$$\|X(\theta)s(\theta) - (1-\theta)\tau e\| \leq \beta(1-\theta)\tau \quad \text{for all} \quad 0 \leq \theta \leq \overline{\theta} \tag{2.4}$$

and consider the predicted pair

$$\overline{x} = x + \overline{\theta}u, \quad \overline{s} = s + \overline{\theta}v. \tag{2.5}$$

We will see later that these are strictly positive vectors. Therefore, the Jacobian $F'(\overline{x}, \overline{s})$ is nonsingular and we can define the corrector direction $\overline{u}, \overline{v}$ as the solution of the following linear system:

$$\overline{S}\overline{u} + \overline{X}\overline{v} = (1-\overline{\theta})\tau e - \overline{X}\overline{s}, \tag{2.6a}$$

$$M\overline{u} - \overline{v} = 0. \tag{2.6b}$$

By taking a unit steplength along the corrector direction, we obtain a new pair:

$$x^+ = \overline{x} + \overline{u}, \quad s^+ = \overline{s} + \overline{v}. \tag{2.7}$$

Clearly,

$$r^+ = (1-\overline{\theta})r. \tag{2.8}$$

Correspondingly, we define

$$\tau^+ = (1-\overline{\theta})\tau. \tag{2.9}$$

In order to have a well-defined algorithm, we will show that $(x^+, s^+) \in \mathcal{N}_{\alpha,\tau^+}$ so that the above steps can be iterated with (x^+, s^+) and τ^+ instead of (x, s) and τ. In the proof we will use the following two technical lemmas. The first one is a slight modification of Lemma 2.1 of [8], while the second one corresponds to Corollary 2.3 of [8].

LEMMA 2.1. *If $(x, s) \in \mathcal{N}_{\alpha,\tau}$, then the largest number $\overline{\theta} \in [0,1]$ satisfying* (2.4) *is given by*

$$\overline{\theta} = 2/(1 + \sqrt{1 + 4/\varphi_1}), \tag{2.10}$$

$$\begin{aligned} \varphi_1 &= \alpha_0/(\alpha_1 + \sqrt{\alpha_1{}^2 + \alpha_0\delta^2}), \\ \delta &= \|g\|, \quad \alpha_0 = \beta^2 - \|f\|^2, \quad \alpha_1 = f^T g, \\ f &= \frac{1}{\tau}Xs - e, \quad g = \frac{1}{\tau}Uv, \end{aligned} \tag{2.11}$$

where u, v is the solution of the linear system (2.3). *Moreover, the pair $(\overline{x}, \overline{s})$ defined by* (2.5) *satisfies*

$$\|\overline{X}\overline{s} - (1-\overline{\theta})\tau e\| = \beta(1-\overline{\theta})\tau, \quad \overline{x} > 0, \quad \overline{s} > 0.$$

Lemma 2.2. *Let x, s, a, b be four n-dimensional vectors with $x > 0$ and $s > 0$, and let $M \in \mathbb{R}^{n\times n}$ be a positive semidefinite matrix. Then the solution (u, v) of the linear system*

$$\begin{aligned} Su + Xv &= a,\\ Mu - v &= b \end{aligned}$$

satisfies the following relations:

$$\begin{aligned} Du = (I+\widetilde{M})^{-1}\widetilde{c}, \quad D^{-1}v &= \widetilde{a} - Du,\\ \|Du\| &\le \|\widetilde{c}\|,\\ \|Du\|^2 + \|D^{-1}v\|^2 &\le \|\widetilde{a}\|^2 + 2\|\widetilde{b}\|\|\widetilde{c}\|,\\ \|Uv\|^2 &\le \frac{1}{8}\|\widetilde{a}\|^4 + \frac{1}{2}\|\widetilde{b}\|\|\widetilde{c}\|(\|\widetilde{a}\|^2 + 2\|\widetilde{b}\|\|\widetilde{c}\|), \end{aligned} \tag{2.12}$$

where

$$D = X^{-1/2}S^{1/2}, \quad \widetilde{M} = D^{-1}MD^{-1}, \tag{2.13a}$$

$$\widetilde{a} = (XS)^{-1/2}a, \quad \widetilde{b} = D^{-1}b, \quad \widetilde{c} = \widetilde{a} + \widetilde{b}. \tag{2.13b}$$

Now we are ready to prove that the algorithm described in this section is well defined. For ease of later reference, let us first formally define our algorithm.

Algorithm 2.3. *Choose $(x^0, s^0) \in \mathcal{N}_{\alpha,\tau_0}$ with $\tau_0 = \frac{(x^0)^T s^0}{n(1+\alpha/\sqrt{n})} = \frac{\mu_0}{1+\alpha/\sqrt{n}}$ and set $\psi_0 = 1$, $\epsilon > 0$. For $k = 0, 1, \ldots,$ do* A1 *through* A5*:*

A1 *Set $x = x^k$, $s = s^k$, $\tau = \tau_k$ and define $\mu = (x^T s)/n$, $r = s - Mx - q$, $\psi = \psi_k$.*

A2 *If $x^T s \le \epsilon$, and $\|r\| \le \epsilon$ then report $(x, s) \in \mathcal{F}_\epsilon$ and terminate.*

A3 *Find the solution u, v of the linear system* (2.3)*, define $\bar{x}, \bar{s}$ as in* (2.5)*, and set $\psi_+ = (1-\bar{\theta})\psi$, where $\bar{\theta}$ is given by* (2.10)*.*

A4 *Find the solution $\bar{u}, \bar{v}$ of the linear system* (2.6) *and define x^+, s^+, τ^+ as in* (2.7) *and* (2.9)*.*

A5 *Set $x^{k+1} = x^+$, $s^{k+1} = s^+$, $\tau_{k+1} = \tau^+$, $\bar{\theta}_k = \bar{\theta}$, $\mu_k = \mu$, $r^k = r$, $\psi_{k+1} = \psi_+$.*

Before stating our main result let us note that the standard choice of starting points

$$x^0 = \rho_p e, \quad s^0 = \rho_d e, \quad \rho_p,\ \rho_d \in \mathbb{R}_{++}$$

gives

$$\tau_0 = \frac{\mu_0}{1+\alpha/\sqrt{n}} = \frac{\rho_p\rho_d}{1+\alpha/\sqrt{n}}$$

and

$$\|X^0 s^0 - \tau_0 e\| = \frac{\rho_p\rho_d\alpha}{1+\alpha/\sqrt{n}} = \alpha\tau_0, \tag{2.14}$$

which shows that $(x^0, s^0) \in \mathcal{N}_{\alpha,\tau_0}$, as required in the algorithm.

Theorem 2.4. *For any integer $k \ge 0$, Algorithm* 2.3 *defines a pair*

$$(x^k, s^k) \in \mathcal{N}_{\alpha,\tau_k}, \tag{2.15}$$

and the corresponding residuals satisfy

$$r^k = \psi_k r^0, \quad \tau_k = \psi_k \tau_0, \tag{2.16}$$

$$\tau_k \leq \mu_k \leq \psi_k \mu_0 = (1 + \alpha/\sqrt{n})\tau_k, \tag{2.17}$$

where

$$\psi_0 = 1, \quad \psi_k = \prod_{i=0}^{k-1} (1 - \overline{\theta}_i). \tag{2.18}$$

Proof. The proof is by induction. For $k = 0$, (2.15), (2.16), and (2.17) are clearly satisfied. Suppose they are satisfied for some $k \geq 0$. As in Algorithm 2.3 we will omit the index k. Therefore, we can write

$$(x, s) \in \mathcal{N}_{\alpha,\tau}, \quad r = \psi r^0, \quad \tau = \psi\tau_0, \quad \tau \leq \mu \leq \psi\mu_0.$$

The fact that (2.16) holds for $k + 1$ follows immediately from (2.8) and (2.9). From (2.6) and (2.7) we have

$$X^+ s^+ = (1 - \overline{\theta})\tau e + \overline{U}\overline{v}, \quad \mu^+ = \frac{1}{n}(x^+)^T s^+ = (1 - \overline{\theta})\tau + \frac{1}{n}\overline{u}^T \overline{v}. \tag{2.19}$$

By using (2.6), (2.10), (2.19), and Lemma 2.2 with $b = 0$ and $\overline{x}, \overline{s}$ instead of x, s, we deduce that

$$\|\overline{U}\overline{v}\| \leq \frac{1}{\sqrt{8}} \|(\overline{X}\,\overline{S})^{-1}\| \|\overline{X}\overline{s} - (1 - \overline{\theta})\tau e\|^2 \leq \frac{\beta^2(1 - \overline{\theta})\tau}{\sqrt{8}(1 - \beta)}. \tag{2.20}$$

On the other hand, by using (2.2), (2.19), and (2.20) we can write

$$\|X^+ s^+ - \tau^+ e\| = \|\overline{U}\overline{v}\| \leq \frac{(1 - \overline{\theta})\beta^2\tau}{\sqrt{8}(1 - \beta)} \leq \alpha\tau^+. \tag{2.21}$$

The positivity of x^+ and s^+ is proved by contradiction. Suppose, for example, that $[x^+]_i \leq 0$ for some i. Since (2.21) implies $[x^+]_i[s^+]_i > 0$, we must have $[x^+]_i < 0$ and $[s^+]_i < 0$. It follows that $[\overline{u}]_i < -[\overline{x}]_i$ and $[\overline{v}]_i < -[\overline{s}]_i$. By virtue of (2.7), we get

$$-[\overline{x}]_i[\overline{s}]_i < (1 - \overline{\theta})\tau - [\overline{x}]_i[\overline{s}]_i = [\overline{s}]_i[\overline{u}]_i + [\overline{x}]_i[\overline{v}]_i < -2[\overline{x}]_i[\overline{s}]_i,$$

which is a contradiction. Hence, (2.15) is satisfied for $k+1$. Using (2.6b), the positive semidefiniteness of M, and (2.21) it follows that

$$0 \leq \overline{u}^T \overline{v} \leq \sqrt{n}\|\overline{U}\overline{v}\| \leq \sqrt{n}\alpha\tau^+.$$

By substituting the above inequalities in (2.19) we obtain

$$\tau^+ \leq \mu^+ \leq (1 - \overline{\theta})\tau(1 + \alpha/\sqrt{n}) = \tau^+(1 + \alpha/\sqrt{n}) = \psi_+\tau_0(1 + \alpha/\sqrt{n}) = \psi_+\mu_0.$$

Hence, (2.17) is also satisfied and the proof of our theorem is complete. □

3. Global convergence and polynomial complexity. In what follows, we assume that $\mathcal{F}^*$ is nonempty. Under this assumption we will prove that Algorithm 2.3, with $\epsilon = 0$, is globally convergent in the sense that if (2.15) is satisfied, then

$$\lim_{k\to\infty} \mu_k = 0, \quad \lim_{k\to\infty} r^k = 0.$$

LEMMA 3.1. *If $\mathcal{F}^*$ is nonempty, then the sequence (x^k, s^k) generated by Algorithm 2.3 satisfies*

$$\psi_k((x^0)^T s^k + (s^0)^T x^k) \leq (1 + \alpha/\sqrt{n})(2 + \zeta) n\tau_k, \; k = 0, 1, 2, \dots, \tag{3.1}$$

where

$$\zeta = \inf \left\{ ((x^0)^T s^* + (s^0)^T x^*)/((x^0)^T s^0) : (x^*, s^*) \in \mathcal{F}^* \right\}. \tag{3.2}$$

Proof. Let $(x^*, s^*) \in \mathcal{F}^*$. By writing x, s, ψ for x^k, s^k, ψ_k, respectively, and by using the fact that $r = \psi r^0$, we have

$$\begin{aligned}&\psi s^0 + (1 - \psi)s^* - s = \psi(s^0 - s^*) - (s - s^*)\\ &= \psi(r^0 + M(x^0 - x^*)) - (r + M(x - x^*)) = M(\psi x^0 + (1 - \psi)x^* - x).\end{aligned}$$

Since M is positive semidefinite, we obtain

$$\begin{aligned}0 &\leq [\psi x^0 + (1 - \psi)x^* - x]^T [\psi s^0 + (1 - \psi)s^* - s]\\ &= \psi^2 n\mu_0 + \psi(1 - \psi)((x^0)^T s^* + (s^0)^T x^*)\\ &\quad - \psi((x^0)^T s + (s^0)^T x) + x^T s - (1 - \psi)(s^T x^* + x^T s^*) + (1 - \psi)^2 (x^*)^T s^*,\end{aligned} \tag{3.3}$$

and the desired inequality (3.1) follows by using the relations $\tau \leq \mu \leq (1 + \alpha/\sqrt{n})\tau$, $(x^*)^T s^* = 0$, $s^T x^* + x^T s^* \geq 0$. □

From the above lemma and Lemma 2.2 we will derive a useful bound for the quantities

$$\delta_k = \|U^k v^k\|/\tau_k, \quad k \geq 0, \tag{3.4}$$

where (u^k, v^k) is obtained at step A3 of Algorithm 2.3. This bound is going to play an important role in our analysis.

LEMMA 3.2. *Let (u^k, v^k) be obtained in the kth iteration at step* A3 *of Algorithm 2.3 and let δ_k be defined by* (3.4). *Then*

$$\delta_k \leq \delta^* = n(1 + \alpha/\sqrt{n})\sqrt{.125 + \eta(1 + \eta)(.5 + \eta(1 + \eta))},$$

where

$$\eta = \sqrt{n}(2 + \zeta)\|(S^0)^{-1} r^0\|_\infty \sqrt{\frac{1 + \alpha/\sqrt{n}}{1 - \alpha}},$$

with ζ given by (3.2).

Proof. We omit the index k and apply Lemma 2.2 with $a = -Xs$ and $b = r$. We have immediately that

$$\|\widetilde{a}\| = \|(XS)^{1/2} e\| = \sqrt{n\mu} \leq \sqrt{(1 + \alpha/\sqrt{n})n\tau}. \tag{3.5}$$

In order to obtain a bound for $\|\widetilde{b}\|$, we write

$$\begin{aligned}\|\widetilde{b}\| &= \|D^{-1}r\| = \|(XS)^{-1/2}Xr\| \leq (1-\alpha)^{-1/2}\tau^{-1/2}\|Xr\| \\ &\leq (1-\alpha)^{-1/2}\tau^{-1/2}\|Xr\|_1 = \psi(1-\alpha)^{-1/2}\tau^{-1/2}\|Xr^0\|_1 \\ &= \psi(1-\alpha)^{-1/2}\tau^{-1/2}\sum_{i=1}^{n}[x]_i[s^0]_i|[r^0]_i|/[s^0]_i \\ &\leq (1-\alpha)^{-1/2}\tau^{-1/2}\|(S^0)^{-1}r^0\|_\infty\psi((s^0)^Tx).\end{aligned}$$

Using Lemma 3.1 and the notation introduced in the statement of Lemma 3.2, we obtain

$$\|\widetilde{b}\| \leq \eta\sqrt{(1+\alpha/\sqrt{n})n\tau}, \tag{3.6}$$

and by using the triangle inequality we get

$$\|\widetilde{c}\| \leq (1+\eta)\sqrt{(1+\alpha/\sqrt{n})n\tau}. \tag{3.7}$$

Finally, the required inequality follows by substituting (3.5), (3.6), and (3.7) in equation (2.12). □

By using the above lemma we can easily prove the following result.

LEMMA 3.3. *Let $\mathcal{F}^*$ be nonempty and consider the notation introduced in Lemma* 3.2. *Then the steplength $\bar{\theta}_k$ of Algorithm* 2.3 *satisfies*

$$\bar{\theta}_k \geq \theta^* = \frac{2}{1+\sqrt{1+4\delta^*/(\beta-\alpha)}}, \quad k \geq 0. \tag{3.8}$$

Proof. According to (2.11), we have

$$\frac{1}{\varphi_1} \leq \left(|\alpha_1| + \sqrt{|\alpha_1|^2+\alpha_0\delta^2}\right)/\alpha_0.$$

The right-hand side of the above inequality is increasing in $|\alpha_1|$ and decreasing in α_0, so by using the obvious inequalities $\alpha_0 \geq \beta^2-\alpha^2$ and $|\alpha_1| \leq \|f\|\|g\| \leq \alpha\delta$ we obtain

$$\frac{1}{\varphi_1} \leq (\alpha\delta+\sqrt{(\alpha\delta)^2+(\beta^2-\alpha^2)\delta^2})/(\beta^2-\alpha^2) = \delta/(\beta-\alpha). \tag{3.9}$$

Then, (3.8) follows from Lemma 3.2 and (2.10). □

With the help of the above lemma and Theorem 2.4 we can easily prove the main result of this section, which basically states that Algorithm 2.3 is globally convergent at a linear rate.

THEOREM 3.4. *Suppose that the solution set $\mathcal{F}^*$ is nonempty.*

(i) *If $\epsilon = 0$ then Algorithm* 2.3 *either finds an optimal solution $z^* \in \mathcal{F}^*$ in a finite number of steps or produces an infinite sequence $z^k = (x^k, s^k)$ such that*

$$\lim_{k\to\infty}(x^k)^Ts^k = 0, \ \lim_{k\to\infty}r^k = 0.$$

(ii) *If $\epsilon > 0$ then Algorithm* 2.3 *terminates with a $z \in \mathcal{F}_\epsilon$ in at most*

$$K_\epsilon = \left\lceil \frac{|\ln(\frac{\epsilon}{\epsilon_0})|}{|\ln(1-\theta^*)|} \right\rceil$$

iterations, where $\epsilon_0 = \max\{(x^0)^T s^0, \|r^0\|\}$ *and* $\lceil \chi \rceil$ *denotes the smallest integer greater than or equal to* χ.

From the above theorem we can obtain polynomial complexity under certain assumptions on the starting point. First, we show that if the starting point is feasible, or close to being feasible, then the algorithm has $O(\sqrt{n}\ln(\epsilon/\epsilon_0))$-iteration complexity.

COROLLARY 3.5. *Assume that* $\mathcal{F}^*$ *is nonempty and that the starting point is chosen such that there is a constant* κ *independent of* n *satisfying the inequality*

$$(2+\zeta)\|(S^0)^{-1}r^0\|_\infty \le n^{-1/2}\kappa,$$

where ζ *is defined by* (3.2). *Then Algorithm* 2.3 *terminates in at most*

$$\tilde{K}_\epsilon = O(\sqrt{n}\ln(\epsilon_0/\epsilon))$$

iterations.

Most of the complexity results on infeasible-interior-point methods are obtained for starting points of the form

$$x^0 = \rho_p e, \quad s^0 = \rho_d e, \tag{3.10}$$

where ρ_p and ρ_d are sufficiently large positive constants (big M initialization). For such starting points, as shown by (2.14), we have $(x^0, s^0) \in \mathcal{N}_{\alpha,\tau_0}$, and

$$\zeta = \inf\left\{\|x^*\|_1/(n\rho_p) + \|s^*\|_1/(n\rho_d) : (x^*, s^*) \in \mathcal{F}^*\right\},$$

$$\|(S^0)^{-1}r^0\|_\infty \le 1 + (\rho_p/\rho_d)\|Me\|_\infty + (1/\rho_d)\|q\|_\infty.$$

Therefore, if ρ_p and ρ_d satisfy the inequalities

$$\rho_p \ge n^{-1}\|x^*\|_1, \tag{3.11}$$

$$\rho_d \ge \max\{\rho_p\|Me\|_\infty, \|q\|_\infty, n^{-1}\|s^*\|_1\} \tag{3.12}$$

for some $(x^*, s^*) \in \mathcal{F}^*$, then $\eta < 36\sqrt{n}$, and we obtain the following complexity result.

COROLLARY 3.6. *Assume that* $\mathcal{F}^*$ *is nonempty and that the starting point is chosen from the form* (3.10) *such that* (3.11) *and* (3.12) *are satisfied for some* $(x^*, s^*) \in \mathcal{F}^*$. *Then, Algorithm* 2.3 *terminates in at most*

$$\tilde{K}_\epsilon = O(n\ln(\epsilon_0/\epsilon))$$

iterations.

All the above results have been proved under the assumption that $\mathcal{F}^*$ is nonempty. It turns out that Algorithm 2.3 can be modified in such a way that it can detect within polynomial time whether $\mathcal{F}^*$ contains points of norm less than a quantity chosen in advance. Let $\overline{\rho}_p$ and $\overline{\rho}_d$ be such quantities and define

$$\overline{\zeta} = (\|x^0\|\overline{\rho}_d + \|s^0\|\overline{\rho}_p)/((x^0)^T s^0),$$

$$\overline{\eta} = \sqrt{n}(2+\overline{\zeta})\|(S^0)^{-1}r^0\|_\infty \sqrt{\frac{1+\alpha/\sqrt{n}}{1-\alpha}}, \tag{3.13}$$

$$\overline{\delta^*} = n(1+\alpha/\sqrt{n})\sqrt{.125+\overline{\eta}(1+\overline{\eta})(.5+\overline{\eta}(1+\overline{\eta}))},$$

$$\overline{\theta^*} = 2/\left(1+\sqrt{1+4\overline{\delta^*}/(\beta-\alpha)}\right).$$

Now we can prove the following theorem.

THEOREM 3.7. *Suppose that the instruction "A2.5 If*

$$(x^0)^T s^k + (s^0)^T x^k > (x^0)^T s^0 \tau_k/\tau_0 + (x^k)^T s^k \tau_0/\tau_k + (1-\tau_k/\tau_0)(\overline{\rho}_d\|x^0\| + \overline{\rho}_p\|s^0\|)$$

then terminate" is inserted in between instructions A2 *and* A3 *of Algorithm* 2.3. *Then, the new algorithm terminates either at* A2 *with* $z \in \mathcal{F}_\epsilon$ *or at* A2.5, *both in at most*

$$\overline{K}_\epsilon = \left\lceil \frac{|\ln(\epsilon/\epsilon_0)|}{|\ln(1-\overline{\theta^*})|} \right\rceil$$

iterations, and in the latter case there is no $z^* = (x^*, s^*) \in \mathcal{F}^*$ *such that* $\|x^*\| \le \overline{\rho}_p$, $\|s^*\| \le \overline{\rho}_d$.

Proof. Suppose that the inequality

$$\begin{aligned}(x^0)^T s^k + (s^0)^T x^k \\ \le (x^0)^T s^0 \tau_k/\tau_0 + (x^k)^T s^k \tau_0/\tau_k + (1-\tau_k/\tau_0)(\overline{\rho}_d\|x^0\| + \overline{\rho}_p\|s^0\|)\end{aligned} \tag{3.14}$$

holds for all $0 \le k \le \overline{K}_\epsilon$. Then we have

$$\psi_k((x^0)^T s^k + (s^0)^T x^k) \le (1+\alpha/\sqrt{n})(2+\overline{\zeta})n\tau_k, \qquad 0 \le k \le \overline{K}_\epsilon. \tag{3.15}$$

With the help of (3.15), we can prove, as in Lemma 3.2, that $\delta_k \le \overline{\delta^*}$. Hence, $\overline{\theta}_k \ge \overline{\theta^*}$ for all $0 \le k \le \overline{K}_\epsilon$, which implies $(x^{\overline{K}_\epsilon}, s^{\overline{K}_\epsilon}) \in \mathcal{F}_\epsilon$.

On the other hand, if there exists $(x^*, s^*) \in \mathcal{F}^*$ such that $\|x^*\| \le \overline{\rho}_p$, $\|s^*\| \le \overline{\rho}_d$, then by (3.3), (3.14) must hold for all $k \ge 0$. Hence, if (3.14) is violated for some $k \le \overline{K}_\epsilon$, then there is no $(x^*, s^*) \in \mathcal{F}^*$ such that $\|x^*\| \le \overline{\rho}_p$, $\|s^*\| \le \overline{\rho}_d$. This completes the proof of our theorem. □

From the above theorem we can obtain polynomial complexity for our new algorithm under certain assumptions on the starting point. Let us choose

$$x^0 = \hat{\rho}_p e, \quad s^0 = \hat{\rho}_d e, \tag{3.16}$$

where

$$\hat{\rho}_p \ge \overline{\rho}_p/\sqrt{n}, \tag{3.17}$$

$$\hat{\rho}_d \ge \max\{\hat{\rho}_p\|Me\|_\infty, \|q\|_\infty, \overline{\rho}_d/\sqrt{n}\}. \tag{3.18}$$

Then we obtain

$$\overline{\zeta} \le 2, \tag{3.19}$$

$$\|(S^0)^{-1}r^0\|_\infty \le 1 + (\hat{\rho}_p/\hat{\rho}_d)\|Me\|_\infty + (1/\hat{\rho}_d)\|q\|_\infty \le 3. \tag{3.20}$$

From (3.13), (3.19), and (3.20), it follows that $\overline{\eta} < 36\sqrt{n}$. Hence, we have the following complexity result.

COROLLARY 3.8. *Suppose that the instruction "A2.5 If*

$$(x^0)^T s^k + (s^0)^T x^k > (x^0)^T s^0 \tau_k/\tau_0 + (x^k)^T s^k \tau_0/\tau_k + (1 - \tau_k/\tau_0)(\overline{\rho_d}\|x^0\| + \overline{\rho_p}\|s^0\|)$$

then terminate" is inserted in between instructions A2 *and* A3 *of Algorithm* 2.3 *and that the starting point is chosen from the form* (3.16) *such that* (3.17) *and* (3.18) *are satisfied. Then, the new algorithm terminates either at* A2 *with* $z \in \mathcal{F}_\epsilon$ *or at* A2.5, *both in at most*

$$\hat{K}_\epsilon = O(n \ln(\epsilon_0/\epsilon))$$

iterations, and in the latter case there is no $z^* = (x^*, s^*) \in \mathcal{F}^*$ *such that* $\|x^*\| \le \overline{\rho_p}$, $\|s^*\| \le \overline{\rho_d}$.

4. Quadratic convergence. In the previous section, we proved that Algorithm 2.3 is globally Q-linearly convergent under very general assumptions. Polynomial complexity was obtained under some additional assumptions on the starting points. In the present section, we will study the asymptotic convergence properties of Algorithm 2.3 in case (1.1) has a strictly complementary solution. Let us denote by $\mathcal{F}^c$ the set of all such solutions, i.e.,

$$\mathcal{F}^c = \left\{(x, s) \in \mathcal{F}^* : [x]_i + [s]_i > 0, i = 1, 2, \ldots, n\right\}.$$

It is well known that there is a unique partition

$$\mathcal{B} \cup \mathcal{N} = \{1, 2, \ldots, n\},\ \mathcal{B} \cap \mathcal{N} = \emptyset$$

such that for any $(x, s) \in \mathcal{F}^c$, we have $[x]_i > 0,\ [s]_i = 0$ for all $i \in \mathcal{B}$ and $[x]_i = 0,\ [s]_i > 0$ for all $i \in \mathcal{N}$. This means that the "basic" and "nonbasic" variables are invariant for any strictly complementary solution. We denote the corresponding partition of M by

$$M = \begin{pmatrix} M_{BB} & M_{BN} \\ M_{NB} & M_{NN} \end{pmatrix}.$$

Also, for any vector $y \in \mathbb{R}^n$, we denote by y_B the vector of components $[y]_i, i \in \mathcal{B}$ and by y_N the vector of components $[y]_i, i \in \mathcal{N}$. In the next lemma we show that all of the components of the vectors x_B and s_N are bounded from below.

LEMMA 4.1. *Let* $(x^*, s^*) \in \mathcal{F}^c$ *and denote*

$$\xi_p^* = \min\left\{[x^*]_i : i \in \mathcal{B}\right\}, \quad \xi_d^* = \min\left\{[s^*]_i : i \in \mathcal{N}\right\}, \quad \xi^* = \min\left\{\xi_p^*, \xi_d^*\right\},$$

$$\hat{\xi} = \frac{(1-\alpha)\xi^*}{n(1+\alpha)^2((2-\bar{\theta}_0)/\bar{\theta}_0 + \zeta^*)},$$

where

$$\zeta^* = ((x^0)^T s^* + (s^0)^T x^*)/((x^0)^T s^0).$$

Then the points x^k, s^k *generated by Algorithm* 2.3 *satisfy the inequality*

$$x_B^k \ge \hat{\xi} e, \ \ s_N^k \ge \hat{\xi} e, \ \ k \ge 1. \tag{4.1}$$

Proof. We drop the index k and use the notation from the proof of Lemma 3.1. According to (3.3), (2.16), and (2.18), we have

$$\begin{aligned} s^T x^* + x^T s^* &\le \frac{\psi^2}{1-\psi} n\mu_0 + \psi((x^0)^T s^* + (s^0)^T x^*) + \frac{x^T s}{1-\psi} \\ &\le (1+\alpha)\psi \left(\frac{1+\psi}{1-\psi} + \zeta^* \right) n\tau_0 \\ &\le (1+\alpha) \left(\frac{1+\psi_1}{1-\psi_1} + \zeta^* \right) x^T s \le (1+\alpha)((2-\bar{\theta}_0)/\bar{\theta}_0 + \zeta^*) x^T s \\ &\equiv \hat{\omega} x^T s, \end{aligned}$$

but then

$$\sum_{i\in\mathcal{B}} [x^*]_i [s]_i + \sum_{i\in\mathcal{N}} [s^*]_i [x]_i \le \hat{\omega} x^T s.$$

Because $(x, s) \in \mathcal{N}_{\alpha,\tau}$, it follows that

$$\frac{[x^*]_i}{[x]_i}(1-\alpha)\tau < \frac{[x^*]_i}{[x]_i}[x]_i[s]_i = [x^*]_i[s]_i \le \hat{\omega} x^T s \le \hat{\omega}(1+\alpha)n\tau \text{ for all } i \in \mathcal{B}.$$

Hence,

$$[x]_i \ge \frac{(1-\alpha)[x^*]_i}{(1+\alpha)n\hat{\omega}} \ge \frac{(1-\alpha)\xi_p^*}{(1+\alpha)n\hat{\omega}} \ge \hat{\xi} \text{ for all } i \in \mathcal{B},$$

which proves the first inequality in (4.1). The second inequality can be proved in a similar manner. □

In order to prove quadratic convergence, we will show that the predictor direction of Algorithm 2.3 satisfies $u^k = O(\tau_k), v^k = O(\tau_k)$. In the proof we will use the following two technical lemmas. The first one is a particular case of Lemma 5.2 of Wright [10] and was first given in the feasible case by Ye and Anstreicher [12]. The second one is a particular case of Lemma 2.2 of Monteiro and Wright [6].

LEMMA 4.2. *If* u, v *is the solution of the linear system* (2.3), *then the vector pair* $(u_B,\ v_N)$ *solves the convex quadratic program*

$$\min_{(w,z)} \frac{1}{2}\|D_B w\|^2 + \frac{1}{2}\|D_N^{-1} z\|^2$$

subject to

$$\begin{aligned} M_{BB} w &= r_B - M_{BN} u_N + v_B, \\ M_{NB} w - z &= r_N - M_{NN} u_N. \end{aligned}$$

LEMMA 4.3. *For any matrix* $H \in \mathbb{R}^{p\times q}$, *there exists a nonnegative constant* $\lambda = \lambda(H)$ *with the property that for any diagonal matrix* $T > 0$ *and any vector* $h \in \operatorname{Range}(H)$, *the (unique) optimal solution* $\overline{w} = \overline{w}(H, T, h)$ *of*

$$\min_w \frac{1}{2}\|Tw\|^2, \quad \textit{subject to } Hw = h,$$

satisfies

$$\|\overline{w}\|_\infty \le \lambda \|h\|_\infty.$$

The proof of the main result of this section is based on the following lemma.

LEMMA 4.4. *There is a constant σ independent of k such that if u^k, v^k are the vectors produced in substep* A3 *of Algorithm* 2.3*, then*

$$\|u^k\| \leq \sigma\tau_k, \quad \|v^k\| \leq \sigma\tau_k, \quad k \geq 0. \tag{4.2}$$

Proof. For simplicity, we again drop the index k and use the notation of Algorithm 2.3 as well as (2.13). From Theorem 2.4 we have $(x, s) \in \mathcal{N}_{\alpha,\tau}$, so by using (4.1) we can write

$$\|D_B\| \leq \|X_B^{-1}\|\|(XS)^{\frac{1}{2}}\| \leq \frac{(1+\alpha)\tau^{\frac{1}{2}}}{\hat{\xi}}.$$

A similar inequality can be obtained for $\|D_N^{-1}\|$. Therefore, we have

$$\|D_B\| \leq \sigma_1\tau^{\frac{1}{2}}, \quad \|D_N^{-1}\| \leq \sigma_1\tau^{\frac{1}{2}}, \tag{4.3}$$

where $\sigma_1 = (1+\alpha)/\hat{\xi}$. From (2.12), (3.5), (3.6), and (3.7) it follows that there is a constant σ_2 independent of k such that

$$\|Du\| \leq \sigma_2\tau^{\frac{1}{2}}, \quad \|D^{-1}v\| \leq \sigma_2\tau^{\frac{1}{2}}. \tag{4.4}$$

Inequalities (4.3) and (4.4) imply

$$\|u_N\| \leq \sigma_1\sigma_2\tau \equiv \sigma_3\tau, \quad \|v_B\| \leq \sigma_3\tau. \tag{4.5}$$

In order to complete the proof of our lemma, we have to show that there is a constant $\sigma_4 > 0$, independent of k, such that

$$\|u_B\| \leq \sigma_4\tau, \quad \|v_N\| \leq \sigma_4\tau.$$

From Lemmas 4.2 and 4.3 it follows that

$$\begin{aligned}\|(u_B, v_N)\| &\leq \sqrt{n}\|(u_B, v_N)\|_\infty \leq \sqrt{n}\lambda\|(r_B - M_{BN}u_N + v_B, r_N - M_{NN}u_N)\| \\ &\leq \sqrt{n}\lambda(\|(-M_{BN}u_N + v_B, -M_{NN}u_N)\| + \|r\|).\end{aligned}$$

Consequently, (4.2) follows from the above inequality, (4.5), and the fact that $\|r\| = \|\psi r^0\| = (\|r^0\|/\tau_0)\tau$. □

We end the paper by stating and proving our quadratic convergence result.

THEOREM 4.5. *If the LCP* (1.1) *has a strictly complementarity solution, then there are two constants γ and $\overline{\gamma}$ independent of k such that the points produced by Algorithm* 2.3 *satisfy*

$$\mu_{k+1} \leq \gamma{\mu_k}^2, \quad \|r^{k+1}\| \leq \overline{\gamma}{\|r^k\|}^2, \quad k \geq 1. \tag{4.6}$$

Proof. From (2.10), (3.9), and (4.2), it follows that

$$\overline{\theta}_k \geq 1 - \delta/(\beta - \alpha) \geq 1 - \hat{\gamma}\tau,$$

with $\hat{\gamma} = \sigma^2/(\beta - \alpha)$. Hence, $\tau_{k+1} \leq \hat{\gamma}\tau_k^2$. Then, from Theorem 2.4, we deduce that (4.6) holds with $\gamma = (1+\alpha)\hat{\gamma}$ and $\overline{\gamma} = \hat{\gamma}\tau_0/\|r^0\|$. □

Acknowledgments. The authors would like to thank Marco Menezes and Yinyu Ye for carefully reading the manuscript.

REFERENCES

[1] M. Kojima, N. Megiddo, and S. Mizuno, *A primal–dual infeasible-interior-point algorithm for linear programming*, Math. Programming, 61 (1993), pp. 263–280.

[2] S. Mizuno, M. Kojima, and M. J. Todd, *Infeasible-interior-point primal–dual potential-reduction algorithms for linear programming*, SIAM J. Optim., 5 (1995), pp. 52–67.

[3] I. J. Lustig, R. E. Marsten, and D. F. Shanno, *Computational experience with a globally convergent primal–dual predictor–corrector algorithm for linear programming*, Math. Programming, 66 (1994), pp. 123–135.

[4] S. Mizuno, *Polynomiality of infeasible–interior–point–algorithms for linear programming*, Math. Programming, 67 (1994), pp. 109–120.

[5] 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.

[6] R. D. C. Monteiro and S. J. Wright, *Superlinear primal-dual affine scaling algorithms for LCP*, Math. Programming, 69 (1995), pp. 311–333.

[7] A. M. Ostrowski, *Solution of Equations in Euclidean and Banach Spaces*, Academic Press, New York, 1973.

[8] F. A. Potra, *An $O(nL)$ Infeasible-Interior-Point Algorithm for LCP with Quadratic Convergence*, Reports on Computational Mathematics 50, Department of Mathematics, The University of Iowa, Iowa City, IA, January, 1994.

[9] S. J. Wright, *A path–following infeasible–interior–point algorithm for linear complementarity problems*, Optim. Methods Software, 2 (1993), pp. 79–106.

[10] S. J. Wright, *An infeasible interior point algorithm for linear complementarity problems*, Math. Programming, 67 (1994), pp. 29–52.

[11] S. J. Wright, *A path–following interior–point algorithm for linear and quadratic problems*, Ann. Oper. Res., 62 (1996), pp. 103–130.

[12] Y. Ye and K. Anstreicher, *On quadratic and $O(\sqrt{n}L)$ convergence of predictor-corrector algorithm for LCP*, Math. Programming, 62 (1993), pp. 537–551.

[13] 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.

[14] 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. 7, No. 2, pp. 318–335, May 1997

A LARGE-STEP INFEASIBLE-INTERIOR-POINT METHOD FOR THE P_*-MATRIX LCP*

FLORIAN A. POTRA† AND RONGQIN SHENG†

Abstract. A large-step infeasible-interior-point method is proposed for solving $P_*(\kappa)$-matrix linear complementarity problems. It is new even for monotone LCP. The algorithm generates points in a large neighborhood of an infeasible central path. Each iteration requires only one matrix factorization. If the problem is solvable, then the algorithm converges from arbitrary positive starting points. The computational complexity of the algorithm depends on the quality of the starting point. If a well-centered starting point is feasible or close to being feasible, then it has $O((1+\kappa)\sqrt{n}\ln(\epsilon_0/\epsilon))$-iteration complexity. With appropriate initialization, a modified version of the algorithm terminates in $O((1+\kappa)^2 n\ln(\epsilon_0/\epsilon))$ steps either by finding a solution or by determining that the problem is not solvable. High-order local convergence is proved for problems having a strictly complementary solution. We note that while the properties of the algorithm (e.g., computational complexity) depend on κ, the algorithm itself does not.

Key words. linear complementarity problems, P_*-matrices, infeasible-interior-point algorithm, polynomiality, superlinear convergence

AMS subject classifications. 90C05, 90C33, 49M35, 49M40, 65K05

PII. S1052623495279359

1. Introduction.

The linear complementarity problem (LCP) consists of determining a vector pair $(x, s) \in \mathbb{R}^{2n}$ which satisfies the conditions

$$s = Mx + q, \qquad x \geq 0, \qquad s \geq 0, \qquad x^T s = 0, \tag{1.1}$$

where $q \in \mathbb{R}^n$ and $M \in \mathbb{R}^{n\times n}$. In this paper, we consider problem (1.1) with M a P_*-matrix. The class of P_*-matrices was introduced by Kojima, Megiddo, Noma, and Yoshise [5], and it contains many types of matrices encountered in practical applications. Let κ be a nonnegative number. A matrix M is called a $P_*(\kappa)$-matrix if

$$(1+4\kappa)\sum_{i\in\mathcal{I}_+(x)} x_i[Mx]_i + \sum_{i\in\mathcal{I}_-(x)} x_i[Mx]_i \geq 0 \quad \forall x \in \mathbb{R}^n \tag{1.2}$$

where

$$\mathcal{I}_+(x) = \{i \ : \ x_i[Mx]_i > 0\} \ , \ \ \mathcal{I}_-(x) = \{i \ : \ x_i[Mx]_i < 0\}$$

or, equivalently, if

$$x^T Mx \geq -4\kappa \sum_{i\in\mathcal{I}_+(x)} x_i[Mx]_i \ \ \forall x \in \mathbb{R}^n \ . \tag{1.3}$$

The class of all $P_*(\kappa)$-matrices is denoted by $P_*(\kappa)$, and the class P_* is defined by

$$P_* = \bigcup_{\kappa\geq 0} P_*(\kappa);$$

*Received by the editors January 3, 1995; accepted for publication (in revised form) January 5, 1996. This work was supported in part by NSF grant DMS 9305760.
http://www.siam.org/journals/siopt/7-2/27935.html

†Department of Mathematics, The University of Iowa, Iowa City, IA 52242 (potra@math.uiowa.edu, rsheng@math.uiowa.edu).

i.e., M is a P_*-matrix if $M \in P_*(\kappa)$ for some $\kappa \geq 0$.

Obviously, $P_*(0) = PSD$ (the class of positive semidefinite matrices) and $P_*(\kappa_1) \subset P_*(\kappa_2)$ for $0 \leq \kappa_1 \leq \kappa_2$. Also, we have $P_* \supset P$, where P is the class of all matrices with positive principal minors. This follows from the fact that a P-matrix M is a $P_*(\kappa)$-matrix for $\kappa = \max\{-\frac{\lambda_{\min}(M)}{4\gamma(M)},\ 0\}$, where $\lambda_{\min}(M)$ is the least eigenvalue of $(M + M^T)/2$ and $\gamma(M) > 0$ is the so-called P-matrix number of M (see [5, Lemma 3.3]).

Most interior-point methods for linear programming problems have been extended for monotone linear complementarity problems, i.e., for the $P_*(0)$-matrix LCP. Some of them have also been extended to the P_*-matrix LCP (cf. [3, 4, 5, 7]).

In [5] and [7] it is assumed that the starting point (x^0, s^0) is strictly feasible in the sense that it belongs to the relative interior of the feasible set

$$\mathcal{F} = \{(x, s) \in \mathbb{R}^{2n}_+ \,:\, s = Mx + q\}.$$

Such a starting point may be very difficult to find in practical applications. Moreover, the existence of a strictly feasible starting point implies that the solution set

$$\mathcal{F}^* = \{(x^*, s^*) \in \mathcal{F} \,:\, {x^*}^T s^* = 0\}$$

is nonempty and bounded, which restricts the class of problems to which the methods apply.

We note that all known infeasible-interior-point algorithms for the P_*-matrix LCP depend on the classification number κ for their implementation. However, sometimes it is extremely difficult to estimate κ. In this paper, we propose a large-step infeasible-path-following algorithm independent of κ. If the problem is solvable, the algorithm is globally convergent when starting from an arbitrary positive starting point (x^0, s^0). The algorithm is defined in a large neighborhood of the central path and requires one matrix factorization per iteration. If a well-centered starting point is feasible or close to being feasible, then the algorithm has $O((1+\kappa)\sqrt{n}\ln(\epsilon_0/\epsilon))$-iteration complexity. If the starting point is large enough, then the iteration complexity is $O((1+\kappa)^2 n \ln(\epsilon_0/\epsilon))$. With appropriate initialization, a slightly modified version of the algorithm terminates in $O((1+\kappa)^2 n \ln(\epsilon_0/\epsilon))$ steps either by finding a solution or by determining that the problem is not solvable. High-order local convergence is proved for problems having a strictly complementary solution.

We mention that our algorithm is new even for monotone LCP, where it attains $O(n \ln(\epsilon_0/\epsilon))$-iteration complexity for infeasible starting points, as compared with $O(n^2 \ln(\epsilon_0/\epsilon))$-iteration complexity of the algorithms, with similar work per iteration, proposed by Wright [19, 17, 18], Wright and Zhang [20], and Zhang [22]. In this paper, we have adapted the algorithm structure of "fast-safe-improve" of Wright and Zhang [20] by using new search directions and new neighborhoods.

2. A generic large-step infeasible-interior-point algorithm. Our convergence results are proved under the assumption that $\mathcal{F}^*$ is not empty. It is easily seen that $(x^*, s^*) \in \mathcal{F}^*$ if and only if (x^*, s^*) is a nonnegative solution of the following nonlinear system:

$$F(x, s) := \begin{pmatrix} Xs \\ Mx - s + q \end{pmatrix} = 0. \tag{2.1}$$

For any given $\epsilon > 0$, we define the set of *ϵ-approximate solutions* of (1.1) as

$$\mathcal{F}_\epsilon = \{(x, s) \in \mathbb{R}^{2n}_{++}, \quad x^T s \leq \epsilon, \quad \|Mx - s + q\| \leq \epsilon\}.$$

In what follows, we present an algorithm that finds a point in this set in a finite number of steps. The starting point of the algorithm can be any pair of strictly positive vectors $(x^0, s^0) > 0$. First, let us define

$$p = \frac{n}{(x^0)^T s^0} X^0 s^0, \quad \rho_0 = \min\left\{[p]_i : 1 \le i \le n\right\}, \quad \tau_0 = (x^0)^T s^0/n.$$

Obviously, we have

$$\rho_0 \le 1, \quad \sum_{i=1}^{n} [p]_i = n.$$

The algorithm depends on seven positive parameters:

$$0 < \alpha_{\min} < \alpha_{\max} \le 1, \tag{2.2a}$$

$$0 < \beta_{\min} < \beta_{\max}, \tag{2.2b}$$

$$0 < \rho < \gamma < 1, \qquad \rho < \overline{\rho}, \tag{2.2c}$$

and a nonnegative integer I. The complexity of the algorithm will not depend on $\overline{\rho}$ and I. However, the asymptotic order of convergence depends on I.

Note that these parameters are independent of κ and n. We define a neighborhood

$$\begin{aligned}\mathcal{N} = \{(x, s, \tau, \alpha, \beta) \in \mathbb{R}^{2n+3}_{++} : Xs \ge \alpha\tau p, \quad \|Xs - \tau\, p\| \le \beta\tau, \quad r = s - Mx - q,\\ r = (\tau/\tau_0) r^0,\ \tau \le \tau_0,\ \alpha_{\min} \le \alpha \le \alpha_{\max},\ \beta_{\min} \le \beta \le \beta_{\max}\}\end{aligned}$$

and denote

$$\mu = \frac{1}{n} x^T s.$$

We note that $\beta \in (0, 1)$ is a typical requirement of short-step algorithms. In our algorithm, β no longer has such a restriction. Therefore, $\mathcal{N}$ is a large neighborhood of the infeasible central path defined by

$$Xs = \tau p, \tag{2.3a}$$

$$s = Mx + q + (\tau/\tau_0) r^0. \tag{2.3b}$$

Also, α, β may change at each step. A neighborhood similar to $\mathcal{N}$ has been used by Xu [21] in the case of a homogeneous self-dual reformulation of a linear programming problem.

In our algorithm, τ is driven to zero in a specified manner. Since all the points $(x, s, \tau, \alpha, \beta)$ are in $\mathcal{N}$, we have

$$\alpha_{\min}\tau \le \mu \le (1 + \beta_{\max})\tau, \tag{2.4}$$

so that μ will be driven to zero at about the same rate as τ. On the other hand, $\|r\|$ is driven to zero at exactly the same rate as τ.

At the beginning of the kth iteration of our algorithm, a point $(x^k, s^k, \tau_k, \alpha_k, \beta_k) \in \mathcal{N}$ is given. The kth iteration consists of several steps. Each step has input $(x, s, \tau, \alpha, \beta) \in \mathcal{N}$ and output $(x^+, s^+, \tau_+, \alpha_+, \beta_+) \in \mathcal{N}$. Such a step is defined as follows:

(a) Choose $\lambda \in [0, 1)$ ($\lambda = 0$ for *fast step*, $\lambda \in (0, 1)$ for *safe step*).

(b) Solve the linear system

$$S^k u + X^k v = \lambda \tau p - Xs, \tag{2.5a}$$

$$Mu - v = (1-\lambda) r. \tag{2.5b}$$

(c) Denote

$$x(\theta) = x + \theta u, \quad s(\theta) = s + \theta v.$$

(d) Choose α_+, β_+ such that $\alpha_{\min} \le \alpha_+ \le \alpha$, $\beta \le \beta_+ \le \beta_{\max}$.

(e) Compute

$$\begin{aligned} \bar{\theta} = \max\{\tilde{\theta} \in [0,1] :\ & X(\theta)s(\theta) \ge \alpha_+(1-(1-\lambda)\theta)\tau p, \\ & \|X(\theta)s(\theta) - (1-(1-\lambda)\theta)\tau\, p\| \le \beta_+(1-(1-\lambda)\theta)\tau \forall \theta \in [0,\tilde{\theta}]\}. \end{aligned} \tag{2.6}$$

(f)Set $x^+ = x + \bar{\theta} u$, $s^+ = s + \bar{\theta} v$, $\tau_+ = (1-(1-\lambda)\bar{\theta})\tau$.

Let us note that the computation of $\bar{\theta}$ above involves the solutions of n quadratic equations and a quartic equation. In the following analysis, we assume that these equations are solved exactly. The results are true for appropriate approximate solutions of these equations by use of the bisection method (see also [2]).

ALGORITHM 2.1.

```
Given $\alpha_{\max}, \alpha_{\min}, \beta_{\max}, \beta_{\min}, \gamma, \rho$ satisfying (2.2a) and $(x^0, s^0) > 0$;
$t_0 \leftarrow 0,\ \alpha_0 \leftarrow \alpha_{\max},\ \beta_0 \leftarrow \beta_{\min},\ \tau_0 \leftarrow \mu_0$;
while $\max\{\mu_k, \|r^k\|\} > \epsilon$
        * fast branch *
        solve (b)–(e) with $(x,s) = (x^k, s^k),\ \lambda = 0$,
        $\alpha_+ = \alpha_{\min} + \gamma^{t_k+1}(\alpha_{\max} - \alpha_{\min}),\ \beta_+ = \beta_{\max} - \gamma^{t_k+1}(\beta_{\max} - \beta_{\min})$;
        if      $1 - \bar{\theta} \le \rho$
                $t_+ \leftarrow t_k + 1,\ \lambda_k \leftarrow 0$;
        else
                * safe branch *
                solve (b)–(e) with $(x,s) = (x^k, s^k),\ \lambda \in (0,1)$,
                $\alpha_+ = \alpha_{\min} + \gamma^{t_k}(\alpha_{\max} - \alpha_{\min}),\ \beta_+ = \beta_{\max} - \gamma^{t_k}(\beta_{\max} - \beta_{\min})$;
                $t_+ \leftarrow t_k,\ \lambda_k \leftarrow \lambda$;
        end if
        * main points *
        $\tau_+ \leftarrow \tau_k(1 - (1-\lambda_k)\bar{\theta})$,
        $(x^+, s^+) \leftarrow (x^k, s^k) + \bar{\theta}(u, v)$;
        improve $(x^k, y^k, (x^+, s^+, \tau_+, \alpha_+, \beta_+, t_+))$;
        * final points *
        $(x^{k+1},\ s^{k+1},\ \tau_{k+1},\ \alpha_{k+1},\ \beta_{k+1}) \leftarrow (x^+,\ s^+,\ \tau_+,\ \alpha_+,\ \beta_+)$,
        $t_{k+1} \leftarrow t_+,\ k \leftarrow k + 1$;
end while
```

The **improve** procedure is defined as follows:

```
Given $\bar{\rho} \in (\rho, 1),\ I \ge 0$,
for $i = 1, 2, \ldots, I$
        $(x, s, \tau, \alpha, \beta, t) \leftarrow (x^+, s^+, \tau_+, \alpha_+, \beta_+, t_+)$;
        if      $\max\{\mu, \|r\|\} \le \epsilon$, then return;
        solve (b)–(e) with $\lambda = 0$,
        $\alpha_+ = \alpha_{\min} + \gamma^{t+1}(\alpha_{\max} - \alpha_{\min}),\ \beta_+ = \beta_{\max} - \gamma^{t+1}(\beta_{\max} - \beta_{\min})$;
        **if**    $1 - \bar\theta \le \rho$
                $t \leftarrow t + 1;$
        **else**
                $t_+ \leftarrow t,$
                *solve* (b)–(e) *with* $\lambda \in (0, 1),$
                $\alpha_+ = \alpha_{\min} + \gamma^t(\alpha_{\max} - \alpha_{\min}),\ \ \beta_+ = \beta_{\max} - \gamma^t(\beta_{\max} - \beta_{\min});$
                **if** $1 - (1 - \lambda)\bar\theta > \bar\rho$ **then** *return;*
        **end if**
        * *intermediate points* *
        $\tau_+ \leftarrow \tau(1 - (1 - \lambda)\bar\theta),$
        $(x^+, s^+) \leftarrow (x, s) + \bar\theta(u, v);$
**end for**
```

The algorithm begins each iteration by trying a fast step, which uses an affine scaling search direction. The fast steps are accepted only if they produce a reduction in τ_k or $\|r^k\|$ of at least a factor of ρ. Otherwise, the algorithm reverts to taking a safe step. Then it goes to **improve** by reusing the coefficient matrix in (2.5) and taking a combination of safe and fast steps, just like in the main algorithm. However, the **improve** procedure terminates if τ and $\|r\|$ are not improved by at least a factor of $\bar\rho \in (\rho, 1)$. The parameter $\bar\rho$ and the nonnegative integer I are supplied by the user, where I is the maximum number of steps that can be taken in **improve**.

In the next lemma, we show that if the main points defined by Algorithm 2.1 are generated in the safe branch then the improvement rate τ_+/τ is bounded by a quantity that increases with $\delta = \|Uv\|/\tau$. Later on, we will prove global convergence by showing that δ is bounded.

LEMMA 2.2. *Assume that* $(x^k, s^k, \tau_k, \alpha_k, \beta_k) \in \mathcal{N}$ *and suppose that the main points in Algorithm* 2.1 *are generated by the safe branch. Then* $\bar\theta \in [0, 1]$ *defined by* (2.6) *satisfies*

$$\bar\theta \ge \hat\theta := \min\left\{1,\ \ \frac{\lambda\ \min((1-\alpha)\rho_0\ \ , \beta)}{\delta}\right\},$$

where $\delta = \|Uv\|/\tau$. *Moreover,*

$$\tau^+/\tau \le 1 - \min\left\{1,\ \ \frac{\lambda\ \min((1-\alpha)\rho_0\ \ , \beta)}{\delta}\right\}(1 - \lambda). \tag{2.7}$$

Proof. As the main points are generated by the safe branch, we have $\alpha_+ = \alpha$, $\beta_+ = \beta$. Since $(x, s, \tau, \alpha, \beta) \in \mathcal{N}$, we obtain

$$Xs \ge \alpha\tau p, \quad \|Xs - \tau p\| \le \beta\tau.$$

By definition, we get

$$\begin{aligned} X(\theta)s(\theta) &= Xs + \theta(Su + Xv) + \theta^2 Uv \\ &= (1-\theta)Xs + \theta\lambda\tau p + \theta^2 Uv. \end{aligned}$$

Hence, we deduce

$$\begin{aligned} &X(\theta)s(\theta) - \alpha(1 - (1-\lambda)\theta)\tau p \\ &= (1-\theta)(Xs - \alpha\tau p) + \theta[(1-\alpha)\lambda\tau p + \theta Uv] \\ &\ge \theta[(1-\alpha)\lambda\tau p + \theta Uv] \\ &\ge \theta\tau[(1-\alpha)\lambda p - \theta\delta e] \\ &\ge 0 \ \text{ for all } \ \theta \in [0, \hat\theta]. \end{aligned}$$

On the other hand, we have

$$\begin{aligned}
&\|X(\theta)s(\theta) - (1-(1-\lambda)\theta)\tau p\| - \beta(1-(1-\lambda)\theta)\tau \\
&= \|(1-\theta)(Xs-\tau p) + \theta^2 Uv\| - \beta(1-(1-\lambda)\theta)\tau \\
&\le (1-\theta)\|Xs-\tau p\| + \theta^2\|Uv\| - \beta(1-(1-\lambda)\theta)\tau \\
&\le \theta\tau(\theta\delta - \lambda\beta) \\
&\le 0 \ \text{ for all } \ \theta \in [0,\hat{\theta}].
\end{aligned}$$

Consequently, $\bar{\theta} \ge \hat{\theta}$, and (2.7) follows immediately. □

We will see that if the problem has a solution then for any $\epsilon > 0$, Algorithm 2.1 terminates in a finite number (say K_ϵ) of iterations. If $\epsilon = 0$ then the algorithm is likely to generate an infinite sequence. However, it may happen that at a certain iteration (K_0, say) we have $(1-\lambda)\bar{\theta} = 1$ which implies that an exact solution is obtained and therefore the algorithm terminates at iteration K_0. If this (unlikely) phenomenon does not happen, we set $K_0 = \infty$.

THEOREM 2.3. *For any integer* $0 \le k < K_0$, *Algorithm* 2.1 *defines* $x^k, s^k, \tau_k, \alpha_k$, *and* β_k *such that*

$$(x^k, s^k, \tau_k, \alpha_k, \beta_k) \in \mathcal{N}, \tag{2.8}$$

$$r^k = \psi_k r^0, \tag{2.9}$$

where

$$\psi_k = \tau_k/\tau_0 \le \prod_{j=1}^{k}(1-(1-\lambda_j)\bar{\theta}_j), \tag{2.10}$$

and $\bar{\theta}_j$ *is defined by* (2.6). *Moreover, all the intermediate points produced in the* **improve** *procedure also satisfy*

$$(x, s, \tau, \alpha, \beta) \in \mathcal{N}. \tag{2.11}$$

Proof. The proof is by induction. First, we note that

$$\|X^0 s^0 - \tau_0 p\| = \|X^0 s^0 - \mu_0 p\| = 0 < \beta_0\tau_0$$

and

$$X^0 s^0 \ge \alpha_{\max} X^0 s^0 = \alpha_0 \mu_0 p = \alpha_0 \tau_0 p,$$

which show that $(x^0, s^0, \tau_0, \alpha_0, \beta_0) \in \mathcal{N}$. Hence, (2.8) and (2.9) are satisfied for $k = 0$. Suppose they are satisfied for $k = 0, 1, \ldots, l$ for some $l \ge 0$ and denote $(x, s) = (x^l, s^l)$. Then by Lemma 2.2, $\bar{\theta}$ exists and we have

$$X(\theta)s(\theta) \ge \alpha(1-(1-\lambda)\theta)\tau p > 0 \ \text{ for all } \ 0 \le \theta \le \bar{\theta}. \tag{2.12}$$

The positivity of x^+ and s^+ is proved by contradiction. Suppose, for example, that $[x^+]_i \le 0$ for some i. Then there must exist $\theta', 0 \le \theta' \le \bar{\theta}$, such that $[x(\theta')]_i = 0$. This implies

$$[x(\theta')]_i [s(\theta')]_i = 0,$$

which contradicts (2.12). Hence, (2.8) is satisfied for $k = l+1$. The fact that (2.9) is verified for $k = l+1$ follows from the definition of Algorithm 2.1. Analogously, we can prove (2.11). □

In next section, we will show that with the appropriate choice of λ and (x^0, s^0), Algorithm 2.1 achieves $O((\kappa+1)^2 n \ln(\epsilon_0/\epsilon))$-iteration complexity.

3. Global convergence.

In this section we assume that $\mathcal{F}^*$ is nonempty. Under this assumption, we will prove that Algorithm 2.1, with $\epsilon = 0$, is globally convergent in the sense that

$$\lim_{k\to\infty} \mu_k = 0, \quad \lim_{k\to\infty} r^k = 0$$

for any starting point $(x^0, s^0) > 0$. We start with the following three technical lemmas.

LEMMA 3.1. *Assume that $\mathcal{F}^*$ is nonempty. Then for any $(x^*, s^*) \in \mathcal{F}^*$ and $(x, s, \tau, \alpha, \beta) \in \mathcal{N}$, we have*

$$\psi((x)^T s^0 + (s)^T x^0) \leq (1 + \beta_{\max})(1 + 4\kappa)(2 + \zeta)n\tau \ , \tag{3.1a}$$

$$(1 - \psi)((x)^T s^* + (s)^T x^*) \leq (1 + \beta_{\max})(1 + 4\kappa)((1 + \psi) + (1 - \psi)\zeta)n\tau \ , \tag{3.1b}$$

where

$$\psi = \tau/\tau_0, \quad \zeta = \frac{(x^0)^T s^* + (s^0)^T x^*}{(x^0)^T s^0}. \tag{3.2}$$

Proof. According to (2.9), we have

$$\begin{aligned}\psi s^0 + (1 - \psi)s^* - s &= \psi(r^0 + M(x^0 - x^*)) - (r + M(x - x^*)),\\ &= M(\psi x^0 + (1 - \psi)x^* - x) \ .\end{aligned}$$

Using the inequalities $(x^*, s^*) \geq 0$, $(x, s) > 0$, and the defining property (1.3) of a $P_*(\kappa)$-matrix, we can write

$$\begin{aligned}&[\psi x^0 + (1 - \psi)x^* - x]^T[\psi s^0 + (1 - \psi)s^* - s]\\ &\geq -4\kappa \sum_{i\in\mathcal{I}_+} [\psi x^0 + (1 - \psi)x^* - x]_i[\psi s^0 + (1 - \psi)s^* - s]_i\\ &\geq -4\kappa \sum_{i\in\mathcal{I}_+} \left(\psi^2[x^0]_i[s^0]_i + (1 - \psi)\psi([x^*]_i[s^0]_i + [x^0]_i[s^*]_i) + [x]_i[s]_i\right)\\ &\geq -4\kappa(\psi^2(x^0)^T s^0 + (1 - \psi)\psi((x^*)^T s^0 + (x^0)^T s^*) + x^T s) \ ,\end{aligned} \tag{3.3}$$

where

$$\mathcal{I}_+ = \{i \ : \ [\psi x^0 + (1 - \psi)x^* - x]_i[\psi s^0 + (1 - \psi)s^* - s]_i > 0\} \ .$$

By expanding (3.3), we obtain

$$\begin{aligned}&[\psi x^0 + (1 - \psi)x^* - x]^T[\psi s^0 + (1 - \psi)s^* - s]\\ &= \psi^2 n\mu_0 + (1 - \psi)\psi((x^0)^T s^* + (s^0)^T x^*)\\ &\quad -\psi((x^0)^T s + (s^0)^T x) + x^T s - (1 - \psi)(s^T x^* + x^T s^*) + (1 - \psi)^2(x^*)^T s^*.\end{aligned} \tag{3.4}$$

The desired inequalities (3.1) follow from (3.3) and (3.4) by applying Theorem 2.3 and the relations $(x^*)^T s^* = 0$, $s^T x^* + x^T s^* \geq 0$, $s^T x^0 + x^T s^0 > 0$, and $\mu \leq (1 + \beta_{\max})\tau$ (see (2.4)) . □

In order to prove global convergence, we have to show that $((1 - \lambda)\bar{\theta})$ is bounded away from zero. An essential tool is the following technical result of [3].

LEMMA 3.2. *Let x, s, a, r be four n-dimensional vectors with $x > 0$ and $s > 0$, and let $M \in \mathbb{R}^{n\times n}$ be a $P_*(\kappa)$-matrix. Then the solution (u, v) of the linear system*

$$\begin{aligned}Su + Xv &= a \ ,\\ Mu - v &= b\end{aligned}$$

satisfies the following relations:

$$\begin{aligned}
\|Du\| &\le \|\widetilde{b}\| + \sqrt{\|\widetilde{a}\|^2 + \|\widetilde{b}\|^2 + 2\kappa\|\widetilde{c}\|^2}\ ,\\
\|D^{-1}v\| &\le \sqrt{\|\widetilde{a}\|^2 + \|\widetilde{b}\|^2 + 2\kappa\|\widetilde{c}\|^2}\ ,\\
\|Du\|^2 + \|D^{-1}v\|^2 &\le \|\widetilde{a}\|^2 + 2\kappa\|\widetilde{c}\|^2 + 2\|\widetilde{b}\|^2 + 2\|\widetilde{b}\|\sqrt{\|\widetilde{a}\|^2 + \|\widetilde{b}\|^2 + 2\kappa\|\widetilde{c}\|^2}\\
&\equiv \chi_1^2\ ,\\
\|Uv\|^2 &\le \frac{1}{8}\|\widetilde{a}\|^4 + \frac{1}{4}\chi_1^2(\chi_1^2 - \|\widetilde{a}\|^2)\ ,
\end{aligned}$$

where

$$D = X^{-1/2}S^{1/2}\ ,\ \widetilde{a} = (XS)^{-1/2}a\ ,\ \widetilde{b} = D^{-1}b\ ,\ \widetilde{c} = \widetilde{a} + \widetilde{b}\ .$$

We use the above result to find bounds for the quantity δ arising in the estimate given by Lemma 2.2 when the algorithm chooses the safe step.

LEMMA 3.3. *During the safe branch, $\delta = \|Uv\|/\tau$ satisfies*

$$\delta \le \phi(\kappa)\left\{(1-\lambda)[\|p\|/\sqrt{\alpha\rho_0} + \|X^{1/2}S^{-1/2}r\|/\sqrt{\tau}] + \beta/\sqrt{\alpha\rho_0}\right\}^2, \tag{3.5}$$

where

$$\phi(\kappa) := [.25(\sqrt{1+2\kappa}+1)^4 + .125]^{1/2}.$$

Moreover,

$$\|X^{1/2}S^{-1/2}r\|/\sqrt{\tau} \le \eta\sqrt{n(1+\beta)},$$

where

$$\eta := \sqrt{n}(1+4\kappa)(2+\zeta)\|(S^0)^{-1}r^0\|_\infty\sqrt{\frac{1+\beta_{\max}}{\rho_0\alpha_{\min}}},$$

with ζ given by (3.2).

Proof. By applying Lemma 3.2 to the linear system (2.5) with $a = \lambda\tau p - Xs$ and $b = (1-\lambda)r$, we have

$$\begin{aligned}
\|\widetilde{a}\| &= \|(XS)^{-1/2}(\lambda\tau p - Xs)\|\\
&\le (1-\lambda)\tau\|(XS)^{-1/2}p\| + \|(XS)^{-1/2}(Xs - \tau p)\|\\
&\le (1-\lambda)\sqrt{\tau}\|p\|/\sqrt{\alpha\rho_0} + \sqrt{\tau}\beta/\sqrt{\alpha\rho_0},\\
\|\widetilde{b}\| &= (1-\lambda)\|D^{-1}r\| = (1-\lambda)\|X^{1/2}S^{-1/2}r\|,\\
\|\widetilde{c}\| &= \|\widetilde{a} + \widetilde{b}\| \le \|\widetilde{a}\| + \|\widetilde{b}\|,\\
\delta &= \|Uv\|/\tau \le \phi(\kappa)(\|\widetilde{a}\| + \|\widetilde{b}\|)^2/\tau,
\end{aligned}$$

and (3.5) follows from the above relations. From Lemma 3.1 we deduce that

$$\begin{aligned}
&\|(XS)^{-1/2}Xr\|/\sqrt{\tau} \le (\rho_0\alpha)^{-1/2}\tau^{-1}\|Xr\|\\
&\le (\rho_0\alpha)^{-1/2}\tau^{-1}\|Xr\|_1 = \psi(\rho_0\alpha)^{-1/2}\tau^{-1}\|Xr^0\|_1\\
&= \psi(\rho_0\alpha)^{-1/2}\tau^{-1}\sum_{i=1}^{n}\left|[x]_i[s^0]_i[r^0]_i/[s^0]_i\right|\\
&\le (\rho_0\alpha)^{-1/2}\tau^{-1}\|(S^0)^{-1}r^0\|_\infty\psi(s^0)^Tx\\
&\le \eta\sqrt{n(1+\beta_{max})}. \qquad \square
\end{aligned}$$

Now let us take

$$\lambda = \bar{\lambda} := \frac{\|p\|/\sqrt{\alpha\rho_0} + \|X^{1/2}S^{-1/2}r\|/\sqrt{\tau} + \beta/\sqrt{\alpha\rho_0}}{\|p\|/\sqrt{\alpha\rho_0} + \|X^{1/2}S^{-1/2}r\|/\sqrt{\tau} + \omega\,\beta/\sqrt{\alpha\rho_0}}, \tag{3.6}$$

where $\omega > 1$ is a fixed number. Clearly,

$$\bar{\lambda} \in (1/\omega, 1). \tag{3.7}$$

From (3.5) and (3.6) we have

$$\begin{aligned}\delta &\leq \phi(\kappa)\left\{(1-\lambda)[\|p\|/\sqrt{\alpha\rho_0} + \|X^{1/2}S^{-1/2}r\|/\sqrt{\tau}] + \beta/\sqrt{\alpha\rho_0}\right\}^2 \\ &\leq \phi(\kappa)\left\{\frac{[(\omega-1)\beta/\sqrt{\alpha\rho_0}\,]\,[\|p\|/\sqrt{\alpha\rho_0} + \|X^{1/2}S^{-1/2}r\|/\sqrt{\tau}\,]}{\|p\|/\sqrt{\alpha\rho_0} + \|X^{1/2}S^{-1/2}r\|/\sqrt{\tau} + \omega\,\beta/\sqrt{\alpha\rho_0}} + \beta/\sqrt{\alpha\rho_0}\right\}^2 \\ &\leq \phi(\kappa)\left(\omega\beta/\sqrt{\alpha\rho_0}\right)^2 = \phi(\kappa)\omega^2\beta^2/(\alpha\rho_0).\end{aligned} \tag{3.8}$$

Then by virtue of (2.7), (3.6), (3.7), and (3.8), we deduce that

$$\begin{aligned}\tau^+/\tau &\leq 1 - \min\left\{1,\ \lambda\frac{\min((1-\alpha)\rho_0\ \ ,\beta)}{\delta}\right\}(1-\lambda) \\ &\leq 1 - \min\left\{1,\ \frac{\min((1-\alpha)\rho_0\ \ ,\beta)}{\omega^3\phi(\kappa)\beta^2/(\alpha\rho_0)}\right\}(1-\bar{\lambda}) \\ &\leq 1 - \min\left\{1,\ \frac{\min\{(1-\alpha)\rho_0\ \ ,\beta\}}{\omega^3\phi(\kappa)\beta^2/(\alpha\rho_0)}\right\}\frac{(\omega-1)\beta}{\|p\| + \eta\sqrt{n\alpha\rho_0(1+\beta)} + \omega\beta} \\ &\leq 1 - \theta^*,\end{aligned} \tag{3.9}$$

where

(3.10)

$$\theta^* := \min\left\{1,\ \frac{\min\{(1-\alpha_{\max})\rho_0\ \ ,\beta_{\min}\}}{\omega^3\phi(\kappa)\beta_{\max}^2/(\alpha_{\min}\rho_0)}\right\}\frac{(\omega-1)\beta_{\min}}{\|p\| + \eta\sqrt{n\alpha_{\max}\rho_0(1+\beta_{\max})} + \omega\beta_{\max}}.$$

Suppose the algorithm takes the fast branch at iteration k. Since the **improve** procedure never increases the value of ψ, we must have $\psi_{k+1}/\psi_k \leq \rho$; if a safe branch is taken we must have $\psi_{k+1}/\psi_k \leq 1 - \theta^*$. Therefore, we obtain

$$\psi_{k+1}/\psi_k \leq \max\{1-\theta^*, \rho\}. \tag{3.11}$$

This observation shows that our algorithm converges at a global linear rate.

THEOREM 3.4. *Suppose that the optimal set $\mathcal{F}^*$ is nonempty and, in the safe branch, λ_k is defined by* (3.6).

(i) *If $\epsilon = 0$ then Algorithm* 2.1 *either finds an optimal solution $z^* \in \mathcal{F}^*$ in a finite number of steps or produces an infinite sequence $z^k = (x^k, s^k)$ such that*

$$\lim_{k\to\infty}(x^k)^T s^k = 0,\ \lim_{k\to\infty}(r^k) = 0.$$

(ii) *If $\epsilon > 0$ then Algorithm* 2.1 *terminates with a $z \in \mathcal{F}_\epsilon$ in at most*

$$K_\epsilon = \lceil \frac{|\ln(\frac{\epsilon}{\epsilon_0})|}{|\ln(\max\{1-\theta^*, \rho\})|} \rceil$$

iterations, where $\epsilon_0 = \max\{(1+\beta)(x^0)^T s^0, \|r^0\|\}$ and $\lceil\chi\rceil$ denotes the smallest integer greater than or equal to χ.

For starting points that are feasible or close to being feasible, we obtain the following complexity result.

COROLLARY 3.5. *Under the hypothesis of Theorem* 3.4, *suppose that* $\rho_0 = \Omega(1)$ *and the starting point is chosen such that there is a constant* C *independent of* n *and* κ *satisfying the inequality*

$$(2+\zeta)\|(S^0)^{-1}r^0\|_\infty \le \frac{C}{(1+\kappa)\sqrt{n}}.$$

Then Algorithm 2.1 *terminates in at most*

$$\tilde{K}_\epsilon = O((1+\kappa)\sqrt{n}\ln(\epsilon_0/\epsilon))$$

iterations.

Most of the complexity results on infeasible-interior-point methods are obtained for starting points of the form

$$x^0 = \rho_p e, \quad s^0 = \rho_d e \ , \tag{3.12}$$

where ρ_p and ρ_d are sufficiently large positive constants (big M initialization). For such starting points, we have clearly $\rho_0 = 1$ and

$$\zeta = \|x^*\|_1/(n\rho_p) + \|s^*\|_1/(n\rho_d) \ \text{ for some } (x^*, s^*) \in \mathcal{F}^* \ ,$$

$$\|(S^0)^{-1}r^0\|_\infty \le 1 + (\rho_p/\rho_d)\|Me\|_\infty + (1/\rho_d)\|q\|_\infty.$$

Therefore, if ρ_p and ρ_d satisfy the inequalities

$$\rho_p \ge n^{-1}\|x^*\|_1 \ , \quad \rho_d \ge \max\{\rho_p\|Me\|_\infty \ , \|q\|_\infty, n^{-1}\|s^*\|_1\} \tag{3.13}$$

for some $(x^*, s^*) \in \mathcal{F}^*$, then $\eta \le O((1+\kappa)\sqrt{n})$. Hence, from (3.10) and Theorem 3.4 we deduce the following computational complexity bound.

COROLLARY 3.6. *Under the hypothesis of Theorem* 3.4, *suppose that the starting point satisfies* (3.13) *for some* $(x^*, s^*) \in \mathcal{F}^*$. *Then Algorithm* 2.1 *terminates in at most*

$$\tilde{K}_\epsilon = O((1+\kappa)^2 n \ln(\epsilon_0/\epsilon)) \tag{3.14}$$

iterations.

Let us end this section by noting that while our algorithm does not use any information on the classification number κ, its computational complexity depends on κ.

4. Infeasibility detection.

All of the above results have been proved under the assumption that $\mathcal{F}^*$ is nonempty. It turns out that Algorithm 2.1 can be modified in such a way that it can detect whether $\mathcal{F}^*$ contains points of norm less than a quantity chosen in advance provided that the parameter κ is known. Let $\bar{\rho}_p$, $\bar{\rho}_d$ be such quantities. Also, let us define

$$\bar{\zeta} = (\|x^0\|\bar{\rho}_d + \|s^0\|\bar{\rho}_p)/((x^0)^T s^0),$$

$$\bar{\eta} = \sqrt{n}(1+4\kappa)(2+\bar{\zeta})\|(S^0)^{-1}r^0\|_\infty \sqrt{\frac{1+\beta_{\max}}{\alpha_{\min}\rho_0}}, \tag{4.1}$$

$$\overline{\theta}^* := \min\left\{1,\ \ \frac{\min\{(1-\alpha_{\max})\rho_0\ \ ,\beta_{\min}\}}{\omega^3\phi(\kappa)\beta^2_{\max}/(\alpha_{\min}\rho_0)}\right\}\frac{(\omega-1)\beta_{\min}}{\|p\|+\overline{\eta}\sqrt{n\alpha_{\max}\rho_0(1+\beta_{\max})}+\omega\beta_{\max}}.$$

In what follows, we always assume that λ is defined by (3.6). Then we can prove the following theorem.

THEOREM 4.1. *Suppose that the following instruction "* **If**

$$(x^0)^T s^k + (s^0)^T x^k$$
$$> (1+4\kappa)[(\tau_k/\tau_0)(x^0)^T s^0 + (\tau_0/\tau_k)(x^k)^T s^k + (1-\tau_k/\tau_0)(\overline{\rho}_d\|x^0\| + \overline{\rho}_p\|s^0\|)]$$

then terminate*" is inserted just before the* **while** *loop of Algorithm* 2.1*. Then the new algorithm terminates in at most*

$$\overline{K}_\epsilon = \lceil \frac{|\ln(\frac{\epsilon}{\epsilon_0})|}{|\ln(\max\{1-\overline{\theta}^*, \rho\})|}\rceil$$

iterations either by finding an ϵ-approximate solution or by determining that either there is no $z^* = (x^*, s^*) \in \mathcal{F}^*$ with $\|x^*\| \le \overline{\rho}_p$, $\|s^*\| \le \overline{\rho}_d$ or M is not a $P_*(\kappa)$-matrix.

Proof. Suppose that M is a $P_*(\kappa)$-matrix and that the inequality

$$\begin{aligned}(4.2)\quad &(x^0)^T s^k + (s^0)^T x^k\\ &\le (1+4\kappa)[(\tau_k/\tau_0)(x^0)^T s^0 + (\tau_0/\tau_k)(x^k)^T s^k + (1-\tau_k/\tau_0)(\overline{\rho}_d\|x^0\| + \overline{\rho}_p\|s^0\|)]\end{aligned}$$

holds for all $0 \le k \le \overline{K}_\epsilon$. Then we have

$$\begin{aligned}(4.3)\qquad &\psi_k((x^0)^T s^k + (s^0)^T x^k)\\ &\le \psi_k(1+4\kappa)[\psi_k n\mu_0 + (1/\psi_k)n\mu_k + (1-\psi_k)(\overline{\rho}_d\|x^0\| + \overline{\rho}_p\|s^0\|)]\\ &= (1+4\kappa)[\psi_k^2 n\mu_0 + n\mu_k + \psi_k(1-\psi_k)\overline{\zeta} n\mu_0]\\ &\le (1+4\kappa)(1+\beta_{\max})[\psi_k^2 n\tau_0 + n\tau_k + \psi_k(1-\psi_k)\overline{\zeta} n\tau_0]\quad \text{(see (2.4))}\\ &= (1+4\kappa)(1+\beta_{\max})[\psi_k n\tau_k + n\tau_k + (1-\psi_k)\overline{\zeta} n\tau_k]\quad \text{(from (2.10))}\\ &\le (1+4\kappa)(1+\beta_{\max})(2+\overline{\zeta})n\tau_k, \qquad 0 \le k \le \overline{K}_\epsilon,\end{aligned}$$

where the last inequality follows from the fact that $\psi_k \in [0,\ 1]$ for all k. With the help of (4.3) we can show that $\tau_{k+1}/\tau_k \le \max\{1-\overline{\theta}^*,\ \rho\}$ for all $0 \le k \le \overline{K}_\epsilon$, which implies $(x^{\overline{K}_\epsilon}, s^{\overline{K}_\epsilon}) \in \mathcal{F}_\epsilon$.

With the help of (4.3), we can show that $\tau_{k+1}/\tau_k \le \max\{1-\overline{\theta}^*,\ \rho\}$ for all $0 \le k \le \overline{K}_\epsilon$, which implies $(x^{\overline{K}_\epsilon}, s^{\overline{K}_\epsilon}) \in \mathcal{F}_\epsilon$.

On the other hand, if there exists $(x^*, s^*) \in \mathcal{F}^*$ such that $\|x^*\| \le \overline{\rho}_p$, $\|s^*\| \le \overline{\rho}_d$, then (3.3) and (3.4) imply that (4.2) must hold for all $k \ge 0$. Therefore, if (4.2) is violated for some $k \le \overline{K}_\epsilon$, then there is no $(x^*, s^*) \in \mathcal{F}^*$ such that $\|x^*\| \le \overline{\rho}_p$, $\|s^*\| \le \overline{\rho}_d$. This completes the proof of our theorem. □

From the above theorem we can obtain the following iteration complexity for our new algorithm under certain assumptions on the starting point.

COROLLARY 4.2. *Under the hypothesis of Theorem* 4.1*, suppose that $\rho_0 = \Omega(1)$ and that the starting point satisfies*

$$(2+\overline{\zeta})\|(S^0)^{-1}r^0\|_\infty \le \frac{C}{(1+\kappa)\sqrt{n}}$$

for some constant C independent of n and κ. Then the new algorithm terminates in at most

$$\tilde{K}_\epsilon = O((1+\kappa)\sqrt{n}\ln(\epsilon_0/\epsilon))$$

iterations either by finding an ϵ-approximate solution or by determining that either there is no $z^* = (x^*, s^*) \in \mathcal{F}^*$ with $\|x^*\| \leq \overline{\rho}_p$, $\|s^*\| \leq \overline{\rho}_d$ or M is not a $P_*(\kappa)$-matrix.

Suppose now that the starting point satisfies

$$x^0 = \hat{\rho}_p e, \quad s^0 = \hat{\rho}_d e, \tag{4.4}$$

where

$$\hat{\rho}_p \geq \overline{\rho}_p/\sqrt{n}, \tag{4.5}$$

$$\hat{\rho}_d \geq \max\{\hat{\rho}_p \|Me\|_\infty, \|q\|_\infty, \overline{\rho}_d/\sqrt{n}\}. \tag{4.6}$$

Then we obtain

$$\overline{\zeta} \leq 2, \tag{4.7}$$

$$\|(S^0)^{-1} r^0\|_\infty \leq 1 + (\hat{\rho}_p/\hat{\rho}_d)\|Me\|_\infty + (1/\hat{\rho}_d)\|q\|_\infty \leq 3. \tag{4.8}$$

From (4.1), (4.7), and (4.8), it follows that $\overline{\eta} < O((1+\kappa)\sqrt{n})$. Hence, we have the following complexity result.

COROLLARY 4.3. *Under the assumption of Theorem* 4.1*, suppose that the starting point satisfies* (4.4)–(4.6)*. Then the new algorithm terminates in at most*

$$\hat{K}_\epsilon = O((1+\kappa)^2 n \ln(\epsilon_0/\epsilon))$$

iterations either by finding an ϵ-approximate solution or by determining either there is no $z^* = (x^*, s^*) \in \mathcal{F}^*$ with $\|x^*\| \leq \overline{\rho}_p$, $\|s^*\| \leq \overline{\rho}_d$ or M is not a $P_*(\kappa)$-matrix.

5. Local convergence. In what follows, we study the asymptotic convergence properties of Algorithm 2.1 under the further assumption that (1.1) has a strictly complementary solution and $K_0 = \infty$ (i.e., the algorithm produces an infinite sequence so that an asymptotic analysis makes sense). Namely, we prove that, asymptotically, our algorithm requires one matrix factorization and $I+1$ backsolves per iteration and has the Q-order of convergence at least $I+2$. It turns out that, asymptotically, each iteration of the algorithm reduces to $I+1$ simplified Newton steps with linesearch applied to the nonlinear system defining the LCP. The fact that such a procedure has Q-order $I+2$ has already been proved in 1964 by Traub [16] for the case of scalar nonlinear equations and in 1967 by Shamanskii [14] for nonlinear systems. For sharp error estimates and other generalizations, see Potra and Ptak [12]. We note that the above mentioned results for nonlinear systems are proved under the assumption that the Jacobian is nonsingular at the solution which, in general, is not the case with the system (2.1). Also, it is assumed that full simplified Newton steps are taken, which is not the case with interior point methods since linesearch is always necessary in order to guarantee the positivity of the iterates. The first high-order local convergence results for interior-point methods based on the idea of reusing the Jacobian matrix were proved by Mehrotra [6]. No global convergence results were given for the proposed algorithm. A very elegant algorithm with both global (polynomial) convergence and high-order local convergence was proposed by Wright and Zhang [20]. In the present paper, we closely follow their analysis.

Let us denote by $\mathcal{F}^c$ the set of all strictly complementary solutions, i.e.,

$$\mathcal{F}^c = \{(x, s) \in \mathcal{F}^* : [x]_i + [s]_i > 0, i = 1, 2, \ldots, n\} .$$

It is well known that there is a unique partition

$$B \cup N = \{1, 2, \ldots, n\}\ , \quad B \cap N = \emptyset$$

such that for any $(x, s) \in \mathcal{F}^c$, we have $([x]_i > 0, [s]_i = 0$ for all $i \in B)$ and $([x]_i = 0, [s]_i > 0$ for all $i \in N)$. Let us denote the corresponding partition of M by

$$M = \begin{pmatrix} M_{BB} & M_{BN} \\ M_{NB} & M_{NN} \end{pmatrix}.$$

Also, for any vector $y \in \mathbb{R}^n$, we denote by y_B the vector of components $[y]_i,\ i \in B$ and by y_N the vector of components $[y]_i,\ i \in N$.

From the next lemma it follows that the points generated by Algorithm 2.1 are bounded.

LEMMA 5.1. *There is a constant $C_1 > 0$ such that*

$$\|(x, s)\| \leq C_1 \tag{5.1}$$

for all $(x, s, \tau, \alpha, \beta) \in \mathcal{N}$.

Proof. The proof is straightforward by observing that (3.1a) implies $x^T s^0 + s^T x^0 \leq (1 + \beta)(1 + 4\kappa)(2 + \zeta)n\tau_0$. □

According to Lemmas 4.3 and 4.5 of [4], we have the following two technical lemmas.

LEMMA 5.2. *If $\mathcal{F}^c$ is nonempty, then there is a constant $C_2 > 0$ such that for any $(x, s, \tau, \alpha, \beta) \in \mathcal{N}$ with $\tau \leq \tau_1$,*

$$\|x_N\| \leq C_2\, \tau, \ \ \|s_B\| \leq C_2 \tau$$

for all $k \geq 1$.

LEMMA 5.3. *There exists a constant C_3 such that for any $(x, s, \tau, \alpha, \beta) \in \mathcal{N}$ we have*

$$\|(u^a, v^a)\| \leq C_3 \tau, \tag{5.2}$$

where (u^a, v^a) satisfies

$$Su^a + Xv^a = -Xs, \tag{5.3a}$$

$$Mu^a - v^a = r. \tag{5.3b}$$

We now turn to the *approximate* fast steps computed by (2.5)–(2.6), where (x, s) is either the current iterate (x^k, s^k) or some intermediate point generated in the call to **improve** at the kth iteration. By the definition of the algorithm, we have

$$\tau \leq \tau_k.$$

Let us assume that the point (x, s) is not too far from (x^k, s^k) in the sense that there is a constant $\chi \geq 1$ independent of k such that

$$\|(x^k - x,\ s^k - s)\| \leq \chi \tau_k. \tag{5.4}$$

We will show later on that such a χ exists (see Lemma 5.7).

The following lemma describes some properties of the actual search direction (u, v) calculated from (2.6) in terms of the *exact* search direction (u^a, v^a) that satisfies (5.3). Similar to Lemma 4.4 of Wright and Zhang [20], we have the following result.

LEMMA 5.4. *Suppose that $\mathcal{F}^c$ is nonempty and consider the notation of Algorithm* 2.1. *Under the assumption* (5.4), *suppose that (u, v) is the solution of* (2.5) *with $\lambda = 0$; then there exists a constant C_4 independent of k and χ such that the following bounds are satisfied:*

$$\|u - u^a\| \le C_4\chi\tau, \quad \|v - v^a\| \le C_4\chi\tau, \tag{5.5a}$$

$$\|(u, v)\| \le C_4\chi\tau, \tag{5.5b}$$

$$\|u_N - u_N^a\| \le C_4\chi\tau\tau_k, \quad \|v_B - v_B^a\| \le C_4\chi\tau\tau_k. \tag{5.5c}$$

Proof. From (2.5), we have that

$$\begin{pmatrix} S^k & X^k \\ M & -I \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -Xs \\ r \end{pmatrix}, \tag{5.6}$$

while from (5.3), we get

$$\begin{pmatrix} S & X \\ M & -I \end{pmatrix} \begin{pmatrix} u^a \\ v^a \end{pmatrix} = \begin{pmatrix} -Xs \\ r \end{pmatrix}, \tag{5.7}$$

and therefore

$$\begin{pmatrix} S^k & X^k \\ M & -I \end{pmatrix} \begin{pmatrix} u^a \\ v^a \end{pmatrix} = \begin{pmatrix} -Xs + (S^k - S)u^a + (X^k - X)v^a \\ r \end{pmatrix}. \tag{5.8}$$

By (5.6) and (5.8) we obtain

$$\begin{pmatrix} S^k & X^k \\ M & -I \end{pmatrix} \begin{pmatrix} u^a - u \\ v^a - v \end{pmatrix} = \begin{pmatrix} (S^k - S)u^a + (X^k - X)v^a \\ 0 \end{pmatrix}. \tag{5.9}$$

Then, from (5.4) and Lemma 5.3, there is a constant C_4' independent of k and χ such that

$$\|(S^k - S)u^a + (X^k - X)v^a\| \le C_4'\chi\tau\tau_k. \tag{5.10}$$

Applying Lemma 3.2 to the linear system (5.9) with $a = (S^k - S)u^a + (X^k - X)v^a$, $b = 0$, we deduce that

$$\begin{aligned} \|D^k(u^a - u)\| &\le \sqrt{1 + 2\kappa}\|(X^k S^k)^{-1/2}((S^k - S)u^a + (X^k - X)v^a)\| \\ &\le \frac{\sqrt{1 + 2\kappa}C_4'}{\sqrt{\alpha_{\min}\rho_0}}\chi\tau\sqrt{\tau_k}, \end{aligned} \tag{5.11}$$

where $D^k = (X^k)^{-1/2}(S^k)^{1/2}$. Therefore,

$$\|u^a - u\| \le \|(D^k)^{-1}\|\|D^k(u^a - u)\| \le \frac{\sqrt{1 + 2\kappa}C_1C_4'}{\alpha_{\min}\rho_0}\chi\tau. \tag{5.12}$$

Then, from (5.11) and Lemma 5.2, we deduce that

$$\begin{aligned} \|u_N^a - u_N\| &\le \|(D_N^k)^{-1}\|\|D^k(u^a - u)\| \\ &= \|(X_N^k)(X_N^k S_N^k)^{-1/2}\|\|D^k(u^a - u)\| \\ &\le \frac{\sqrt{1 + 2\kappa}C_2C_4'}{\alpha_{\min}\rho_0}\chi\tau\tau_k. \end{aligned}$$

The bounds for $(v^a - v)$ and $(v^a_B - v_B)$ are obtained similarly. Since

$$\|(u, v)\| \le \|(u^a, v^a)\| + \|(u - u^a, v - v^a)\|,$$

relation (5.5) follows from Lemma 5.3 with an appropriate $C_4 > 0$. □

The next lemma, which is similar to Lemma 4.5 of Wright and Zhang [20], gives an estimate for the step length $\bar{\theta}$ along a (possibly approximate) fast step direction (u, v). The point (x, s) considered in the lemma represents either the main iterate (x^k, y^k) itself or one of the intermediate points generated by the **improve** procedure during the kth iteration. In what follows we use the following notations:

$$C'_5 = \rho_0(1 - \gamma) \min\{\alpha_{\max} - \alpha_{\min},\ \beta_{\max} - \beta_{\min}\},$$

$$C''_5 = \max\{2C_4^2,\ 4C_4(C_1 + C_2)\},$$

$$C_5 = C''_5 / C'_5.$$

LEMMA 5.5. *During the kth iteration of Algorithm* 2.1, *suppose* $(x, s, \tau, \alpha, \beta) \in \mathcal{N}$, *where*

$$\alpha = \alpha_{\min} + \gamma^t(\alpha_{\max} - \alpha_{\min}), \quad \beta = \beta_{\max} - \gamma^t(\beta_{\max} - \beta_{\min})$$

and the quantity t satisfies

$$C_5 \chi^2 \frac{\tau_k}{\gamma^t} \le \rho. \tag{5.13}$$

If $\bar{\theta}$ is computed from (b)–(e) *with $\lambda = 0$ and*

$$\alpha_+ = \alpha_{\min} + \gamma^{t+1}(\alpha_{\max} - \alpha_{\min}), \quad \beta_+ = \beta_{\max} - \gamma^{t+1}(\beta_{\max} - \beta_{\min}),$$

then

$$1 - \bar{\theta} \le C_5 \chi^2 \frac{\tau_k}{\gamma^t} \le \rho. \tag{5.14}$$

Proof. By definition, we have

$$\begin{aligned} X(\theta)s(\theta) &= Xs + \theta(Su + Xv) + \theta^2 Uv \\ &= Xs + \theta(Su^a + Xv^a) + \theta[S(u - u^a) + X(v - v^a)] + \theta^2 Uv \\ &= (1 - \theta)Xs + \theta[S(u - u^a) + X(v - v^a)] + \theta^2 Uv. \end{aligned} \tag{5.15}$$

Then from Lemmas 5.1, 5.2, and 5.4, we deduce that

$$\begin{aligned} &\|S(u - u^a) + X(v - v^a)\| \\ &\quad \le \|s_B\|\|u_B - u^a_B\| + \|x_B\|\|v_B - v^a_B\| + \|s_N\|\|u_N - u^a_N\| + \|x_N\|\|v_N - v^a_N\| \\ &\quad \le 2C_4(C_1 + C_2)\chi\tau\tau_k \le \frac{C''_5}{2}\chi^2\tau\tau_k. \end{aligned} \tag{5.16}$$

Let θ'' be the unique positive root of the quadratic equation

$$C'_5 \gamma^t (1 - \theta) - \frac{C''_5}{2}\chi^2\tau_k\theta - \frac{C''_5}{2}\chi^2\tau_k\theta^2 = 0.$$

Then, it is easily seen that $\theta'' \in (0,1)$ and

$$C_5'\gamma^t(1-\theta) - \frac{C_5''}{2}\chi^2\tau_k\theta - \frac{C_5''}{2}\chi^2\tau_k\theta^2 \geq 0 \ \text{ for all } \ \theta \in [0, \theta''].$$

Hence, if $\theta \in [0, \theta'']$, then from (5.15), (5.16), and Lemma 5.4, we deduce that

$$\begin{aligned}
&X(\theta)s(\theta) - \alpha_+(1-\theta)\tau p \\
&= (1-\theta)(Xs - \alpha_+\tau p) + \theta[S(u-u^a) + X(v-v^a)] + \theta^2 Uv \\
&= (1-\theta)(Xs - \alpha\tau p) + (1-\theta)(\alpha - \alpha_+)\tau p + \theta[S(u-u^a) + X(v-v^a)] + \theta^2 Uv \\
&\geq (1-\gamma)\gamma^t(\alpha_{\max} - \alpha_{\min})(1-\theta)\tau p - \theta\|S(u-u^k) + X(v-v^k)\|e - \theta^2\|Uv\|e \\
&\geq C_5'\gamma^t(1-\theta)\tau e - \frac{C_5''}{2}\chi\tau\tau_k\theta e - \frac{C_5''}{2}\chi^2\tau^2\theta^2 e \\
&\geq \tau[C_5'\gamma^t(1-\theta) - \frac{C_5''}{2}\chi^2\tau_k\theta - \frac{C_5''}{2}\chi^2\tau_k\theta^2]e \geq 0
\end{aligned}$$

and

$$\begin{aligned}
&\|X(\theta)s(\theta) - (1-\theta)\tau p\| - \beta_+(1-\theta)\tau \\
&\leq (1-\theta)\|Xs - \tau p\| + \theta\|S(u-u^a) + X(v-v^a)\| + \theta^2\|Uv\| - \beta_+(1-\theta)\tau \\
&\leq -(1-\gamma)\gamma^t(\beta_{\max} - \beta_{\min})(1-\theta)\tau + \theta\|S(u-u^a) + X(v-v^a)\| + \theta^2\|Uv\| \\
&\leq -C_5'\gamma^t(1-\theta)\tau + \frac{C_5''}{2}\chi^2\tau\tau_k\theta + \frac{C_5''}{2}\chi^2\tau^2\theta^2 \\
&\leq \tau\left[-C_5'\gamma^t(1-\theta) + \frac{C_5''}{2}\chi^2\tau_k\theta + \frac{C_5''}{2}\chi^2\tau_k\theta^2\right] \leq 0.
\end{aligned}$$

Therefore, $\bar{\theta} \geq \theta''$, which implies

$$\begin{aligned}
1 - \bar{\theta} \leq 1 - \theta'' &= \frac{1}{C_5'\gamma^t}\left(\frac{C_5''}{2}\chi^2\tau_k\theta'' + \frac{C_5''}{2}\chi^2\tau_k(\theta'')^2\right) \\
&\leq C_5(\chi^2\tau_k/\gamma^t) \leq \rho. \qquad \square
\end{aligned}$$

LEMMA 5.6. *There exists a constant $C_6 \in (0,1)$ such that*

$$\frac{\tau_{k+1}}{\gamma^{t_{k+1}}} \leq C_6\frac{\tau_k}{\gamma^{t_k}} \quad \textit{for all} \ \ k \geq 0. \tag{5.17}$$

Proof. See Lemma 4.6 of Wright and Zhang [20]. $\square$

We are ready to show that there exists a threshold value of τ_k/γ^{t_k} below which both the main algorithm and the procedure **improve** take only fast steps. By following the elegant proof technique of Theorem 5.1 of Wright and Zhang [20] and using Lemmas 5.3–5.6, we can prove the following lemma.

LEMMA 5.7. *Define*

$$\chi = \max\left\{1, \quad C_3\exp\left(\frac{C_4\rho}{1-\rho}\right)\right\},$$

and let K be the smallest index such that

$$C_5\chi^2\frac{\tau_K}{\gamma^{t_K+I}} \leq \rho.$$

Then, at the kth iteration with $k \geq K$, the fast branch is taken in the main algorithm and I fast steps are taken in the call to **improve**.

We end this section by showing that the sequence $\{\mu_k\}$ converges to zero with Q-order at least $I+2$. Since $\gamma_{\min}\rho_0\tau \leq \mu \leq (1+\beta_{\max})\tau$, this is equivalent to showing that for any $\epsilon > 0$,

$$\limsup_{k\to\infty} \frac{\tau_{k+1}}{{\tau_k}^{I+2-\epsilon}} = 0.$$

An equivalent characterization of the Q-order $I+2$ convergence is

$$\liminf_{k\to\infty} \frac{\ln \tau_{k+1}}{\ln \tau_k} \geq I+2 \tag{5.18}$$

(see Ortega and Rheinboldt [9] or Potra [10]). By using the proof technique from Theorem 5.2 of Wright and Zhang [20], we obtain the following theorem.

THEOREM 5.8. *Suppose that $\mathcal{F}^c$ is not empty and that during the safe branch λ is defined by* (3.6). *Then the sequence $\{\mu_k\}, k = 0, 1, \ldots,$ converges to zero with Q-order at least $I+2$.*

6. Concluding remarks.

- Our algorithm can solve all feasible P_*-matrix linear complementarity problems, without knowing the classification number κ in advance. The computational complexity of the algorithm depends on κ.
- The algorithm is new even for monotone LCP where it attains $O(n\ln(\epsilon_0/\epsilon))$-iteration complexity, which improves the $O(n^2\ln(\epsilon_0/\epsilon))$-iteration complexity of the algorithm of [20].
- The algorithm requires only one matrix factorization at each iteration, compared to two matrix factorizations of predictor-corrector algorithms (cf. [8, 11, 15, 13]).
- Asymptotically, the algorithm requires $I+1$ backsolves at each iteration to get fast local convergence of Q-order at least $I+2$ for problems having a strictly complementary solution.
- The algorithm easily can be extended to horizontal LCP and mixed LCP (see, i.e., [1]).

Acknowledgment. The authors would like to thank Mihai Anitescu for carefully reading the manuscript.

REFERENCES

[1] M. Anitescu, G. Lesaja, and F. A. Potra, *On the Equivalence Between Different Formulations and Algorithms for the $P_*(\kappa)$-LCP*, Reports on computational mathematics 71, Department of Mathematics, The University of Iowa, Iowa City, IA, June, 1995.

[2] C. C. Gonzaga, *The Largest Step Path Following Algorithm for Monotone Linear Complementary Problems*, Technical reports, Delft Technical University, Delft, The Netherlands, January, 1993.

[3] J. Ji and F. A. Potra, *An Infeasible-Interior-Point Method for the P_*-Matrix LCP*, Reports on computational mathematics 52, Department of Mathematics, The University of Iowa, Iowa City, IA, February, 1994.

[4] J. Ji, F. A. Potra, and R. Sheng, *A predictor–corrector method for solving the $P_*(\kappa)$-matrix LCP from infeasible starting points*, Optim. Methods Software, 6 (1995), pp. 109–126.

[5] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise, *A unified approach to interior point algorithms for linear complementarity problems*, Lecture Notes in Computer Science 538, Springer-Verlag, Berlin, New York, 1991.

[6] S. Mehrotra, *Asymptotic Convergence in a Generalized–Predictor–Corrector Method*, Technical report, Department of Industrial Engineering and Management Science, Northwestern University, Evanston, IL, October, 1992.

[7] J. Miao, *A quadratically convergent $O((1+\kappa)\sqrt{n}L)$-iteration algorithm for the $P_*(\kappa)$-matrix linear complementarity problem*, Math. Programming, 69 (1995), pp. 355–368.

[8] S. Mizuno, F. Jarre, and J. Stoer, *A unified approach to infeasible-interior-point algorithms via geometrical linear complementarity problems*, Appl. Math. Optim., 33 (1996), pp. 315–341.

[9] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[10] F. A. Potra, *On Q-order and R-order of convergence*, J. Optim. Theory Appl., 63 (1989), pp. 415–431.

[11] F. A. Potra, *An $O(nL)$ Infeasible-Interior-Point Algorithm for LCP with Quadratic Convergence*, Reports on computational mathematics 50, Department of Mathematics, The University of Iowa, Iowa City, IA, January, 1994.

[12] F. A. Potra and V. Ptak, *Nondiscrete induction and iterative processes*, Research Notes in Mathematics 103, Pitman, Boston, MA, 1984.

[13] F. A. Potra and R. Sheng, *Predictor-Corrector Algorithms for Solving $P_*(\kappa)$-Matrix LCP from Arbitrary Positive Starting Points*, Reports on computational mathematics 58, Department of Mathematics, The University of Iowa, Iowa City, IA, August, 1994.

[14] V. E. Shamanskii, *On a modification of Newton's method*, Ukrainian Math. J., 19 (1967), pp. 133–138, (in Russian).

[15] 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.

[16] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1964.

[17] S. J. Wright, *A path–following infeasible–interior–point algorithm for linear complementarity problems*, Optim. Methods Software, 2 (1993), pp. 79–106.

[18] S. J. Wright, *A path–following interior–point algorithm for linear and quadratic problems*, Ann. Oper. Res., 62 (1996), pp. 103–130.

[19] S. J. Wright, *An infeasible-interior-point algorithm for linear complementarity problems*, Math. Programming, 67 (1994), pp. 29–52.

[20] S. J. Wright and Y. Zhang, *A superquadratic infeasible–interior–point algorithm for linear complementarity problems*, Math. Programming, 73 (1996), pp. 269–289.

[21] X. Xu, *An $O(\sqrt{n}L)$-Iteration Large-Step Infeasible Path-Following Algorithm for Linear Programming*, Department of Management Sciences, The University of Iowa, Iowa City, IA, August, 1994.

[22] 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. 7, No. 2, pp. 336–346, May 1997

EFFICIENCY OF THE ANALYTIC CENTER CUTTING PLANE METHOD FOR CONVEX MINIMIZATION*

KRZYSZTOF C. KIWIEL[†]

Abstract. We consider the analytic center cutting plane method of Sonnevend and of Goffin et al. for minimizing a convex (possibly nondifferentiable) function subject to box constraints. At each iteration, accumulated subgradient cuts define a polytope that localizes the minimum. The objective and its subgradient are evaluated at the analytic center of this polytope to produce a cut that improves the localizing set. While complexity results have been recently established for several related methods, the question of whether the original method converges has remained open. We show that the method converges and establish its efficiency.

Key words. nondifferentiable optimization, cutting plane methods, analytic center, potential function

AMS subject classifications. 65K05, 90C25

PII. S1052623494275768

1. Introduction.

In this paper we consider the analytic center cutting plane method of [GHV92, Son88] for minimizing a convex (possibly nondifferentiable) function subject to box constraints. At each iteration, the method localizes the minimum via a polytope defined by accumulated subgradient cuts. The objective and its subgradient are evaluated at the analytic center of this polytope to produce a new cut. This method performs well in practice [BdMGV95, BGVdM93, GGSV94, GHV92], but so far its convergence has remained an open problem.

Recently, complexity results have been established for several related methods. The methods of [AtV95, Gof94, GLY94, GLY96, Luo94, Ye94] are restricted to convex feasibility problems. An extension of [AtV95] to optimization problems is given in [MiR93]. The method of [Nes95] for unconstrained minimization employs another potential function, whereas volumetric rather than analytic centers are used in [Vai96] for optimization and in [Ans94] for feasibility problems. The modifications of [Alt94, AlK96, Kiw96] have exploited the feasibility framework of [GLY94] in the context of optimization.

In this paper, we show that the original analytic center cutting plane method has the same efficiency as its modifications of [Kiw96]. There are, however, technical differences in their analyses (so that neither one subsumes the other).

Following [GLY96], our results can be extended to implementable methods that use approximate analytic centers. To save space, we omit such extensions (also because the accuracy of a close approximate center can be improved quadratically by Newton steps [AtV92, AtV95, Gof94, GLY96, GoV93, NeN94, Nes95, RaM94, Ren88, Vai90]).

The paper is organized as follows. In section 2 we recall basic properties of analytic centers. To ease notation, a slightly modified method is introduced in section 3, and its efficiency is analyzed in sections 4–5. Implications for the original method are discussed in section 6.

* Received by the editors October 17, 1994; accepted for publication (in revised form) October 20, 1995. This research was supported by the State Committee for Scientific Research under grant 8S50502206.

http://www.siam.org/journals/siopt/7-2/27576.html

[†] Systems Research Institute, Newelska 6, 01–447 Warsaw, Poland (kiwiel@ibspan.waw.pl).

We use the following notation. The l_2, l_1, and l_∞ norms of $z \in \mathbb{R}^n$ are denoted by $|z| = (\sum_{i=1}^n z_i^2)^{1/2}$, $\|z\|_1 = \sum_{i=1}^n |z_i|$, and $\|z\|_\infty = \max_{i=1:n} |z_i|$, respectively; e_i is column i of the identity matrix I and e is the vector of ones (of varying dimensions).

2. Analytic centers of polytopes.

Let y^a denote the analytic center of a full-dimensional polytope

$$\Omega = \{y \in \mathbb{R}^{\bar{n}} : A^T y \le c\} = \{y \in \mathbb{R}^{\bar{n}} : s = c - A^T y \ge 0\}, \tag{2.1}$$

where $c \in \mathbb{R}^m$ and $A \in \mathbb{R}^{\bar{n} \times m}$; i.e.,

$$y^a = \arg\max \left\{ \prod_{j=1}^m (c_j - a_j^T y) : y \in \Omega \right\} = \arg\min\{\Psi_\Omega(y) : y \in \Omega\}, \tag{2.2}$$

where a_j denotes column j of A and $\Psi_\Omega(y) = -\sum_{j=1}^m \ln(c_j - a_j^T y)$ is the logarithmic barrier of Ω. Let $s^a = c - A^T y^a$, so $0 = \nabla\Psi_\Omega(y^a) = \sum_j a_j / s_j^a$. Define the *potential* of Ω as

$$P(\Omega) = \sum_{j=1}^m \ln(c_j - a_j^T y^a) = \sum_{j=1}^m \ln(s_j^a) = -\Psi_\Omega(y^a). \tag{2.3}$$

Changing the right-hand side of the last inequality, consider the new polytope

$$\Omega_\beta^+ = \{y \in \mathbb{R}^{\bar{n}} : a_j^T y \le c_j, j = 1\colon m-1, a_m^T y \le a_m^T y^a + \beta s_m^a\}, \tag{2.4}$$

where β is a parameter. The following lemma of [Kiw96] extends a result of [Ye92, Thm. 1] obtained for $\beta \ge 0$, whereas $\beta < 0$ will allow us to analyze deeper cuts.

LEMMA 2.1. *Let Ω and Ω_β^+ be full-dimensional polytopes of the forms* (2.1) *and* (2.4)*, respectively. Then*

$$P(\Omega_\beta^+) \le P(\Omega) - 1 + \beta. \tag{2.5}$$

Proof. The proof given in [Ye92, p. 9] for $\beta \ge 0$ holds also for $\beta < 0$. □

Adding a new inequality (say the $(m+1)$th), consider the polytope

$$\Omega_\beta^+ = \{y \in \mathbb{R}^{\bar{n}} : a_j^T y \le c_j, j = 1\colon m, a_{m+1}^T y \le a_{m+1}^T y^a + \beta\bar{r}\}, \tag{2.6}$$

where β is a parameter and

$$\bar{r} = \sqrt{a_{m+1}^T (A(S^a)^{-2}A^T)^{-1} a_{m+1}}, \tag{2.7}$$

where $S^a = \operatorname{diag}(s^a)$. Our next lemma generalizes [Ye92, Thm. 2] to the case $\beta < 0$.

LEMMA 2.2. *Let Ω and Ω_β^+ be full-dimensional polytopes of the forms* (2.1) *and* (2.6)*, respectively, and let $\bar{\alpha} = 1.5 - \ln 4 > 0$ (≈ 0.1137). Then*

$$P(\Omega_\beta^+) \le P(\Omega) + \ln\bar{r} - \bar{\alpha} + \max\{\beta, 0\}. \tag{2.8}$$

Proof. For $\beta \ge 0$, see the proof of [Ye92, Thm. 2]. When $\beta < 0$, we have $P(\Omega_\beta^+) \le P(\Omega_0^+)$ from the definition of max-potential; cf. (2.1)–(2.3) and below. □

We shall also exploit strict monotonicity of the potential with respect to the right-hand sides.

LEMMA 2.3. *Let $y(c)$ be the analytic center of a full-dimensional polytope $\Omega_c = \{y : A^T y \le c\}$ with potential $\Pi(c) := P(\Omega_c)$ and barrier $\psi(y;c) = -\sum_{j=1}^m \ln(c_j - a_j^T y)$, i.e., $y(c) = \arg\min_y \psi(y;c)$ and $\Pi(c) = -\psi(y(c);c)$ (cf.* (2.1)–(2.3)*). Let $s(c) = c - A^T y(c)$ and $S(c) = \operatorname{diag}(s(c))$. Then $\nabla\Pi(c) = S^{-1}(c)e > 0$, i.e., $P(\Omega_c)$ strictly increases with c.*

Proof. Use $-\nabla\Pi(c) = \nabla_y \psi(y(c);c)^T \nabla y(c) + \nabla_c \psi(y(c);c)$ with $\nabla_y \psi(y(c);c) = 0$. One may also use the original idea of [FiM68]: if $\bar c > c$ then $y(c) \in \Omega_{\bar c}$, and hence

$$\Pi(\bar c) = \sum_{j=1}^m \ln(\bar c_j - a_j^T y(\bar c)) \ge \sum_{j=1}^m \ln(\bar c_j - a_j^T y(c)) > \sum_{j=1}^m \ln(c_j - a_j^T y(c)) = \Pi(c). \quad \square$$

We may add that several related results on the effect of adding and shifting constraints are given in [dHRT94]; these results generalize some of those in [Ye92].

3. An analytic center cutting plane method. To simplify notation as in [Alt94, Kiw96], we consider first the *canonical convex problem*

$$f_* = \min\{f(z) : 0 \le z \le e\}, \tag{3.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex,

$$f_* \ge 0, \quad \text{and} \quad f(\tfrac{1}{2}e) = 1. \tag{3.2}$$

We assume that we can evaluate f and its subgradient $g(z) \in \partial f(z)$ at each $z \in (0,1)^n$. (We denote the vector of variables by z because, in order to work in the space of the epigraph of f, we shall need to augment z with a "vertical" coordinate to form the vector y employed in the preceding section.)

Constraints of the form $z^{\text{low}} \le z \le z^{\text{up}}$ can be transformed into $0 \le z \le e$ by shifting and scaling. Also the (seemingly more general) problem

$$\tilde f_* = \min\{\tilde f(z) : 0 \le z \le e\} \tag{3.3}$$

with a convex objective $\tilde f : \mathbb{R}^n \to \mathbb{R}$ and subgradient mapping $\tilde g(\cdot) \in \partial \tilde f(\cdot)$ can be put into the canonical form (cf. (3.2)) with

$$f(\cdot) = \frac{2}{\|\tilde g(\frac{1}{2}e)\|_1}[\tilde f(\cdot) - \tilde f(\tfrac{1}{2}e)] + 1 \qquad \text{and} \qquad g(\cdot) = \frac{2}{\|\tilde g(\frac{1}{2}e)\|_1}\tilde g(\cdot). \tag{3.4}$$

Indeed, for $z^0 = \frac{1}{2}e$ and $\tilde g^0 \in \partial \tilde f(z^0)$, the subgradient inequality yields

$$\tilde f_* \ge \min\{\tilde f(z^0) + (z - z^0)^T \tilde g^0 : 0 \le z \le e\} = \tilde f(z^0) - \|\tilde g^0\|_1/2. \tag{3.5}$$

Conditions (3.2) will simplify our analysis only, and it will be seen that the transformation (3.4) need not be applied in practice.

Let $\mathcal{Z} = [0,1]^n$ denote the feasible set of (3.1) and $f_{\mathcal{Z}}$ its essential objective, i.e., $f_{\mathcal{Z}}(z) = f(z)$ if $z \in \mathcal{Z}$; $f_{\mathcal{Z}}(z) = \infty$ if $z \notin \mathcal{Z}$. For any tolerance $\bar\epsilon > 0$, an $\bar\epsilon$-*solution* of (3.1) (i.e., z such that $f_{\mathcal{Z}}(z) \le f_* + \bar\epsilon$) can be found in the set

$$\mathcal{Y}_{\bar\epsilon} = \{y \in [0,1]^{\bar n} : y = (z,\zeta), f(z) \le \zeta, \zeta \le f_* + \bar\epsilon\},$$

where $\bar n = n + 1$; a point $y \in \mathbb{R}^{\bar n}$ is denoted by (z,ζ), i.e., $\zeta = y_{\bar n}$. Thus, using the original bounds $0 \le z \le e$ and the vertical bounds $0 \le \zeta \le 1$ due to (3.2), the method may start its search from the initial polytope $\Omega^0 = [0,1]^{\bar{n}}$ that localizes points in $\operatorname{epi} f_{\mathcal{Z}} = \{(z,\zeta) : f_{\mathcal{Z}}(z) \leq \zeta\}$ (the epigraph of $f_{\mathcal{Z}}$) with the lowest vertical coordinate ζ. At iteration $k \geq 1$, the localizing polytope Ω^k is obtained from Ω^0 by appending accumulated subgradient cuts and by replacing the horizontal cut $\zeta \leq 1$ with the objective cut $\zeta \leq f_{\text{rec}}^k$, where f_{rec}^k is the best f-value obtained so far.

ALGORITHM 3.1 (for the canonical problem).

Step 0 (Initiation). Set $A^0 = (I, -I) \in \mathbb{R}^{\bar{n} \times 2\bar{n}}$, $c^0 = \binom{e}{0} \in \mathbb{R}^{2\bar{n}}$, $f_{\text{rec}}^0 = 1$ and $k = 0$.

Step 1 (Center computation). Find the analytic center $y^k = (z^k, \zeta_k)$ of the polytope $\Omega^k = \{y \in \mathbb{R}^{\bar{n}} : (A^k)^T y \leq c^k\}$, given by $m_k = 2\bar{n} + k$ inequalities. Set $s^k = c^k - (A^k)^T y^k$.

Step 2 (Stopping criterion). If z^k is a satisfactory approximate solution, then stop.

Step 3 (Cut generation). Find $f(z^k)$ and $g^k \in \partial f(z^k)$. *Generate a subgradient cut*: set

$$a_{m_k+1} = \begin{pmatrix} g^k \\ -1 \end{pmatrix} \quad \text{and} \quad c_{m_k+1} = (g^k)^T z^k - f(z^k), \tag{3.6}$$

$A^{k+1} = (A^k, a_{m_k+1})$ and $c^{k+1} = \binom{c^k}{c_{m_k+1}}$. Set $f_{\text{rec}}^{k+1} = \min\{f(z^k), f_{\text{rec}}^k\}$. If $f(z^k) < c_{\bar{n}}^k$, then *lower the horizontal cut*: set $c_{\bar{n}}^{k+1} = f_{\text{rec}}^{k+1}$.

Step 4. Increase k by 1 and go to Step 1.

Remark 3.2. Due to Step 0 and (3.2), $y^0 = \frac{1}{2}e$ and $c_{\bar{n}}^0 = 1 = f_{\text{rec}}^0 = f(z^0)$. By induction, $c_{\bar{n}}^k = f_{\text{rec}}^k = \min_{j=0:k} f(z^j)$ (cf. Step 3) and $y^k \in \Omega^k \subset \Omega^0 = [0,1]^{\bar{n}}$ for all k, since $\Omega^{k+1} \subset \Omega^k$. Note that constraint $\bar{n}$ of Ω^k is $\zeta \leq f_{\text{rec}}^k$ (the horizontal cut), and $f_{\text{rec}}^k \leq 1$. As in [GHV92], if Step 1 finds an underestimate $f_{\text{low}}^k \leq f_*$, then Step 2 may terminate if $f_{\text{rec}}^k - f_{\text{low}}^k \leq \bar{\epsilon}$, in which case $z_{\text{rec}}^k \in \operatorname{Arg\,min}\{f(z) : z \in \{z^j\}_{j=0}^{k-1}\}$ is an $\bar{\epsilon}$-solution; we let $z_{\text{rec}}^0 = z^0$.

4. Potential reduction. In this section we establish bounds on changes in the potential $P(\Omega^k)$. Such bounds are crucial for the complexity analysis presented in section 5.

Due to Step 3, we distinguish two cases in bounding the potential of

$$\Omega^{k+1} = \{y : a_j^T y \leq c_j, j \in \{1\colon m_k\} \setminus \{\bar{n}\}, a_{\bar{n}}^T y \leq c_{\bar{n}}^{k+1}, a_{m_k+1}^T y \leq c_{m_k+1}\}, \tag{4.1}$$

where

$$c_{\bar{n}}^{k+1} = f_{\text{rec}}^{k+1} \qquad \text{and} \qquad c_{m_k+1} = a_{m_k+1}^T y^k + \zeta_k - f(z^k). \tag{4.2}$$

Recall that Ω^{k+1} is obtained from Ω^k by appending the subgradient cut $a_{m_k+1}^T y \leq c_{m_k+1}$ and possibly lowering the horizontal cut $\zeta \leq f_{\text{rec}}^k$ to $\zeta \leq f_{\text{rec}}^{k+1}$. Let (cf. (2.7))

$$r_k = \sqrt{a_{m_k+1}^T (A^k (S^k)^{-2} (A^k)^T)^{-1} a_{m_k+1}}. \tag{4.3}$$

LEMMA 4.1. *If* $f(z^k) \geq \zeta_k$ *then*

$$P(\Omega^{k+1}) \leq P(\Omega^k) + \ln r_k - \bar{\alpha}. \tag{4.4}$$

Proof. By (4.2), $c_{m_k+1} = a_{m_k+1}^T y^k + \beta_k r_k$ with $\beta_k = [\zeta_k - f(z^k)]/r_k \leq 0$, so for

$$\tilde{\Omega}^{k+1} = \{y : (A^k)^T y \leq c^k, a_{m_k+1}^T y \leq a_{m_k+1}^T y^k + \beta_k r_k\},$$

since $\Omega^k = \{y : (A^k)^T y \leq c^k\}$, (2.8) yields $P(\tilde{\Omega}^{k+1}) \leq P(\Omega^k) + \ln r_k - \bar{\alpha}$. Since Ω^{k+1} is obtained from $\tilde{\Omega}^{k+1}$ by replacing $c_{\bar{n}}^k = f_{\rm rec}^k$ with $c_{\bar{n}}^{k+1} = f_{\rm rec}^{k+1} \leq c_{\bar{n}}^k$ (cf. Step 3), we have $P(\Omega^{k+1}) \leq P(\tilde{\Omega}^{k+1})$ (Lem. 2.3) and, hence, (4.4). □

LEMMA 4.2. *If $f(z^k) < \zeta_k$ then* (4.4) *holds.*

Proof. If $f(z^k) < \zeta_k$, then (cf. Step 3) $c_{\bar{n}}^{k+1} = f_{\rm rec}^{k+1} = f(z^k) < \zeta_k < c_{\bar{n}}^k = f_{\rm rec}^k$ (since $\zeta \leq f_{\rm rec}^k$ is constraint $\bar{n}$ of Ω^k). Let $\delta = \zeta_k - f(z^k)$ (note that $\delta > 0$). By (4.2),

$$c_{\bar{n}}^{k+1} = c_{\bar{n}}^k - \delta - (c_{\bar{n}}^k - \zeta_k) \qquad \text{and} \qquad c_{m_k+1} = a_{m_k+1}^T y^k + \delta. \tag{4.5}$$

We now define a parametric family of polytopes whose potentials may be compared to those of Ω^k and Ω^{k+1}. For each $t \in [0, 1]$, let $\bar{y}(t)$ be the analytic center of $\bar{\Omega}(t) = \{y : (A^{k+1})^T y \leq c(t)\}$ with potential $P(\bar{\Omega}(t)) =: \bar{\Pi}(t)$ and slack $\bar{s}(t) = c(t) - (A^{k+1})^T \bar{y}(t)$, where $c(t) \in \mathbb{R}^{m_k+1}$ has components $c_j(t) = c_j$ for $j \in \{1{:}\, m_k\} \setminus \{\bar{n}\}$, $c_{\bar{n}}(t) = c_{\bar{n}}^k - t\delta$, $c_{m_k+1}(t) = a_{m_k+1}^T y^k + t\delta$. Then $\bar{\Omega}(0) = \{y : (A^k)^T y \leq c^k, a_{m_k+1}^T y \leq a_{m_k+1}^T y^k\}$, so $P(\bar{\Omega}(0)) \leq P(\Omega^k) + \ln r_k - \bar{\alpha}$ (cf. (2.8)). Next, $P(\bar{\Omega}(1)) = P(\bar{\Omega}(0)) + \int_0^1 \bar{\Pi}'(t)\, dt$ with $\bar{\Pi}'(t) = [\bar{S}^{-1}(t)e]^T c'(t) = -\delta/\bar{s}_{\bar{n}}(t) + \delta/\bar{s}_{m_k+1}(t)$ from Lemma 2.3, since $c'(t) = -\delta e_{\bar{n}} + \delta e_{m_k+1}$. But, since $\bar{y}$ is the analytic center of $\bar{\Omega}(t)$, we have $\nabla\Psi_{\bar{\Omega}(t)}(\bar{y}(t)) = \sum_j \bar{s}_j^{-1}(t) a_j = 0$, where a_js are columns of $A^{k+1} = \Big[I, -I, \binom{g^0}{-1}, \dots, \binom{g^k}{-1}\Big]$, so from the last row $\bar{s}_{\bar{n}}^{-1}(t) - \bar{s}_{2\bar{n}}^{-1}(t) - \sum_{j=2\bar{n}+1}^{m_k} \bar{s}_j^{-1}(t) = 0$. Therefore, $\bar{\Pi}'(t) = -\sum_{j=2\bar{n}}^{m_k} \delta/\bar{s}_j(t) < 0$ and hence $P(\bar{\Omega}(1)) < P(\bar{\Omega}(0))$. Finally, Ω^{k+1} is obtained from $\bar{\Omega}(1)$ by replacing $c_{\bar{n}}^k - \delta$ with $c_{\bar{n}}^k - \delta - (c_{\bar{n}}^k - \zeta_k) < c_{\bar{n}}^k - \delta$ (cf. (4.5)), so $P(\Omega^{k+1}) < P(\bar{\Omega}(1))$ (Lem. 2.3). Collecting these estimates yields the result. □

5. Convergence and complexity. In this section, using results of [Nes95] as in [GLY94], we derive an efficiency estimate for Algorithm 3.1 by showing that $P(\Omega^k)$ grows more slowly than $2\bar{n} + k$. In view of Lemmas 4.1–4.2, this means finding upper bounds on r_k. This is achieved by using a construction due to [Nes95] which bounds $A^k(S^k)^{-2}(A^k)^T$ from below by a certain matrix B^k which is simple enough to handle.

We assume that the algorithm does not terminate. Recall that at Step 3, the matrix

$$A^k = [I, -I, a_{2\bar{n}+1}, \dots, a_{2\bar{n}+k}] = \left[I, -I, \begin{pmatrix} g^0 \\ -1 \end{pmatrix}, \dots, \begin{pmatrix} g^{k-1} \\ -1 \end{pmatrix}\right] \tag{5.1}$$

has $m_k = 2\bar{n} + k$ columns ($A^0 = (I, -I)$). Let

$$B^k = 8I + \sum_{j=1}^{k} a_{2\bar{n}+j} a_{2\bar{n}+j}^T / \|a_{2\bar{n}+j}\|_1^2. \tag{5.2}$$

We now modify a series of technical results from [GLY94, Kiw96]. First, we bound the slacks s^k.

LEMMA 5.1. *We have $s_j^k \in (0, 1)$ for $j = 1{:}\, 2\bar{n}$ and $s_j^k \in (0, \|a_j\|_1)$ for $j = 2\bar{n} + 1{:}\, 2\bar{n} + k$.*

Proof. Since y^k is the analytic center of $\Omega^k \subset \Omega^0$, $s^k = c^k - (A^k)^T y^k > 0$ and $y^k \in (0, 1)^{\bar{n}}$. For $j = 1{:}\, \bar{n} - 1$, $s_j^k = 1 - y_j^k < 1$ and $s_{\bar{n}}^k = c_{\bar{n}}^k - y_{\bar{n}}^k < 1$, since

$c^k_{\bar n} = f^k_{\rm rec} \le f(z^0) = 1$ (cf. Remark 3.2). For $j = \bar n + 1\colon 2\bar n$, $s^k_j = y^k_{j-\bar n} < 1$. For $j = 2\bar n + 1\colon 2\bar n + k$ and $l = j - 2\bar n - 1$, $\zeta_k < c^k_{\bar n} = f^k_{\rm rec} \le f(z^l)$ by Remark 3.2 (since $f^k_{\rm rec} = \min_{i\le k} f(z^i)$ and $l \le k$) and (cf. (3.6))

$$s^k_j = c_j - a^T_j y^k = (g^l)^T z^l - f(z^l) - (g^l)^T z^k + \zeta_k = \zeta^k - f(z^l) + (g^l)^T(z^l - z^k),$$

so

$$s^k_j < (g^l)^T(z^l - z^k) \le \|g^l\|_1\|z^l - z^k\|_\infty < \|g^l\|_1 < \|a_j\|_1,$$

where the first inequality follows $\zeta_k < f(z^l)$, the second one is from Hölder's inequality, the third one is from $z^l, z^k \in (0,1)^n$, and the fourth one is from $a_j = \binom{g^l}{-1}$ (cf. (5.1)). □

We now relate $A^k(S^k)^{-2}(A^k)^T$ to B^k (cf. (5.2)) as in [Nes95] and [GLY94].

LEMMA 5.2. *$A^k(S^k)^{-2}(A^k)^T \succeq B^k$, i.e., $A^k(S^k)^{-2}(A^k)^T - B^k$ is positive semidefinite.*

Proof. Let $Y^k = \operatorname{diag}(y^k)$. Then

$$\begin{aligned} A^k(S^k)^{-2}(A^k)^T &= \operatorname{diag}(1 - y^k_1, \ldots, 1 - y^k_{\bar n-1}, c^k_{\bar n} - y^k_{\bar n})^{-2} + (Y^k)^{-2} + \sum_{j=1}^k \frac{a_{2\bar n+j}a^T_{2\bar n+j}}{(s^k_{2\bar n+j})^2} \\ &\succeq (I - Y^k)^{-2} + (Y^k)^{-2} + \sum_{j=1}^k a_{2\bar n+j}a^T_{2\bar n+j}/\|a_{2\bar n+j}\|_1^2 \\ &\succeq 8I + \sum_{j=1}^k a_{2\bar n+j}a^T_{2\bar n+j}/\|a_{2\bar n+j}\|_1^2 = B^k, \end{aligned}$$

where the first relation follows from the forms of $s^k = c^k - (A^k)^T y^k$, c^k and A^k (cf. (5.1), $c^0 = \binom{e}{0}$ at Step 0, and Remark 3.2), the second one from $0 < c^k_{\bar n} - y^k_{\bar n} \le 1 - y^k_{\bar n}$ (since $c^k_{\bar n} \le f(z^0) = 1$; cf. Remark 3.2) and Lemma 5.1, and the third one from $y^k \in (0,1)^{\bar n}$ and $\min_{0<t<1} t^{-2} + (1-t)^{-2} = 8$. □

For each $k \ge 0$, let

$$\omega_k = \sqrt{a^T_{m_k+1}(B^k)^{-1}a_{m_k+1}} \tag{5.3}$$

and

$$\nu_k = \|a_{m_k+1}\|_1 = 1 + \|g^k\|_1 \tag{5.4}$$

(cf. (3.6)). By (4.3) and Lemma 5.2,

$$r_k \le \omega_k \qquad \text{for all } k. \tag{5.5}$$

Thus, upper bounds on the sequence $\{\omega_k\}$ will lead to upper bounds on the sequence $\{r_k\}$. The proof of our next lemma is patterned after that of [GLY94, Lem. 3.4], which in turn adapted a result of [Nes95].

LEMMA 5.3.

$$\sum_{j=0}^k \omega_j^2/\nu_j^2 \le \frac{\bar n}{8\ln\frac{9}{8}} \ln\left(1 + \frac{k+1}{8\bar n}\right). \tag{5.6}$$

Proof. By (5.2), (5.3), and (5.4),

$$\det B^{k+1} = \det(B^k + a_{m_k+1}a_{m_k+1}^T/\nu_k^2) = (1+\omega_k^2/\nu_k^2)\det B^k,$$

so

$$\ln\det B^{k+1} = \ln\det B^k + \ln(1+\omega_k^2/\nu_k^2). \tag{5.7}$$

Since $B^k \succeq 8I$ (cf. (5.2)) and $|\cdot| \le \|\cdot\|_1$, by (5.3)

$$\omega_k^2 = a_{m_k+1}^T(B^k)^{-1}a_{m_k+1} \le |a_{m_k+1}|^2/8 \le \|a_{m_k+1}\|_1^2/8,$$

so $\omega_k^2/\nu_k^2 \ge \frac{1}{8}$ (cf. (5.4)). Using Nesterov's [Nes95] inequality $\ln(1+\alpha\beta) \ge \alpha\ln(1+\beta)$ for all $\alpha \in [0,1]$, $\beta \ge 0$ with $\alpha = 8\omega_k^2/\nu_k^2$ and $\beta = \frac{1}{8}$, we get $\ln(1+\omega_k^2/\nu_k^2) \ge 8\ln(\frac{9}{8})\omega_k^2/\nu_k^2$. Combining this with (5.7) yields

$$\ln\det B^{k+1} \ge \ln\det B^k + 8\ln(\tfrac{9}{8})\omega_k^2/\nu_k^2.$$

Hence, by summing,

$$\ln\det B^{k+1} \ge \ln\det B^0 + 8\ln(\tfrac{9}{8})\sum_{j=0}^{k}\omega_j^2/\nu_j^2 = \bar{n}\ln 8 + 8\ln(\tfrac{9}{8})\sum_{j=0}^{k}\omega_j^2/\nu_j^2 \tag{5.8}$$

(using $B^0 = 8I$ and $\det B^0 = 8^{\bar{n}}$). But $(\det B^{k+1})^{1/\bar{n}} \le \operatorname{tr} B^{k+1}/\bar{n}$, where (cf. (5.2))

$$\operatorname{tr} B^{k+1} = 8\bar{n} + \sum_{j=0}^{k}|a_{m_j+1}|^2/\|a_{m_j+1}\|_1^2 \le 8\bar{n} + k + 1,$$

so

$$\frac{1}{\bar{n}}\ln\det B^{k+1} \le \ln\frac{\operatorname{tr} B^{k+1}}{\bar{n}} \le \ln\left(8 + \frac{k+1}{\bar{n}}\right).$$

Combining this with (5.8) yields

$$8\ln(\tfrac{9}{8})\sum_{j=0}^{k}\omega_j^2/\nu_j^2 \le \bar{n}\ln\left(8+\frac{k+1}{\bar{n}}\right) - \bar{n}\ln 8,$$

which implies (5.6). □

We now bound $P(\Omega^k)$ from below as in [GLY94, Lem. 3.1], but we scale the polytope, since we work with $\|\cdot\|_1$ instead of $|\cdot|$ without assuming that $\|a_j\|_1 = 1$.

LEMMA 5.4. *Consider the scaled polytope* $\tilde{\Omega}^k = \{y : \tilde{a}_j^T y \le \tilde{c}_j^k, j = 1\colon m_k\}$, *where* $\tilde{a}_j = a_j/\|a_j\|_1$ *and* $\tilde{c}_j^k = c_j^k/\|a_j\|_1$. *Let* ϵ *satisfy* $0 < \epsilon \le \tilde{c}_j^k - \tilde{a}_j^T\bar{y}$, $j = 1\colon m_k$ *for some* $\bar{y}$. *Then* $P(\Omega^k) - \sum_{j=2\bar{n}+1}^{m_k}\ln\|a_j\|_1 = P(\tilde{\Omega}^k) \ge m_k\ln\epsilon$.

Proof. Since y^k is the analytic center of both Ω^k and $\tilde{\Omega}^k$ (cf. (2.1)–(2.3)),

$$P(\tilde{\Omega}^k) = \sum_{j=1}^{m_k}\ln(\tilde{c}_j^k - \tilde{a}_j^T y^k) \ge \sum_{j=1}^{m_k}\ln(\tilde{c}_j^k - \tilde{a}_j^T\bar{y}) \ge \sum_{j=1}^{m_k}\ln\epsilon$$

and $P(\tilde{\Omega}^k) = P(\Omega^k) - \sum_{j=1}^{m_k}\ln\|a_j\|_1$, where $\|a_j\|_1 = 1$ for $j = 1\colon 2\bar{n}$ (cf. (5.1)). □

The following lemma (whose proof is modelled after that of [GLY94, Thm. 3.1]) shows that for large k, Ω^k cannot contain a point with "large" slacks; this will be translated into bounds on $f_{\text{rec}}^k - f_*$ in the proof of the subsequent theorem.

LEMMA 5.5. *Let ϵ satisfy the assumption of Lemma 5.4. Then*

$$\frac{\epsilon^2}{\bar{n}} \le \frac{\frac{1}{2} + \ln\left(1 + \frac{k}{8\bar{n}}\right)/8\ln\frac{9}{8}}{2\bar{n}+k} \exp\left(-2\bar{\alpha}\frac{k}{2\bar{n}+k}\right). \tag{5.9}$$

Proof. Using (4.4), we get

$$P(\Omega^{k+1}) \le P(\Omega^0) + \sum_{j=0}^{k}(\ln r_j - \bar{\alpha}) = P(\Omega^0) + \sum_{j=0}^{k} \ln r_j - \bar{\alpha}(k+1).$$

However, $P(\Omega^0) = 2\bar{n}\ln\frac{1}{2}$ at Step 0 and $m_{k+1}\ln\epsilon \le P(\Omega^{k+1}) - \sum_{j=0}^{k}\ln\nu_j$ (cf. (5.4)) from Lemma 5.4 with k increased by 1 (temporarily for ease of notation), so

$$m_{k+1}\ln\epsilon + \bar{\alpha}(k+1) \le 2\bar{n}\ln\frac{1}{2} + \frac{1}{2}\sum_{j=0}^{k}\ln(r_j^2/\nu_j^2).$$

Hence,

$$\begin{aligned}\ln\epsilon + \frac{\bar{\alpha}(k+1)}{2\bar{n}+k+1} &\le \frac{1}{2(2\bar{n}+k+1)}\left[2\bar{n}\ln\frac{1}{4} + \sum_{j=0}^{k}\ln(r_j^2/\nu_j^2)\right] \\ &\le \frac{1}{2}\ln\frac{2\bar{n}\frac{1}{4} + \sum_{j=0}^{k} r_j^2/\nu_j^2}{2\bar{n}+k+1} \\ &\le \frac{1}{2}\ln\frac{\bar{n}/2 + \sum_{j=0}^{k}\omega_j^2/\nu_j^2}{2\bar{n}+k+1} \\ &\le \frac{1}{2}\ln\frac{\bar{n}/2 + \bar{n}\ln\left(1+\frac{k+1}{8\bar{n}}\right)/8\ln\frac{9}{8}}{2\bar{n}+k+1},\end{aligned}$$

where the second inequality follows from the concavity of $\ln(\cdot)$, the third one from (5.5), and the fourth one from Lemma 5.3. Thus,

$$\frac{\epsilon^2}{\bar{n}} \le \frac{\frac{1}{2} + \ln\left(1+\frac{k+1}{8\bar{n}}\right)/8\ln\frac{9}{8}}{2\bar{n}+k+1}\exp\left(-2\frac{\bar{\alpha}(k+1)}{2\bar{n}+k+1}\right),$$

and the assertion follows by replacing $k+1$ by k. ∎

Following [Kiw96], we now present an efficiency estimate for Algorithm 3.1.

THEOREM 5.6. *Let $L_\infty = \sup\{\|g\|_1 : g \in \partial f(z), z \in (0,1)^n\}$. Then*

$$f_{\text{rec}}^k - f_* \le 2(1+L_\infty)\sqrt{\frac{\frac{\bar{n}}{2} + \bar{n}\ln\left(1+\frac{k}{8\bar{n}}\right)/8\ln\frac{9}{8}}{2\bar{n}+k}}\exp\left(-\bar{\alpha}\frac{k}{2\bar{n}+k}\right). \tag{5.10}$$

Proof. Suppose $f_{\text{rec}}^k > f_*$. In view of Lemma 5.5, it suffices to show that $\epsilon = (f_{\text{rec}}^k - f_*)/2(1+L_\infty)$ satisfies the assumption of Lemma 5.4. Note that $\epsilon < \frac{1}{4}$, since $L_\infty \ge \|g^0\|_1 \ge 2[f(z^0) - f_*]$ as in (3.5). Let $z^* \in \operatorname{Arg\,min} f_{\mathcal{Z}}$, $\bar{\zeta} = f_{\text{rec}}^k - \epsilon$, $\bar{y} = (\bar{z}, \bar{\zeta})$, and $\bar{s} = c^k - (A^k)^T\bar{y}$, where $\bar{z}_i = z_i^* + \epsilon$ if $z_i^* \le \frac{1}{2}$, $\bar{z}_i = z_i^* - \epsilon$ if $z_i^* > \frac{1}{2}$, and

$i = 1{:}\,n$. Since $z^* \in [0,1]^n$ and $\epsilon < \frac{1}{4}$, $\epsilon e \le \bar{z} \le (1-\epsilon)e$. Hence, $\bar{s}_j = 1 - \bar{y}_j \ge \epsilon$ for $j = 1{:}\,\bar{n}-1$, $\bar{s}_{\bar{n}} = c^k_{\bar{n}} - \bar{\zeta} \ge \epsilon$ since $c^k_{\bar{n}} = f^k_{\text{rec}}$ (cf. Remark 3.2), $\bar{s}_j = \bar{y}_{j-\bar{n}} \ge \epsilon$ for $j = \bar{n}+1{:}\,2\bar{n}-1$, and $\bar{s}_{2\bar{n}} = \bar{\zeta} \ge \epsilon$ since constraint $2\bar{n}$ of Ω^k is $-\zeta \le 0$, $\bar{\zeta} = f^k_{\text{rec}} - \epsilon$, and $2\bar{\epsilon} \le f^k_{\text{rec}} - f_* \le f^k_{\text{rec}}$ from $f_* \ge 0$ (cf. (3.2)). Also, $\|a_j\|_1 = 1$ for $j = 1{:}\,2\bar{n}$ (cf. (5.1)). Next, for $j \in \{2\bar{n}+1{:}\,2\bar{n}+k\}$ and $l = j - 2\bar{n} - 1$, we have $\|a_j\|_1 = 1 + \|g^l\|_1 \le 1 + L_\infty$ since $g^l \in \partial f(z^l)$, and

$$\bar{s}_j = \bar{\zeta} - f(z^l) - (g^l)^T(\bar{z} - z^l) \ge \bar{\zeta} - f(z^*) - (g^l)^T(\bar{z} - z^*),$$

using (3.6) and the subgradient inequality $f(z^*) \ge f(z^l) + (g^l)^T(z^* - z^l)$. Hence,

$$\bar{s}_j \ge \bar{\zeta} - f_* - \|g^l\|_1 \|\bar{z} - z^*\|_\infty \ge f^k_{\text{rec}} - f_* - \epsilon - L_\infty \epsilon = (1 + L_\infty)\epsilon,$$

since $\bar{\zeta} = f^k_{\text{rec}} - \epsilon$, $f(z^*) = f_*$, $\|\bar{z} - z^*\|_\infty = \epsilon$, and $f^k_{\text{rec}} - f_* = 2(1 + L_\infty)\epsilon$. Thus, $\bar{s}_j \ge \|a_j\|_1 \epsilon$ for $j = 1{:}\,m_k$, as required for invoking Lemma 5.4. □

COROLLARY 5.7. *$f^k_{\text{rec}} \downarrow f_*$ and every cluster point of $\{z^k_{\text{rec}}\}$ solves* (3.1).

Proof. The right side of (5.10) tends to zero as $k \to \infty$, since the square root term tends to zero and the exponential term is bounded. Thus, $f(z^k_{\text{rec}}) \downarrow f_*$. □

Remark 5.8. If problem (3.3) is put into the canonical form (3.1)–(3.2) via the transformation (3.4), then (5.10) yields the efficiency estimate

$$\tilde{f}^k_{\text{rec}} - \tilde{f}_* \le 3\tilde{L}_\infty \sqrt{\frac{\frac{\bar{n}}{2} + \bar{n} \ln\left(1 + \frac{k}{8\bar{n}}\right)/8 \ln \frac{9}{8}}{2\bar{n} + k}} \exp\left(-\bar{\alpha} \frac{k}{2\bar{n} + k}\right), \tag{5.11}$$

where $\tilde{f}^k_{\text{rec}} = \min_{j<k} \tilde{f}(z^j)$ and $\tilde{L}_\infty = \sup\{\|\tilde{g}\|_1 : \tilde{g} \in \partial \tilde{f}(z), z \in (0,1)^n\}$. Indeed, for $L_\infty = 2\tilde{L}_\infty / \|\tilde{g}^0\|_1$, $\tilde{f}^k_{\text{rec}} - \tilde{f}_* = (f^k_{\text{rec}} - f_*)\|\tilde{g}^0\|_1/2$ and $\|\tilde{g}^0\|_1(1 + 2\tilde{L}_\infty/\|\tilde{g}^0\|_1) \le 3\tilde{L}_\infty$. In fact (cf. [NeY79, Ex. II.1.15]),

$$\tilde{L}_\infty = \sup\{|\tilde{f}(z) - \tilde{f}(z')| / \|z - z'\|_\infty : z, z' \in [0,1]^n, z \ne z'\} \tag{5.12}$$

is the l_∞ Lipschitz constant of $\tilde{f}$ on $[0,1]^n$. We could use the l_2 constant $\tilde{L}_2$, obtained by replacing $\|\cdot\|_\infty$ with $|\cdot|$ in (5.12), since $\tilde{L}_\infty \le \sqrt{n}\tilde{L}_2$, but this would worsen our estimate.

6. Application to the original method. Compared with the original analytic center cutting plane method of [GHV92, Son88], Algorithm 3.1 employs the additional constraint $\zeta \ge 0$ related to the canonical conditions (3.2), which in turn might seem to require the transformation (3.4). Hence, we now address these issues in more detail.

Remark 6.1. If $\|g^0\|_1 \le 2$ (cf. (3.4)), then *the constraint $\zeta \ge 0$ can be omitted* at Step 0 by setting $a_{2\bar{n}} = \binom{g^0}{-1}$, $c_{2\bar{n}} = (g^0)^T z^0 - f(z^0)$, $A^1 = A^0$, $c^1 = c^0$, and $k = 1$. Then for $y \in \Omega^0$, $\zeta \ge f(z^0) - \|g^0\|_1/2 \ge 0$ from $a^T_{2\bar{n}} y \le c_{2\bar{n}}$ (cf. (3.5)), so $P(\Omega^0) \le 2\bar{n} \ln \frac{1}{2}$ and the proof of Theorem 5.6 goes through (with $m_k = 2\bar{n}+k-1$). In fact, omitting the constraint $\zeta \ge 0$ and setting $c^0_{\bar{n}} = f(z^0)$ ensures that the algorithm is *insensitive to the objective scaling.* Indeed, if $f(\cdot)$ and $g(\cdot)$ are replaced by $\breve{f}(\cdot) = \alpha f(\cdot)$ and $\breve{g}(\cdot) = \alpha g(\cdot)$, respectively, where $\alpha > 0$, then it generates $\breve{z}^k = z^k$, $\breve{\zeta}_k = \alpha \zeta^k$, $\breve{c}^k_{\bar{n}} = \alpha c^k_{\bar{n}}$, $\breve{g}^k = \alpha g^k$, and $\breve{c}_{2\bar{n}+j} = \alpha c_{2\bar{n}+j}$ for $j \le k$, whereas replacing $f(\cdot)$ by $f(\cdot) + \beta$ only increases ζ_k by β. Hence, the estimate (5.11) holds also if the algorithm is applied to $\tilde{f}$ *directly*, in which case the transformation (3.4) may be used *implicitly only* in the preceding proofs.

COROLLARY 6.2. *Consider the following method for problem* (3.3): *for* $k \geq 1$, *set*

$$y^k = (z^k, \zeta_k) = \arg\min_{z,\zeta} - \sum_{j=0}^{k-1} \ln[\zeta - \tilde{f}_j(z)] - \ln(\tilde{f}_{\rm rec}^k - \zeta) - \sum_{i=1}^{n} [\ln z_i + \ln(1 - z_i)],$$

$\tilde{f}_{\rm rec}^{k+1} = \min\{\tilde{f}(z^k), \tilde{f}_{\rm rec}^k\}$, $\tilde{f}_k(\cdot) = \tilde{f}(z^k) + (\tilde{g}^k)^T(\cdot - z^k)$ *with* $\tilde{g}^k \in \partial\tilde{f}(z^k)$, *starting from* $z^0 = \frac{1}{2}e$, $\tilde{f}_{\rm rec}^1 = \tilde{f}(z^0)$, $\tilde{f}_0(\cdot) = \tilde{f}(z^0) + (\tilde{g}^0)^T(\cdot - z^0)$, *and* $\tilde{g}^0 \in \partial\tilde{f}(z^0)$. *Then this method enjoys the efficiency estimate* (5.11)–(5.12).

Proof. This follows from Remark 6.1. □

Remark 6.3. Corollary 6.2 describes the method of [Son88] and that of [GHV92] (with unit weights).

Acknowledgments. I would like to thank the associate editor and the two anonymous referees for their valuable comments.

REFERENCES

[AlK96] A. ALTMAN AND K. C. KIWIEL, *A note on some analytic center cutting plane methods for convex feasibility and minimization problems*, Comput. Optim. Appl., 5 (1996), pp. 175–180.

[Alt94] A. ALTMAN, *Generalized Karmarkar Projective Methods for Convex Nondifferentiable Optimization Problems*, Ph.D. thesis, Systems Research Institute, Warsaw, Poland, 1994 (in Polish).

[Ans94] K. M. ANSTREICHER, *On Vaidya's Volumetric Cutting Plane Method for Convex Programming*, Tech. report, Department of Management Sciences, University of Iowa, Iowa City, IA, September, 1994.

[AtV92] D. S. ATKINSON AND P. M. VAIDYA, *A scaling technique for finding the weighted analytic center of a polytope*, Math. Programming, 57 (1992), pp. 163–192.

[AtV95] D. S. ATKINSON AND P. M. VAIDYA, *A cutting plane algorithm for convex programming that uses analytic centers*, Math. Programming, 69 (1995), pp. 1–43.

[BdMGV95] O. BAHN, O. DU MERLE, J.-L. GOFFIN, AND J.-PH. VIAL, *A cutting plane method from analytic centers for stochastic programming*, Math. Programming, 69 (1995), pp. 45–73.

[BGVdM93] O. BAHN, J.-L. GOFFIN, J.-PH. VIAL, AND O. DU MERLE, *Implementation and behavior of an interior point cutting plane algorithm for convex programming: An application to geometric programming*, Discrete Appl. Math., 49 (1993), pp. 3–23.

[dHRT94] D. DEN HERTOG, C. ROOS, AND T. TERLAKY, *Adding and deleting constraints in the logarithmic barrier method for linear programming*, in Advances in Optimization and Approximation, D.-Z. Du and J. Sun, eds., Nonconvex Optimization and Applications, Kluwer, Dordrecht, 1994, pp. 166–185.

[FiM68] A. V. FIACCO AND G. P. MCCORMICK, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Wiley, New York, 1968.

[GGSV94] J.-L. GOFFIN, J. GONDZIO, R. SARKISSIAN, AND J.-PH. VIAL, *Solving nonlinear multicommodity flow problems by the analytic center cutting plane method*, Tech. report, Département d'économie commerciale et industrielle, Université de Genève, Genève, Switzerland, October, 1994.

[GHV92] J.-L. GOFFIN, A. HAURIE, AND J.-PH. VIAL, *Decomposition and nondifferentiable optimization with the projective algorithm*, Management Sci., 37 (1992), pp. 284–302.

[GLY94] J.-L. GOFFIN, Z.-Q. LUO, AND Y. YE, *On the complexity of a column generation algorithm for convex or quasiconvex feasibility problems*, in Large Scale Optimization: State of the Art, W. W. Hager, D. W. Hearn, and P. M. Pardalos, eds., Kluwer, Dordrecht, 1994, pp. 182–191.

[GLY96] J.-L. GOFFIN, Z.-Q. LUO, AND Y. YE, *Complexity analysis of an interior point cutting plane method for convex feasibility problems*, SIAM J. Optim., (1996), pp. 638–652.

[Gof94] J.-L. GOFFIN, *Using the Primal Dual Infeasible Newton Method in the Analytic Center Method for Problems Defined by Deep Cutting Planes*, Tech. report, Faculty of Management, McGill University, Montreal, Quebec, September, 1994.

[GoV93] J.-L. Goffin and J.-Ph. Vial, *On the computation of weighted analytic centers and dual ellipsoids with the projective algorithm*, Math. Programming, 60 (1993), pp. 81–92.

[Kiw96] K. C. Kiwiel, *Complexity of some cutting plane methods that use analytic centers*, Math. Programming, 74 (1996), pp. 47–54.

[Luo94] Z.-Q. Luo, *Analysis of a Cutting Plane Method that Uses Analytic Center and Multiple Cuts*, Tech. report, Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, Canada, September, 1994.

[MiR93] J. E. Mitchell and S. Ramaswamy, *A long-step, cutting plane algorithm for linear and convex programming*, DSES Tech. report 37-93-387, Department of Decision Sciences & Engineering Systems, Rensselaer Polytechnic Institute, Troy, NY, August, 1993 (revised August, 1994).

[NeN94] Yu. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, Studies in Applied Mathematics 13, SIAM, Philadelphia, PA, 1994.

[Nes95] Yu. E. Nesterov, *Complexity estimates of some cutting plane methods based on the analytic barrier*, Math. Programming, 69 (1995), pp. 149–176.

[NeY79] A. S. Nemirovskii and D. B. Yudin, *Problem Complexity and Method Efficiency in Optimization*, Nauka, Moscow, 1979 (in Russian).

[RaM94] S. Ramaswamy and J. E. Mitchell, *On updating the analytic center after the addition of multiple cuts*, DSES Tech. report 37-94-423, Department of Decision Sciences & Engineering Systems, Rensselaer Polytechnic Institute, Troy, NY, October, 1994.

[Ren88] J. Renegar, *A polynomial-time algorithm, based on Newton's method, for linear programming*, Math. Programming, 40 (1988), pp. 59–93.

[Son88] G. Sonnevend, *New algorithms in convex programming based on the notion of "centre" (for systems of analytic inequalities) and on rational extrapolation*, in Trends in Mathematical Optimization, K. H. Hoffman, J.-B. Hiriart-Urruty, C. Lemaréchal, and J. Zowe, eds., International Series of Numerical Mathematics 84, Birkhäuser-Verlag, Basel, Switzerland, 1988, pp. 311–326.

[Vai90] P. M. Vaidya, *An algorithm for linear programming which requires $O(((m+n)n^2 + (m+n)^{1.5}n)L)$ arithmetic operations*, Math. Programming, 47 (1990), pp. 175–201.

[Vai96] P. M. Vaidya, *A new algorithm for minimizing convex functions over convex sets*, Math. Programming, 73 (1996), pp. 291–341.

[Ye92] Y. Ye, *A potential reduction algorithm allowing column generation*, SIAM J. Optim., 2 (1992), pp. 7–20.

[Ye94] Y. Ye, *Complexity Analysis of the Analytic Center Cutting Plane Method that Uses Multiple Cuts*, Tech. report, Department of Management Sciences, University of Iowa, Iowa City, IA, September, 1994.

SIAM J. OPTIM.
Vol. 7, No. 2, pp. 347–366, May 1997

PENALTY/BARRIER MULTIPLIER METHODS FOR CONVEX PROGRAMMING PROBLEMS*

AHARON BEN-TAL[†] AND MICHAEL ZIBULEVSKY[†]

Abstract. We study a class of methods for solving convex programs, which are based on nonquadratic augmented Lagrangians for which the penalty parameters are functions of the multipliers. This gives rise to Lagrangians which are nonlinear in the multipliers. Each augmented Lagrangian is specified by a choice of a penalty function φ and a penalty-updating function π. The requirements on φ are mild and allow for the inclusion of most of the previously suggested augmented Lagrangians. More importantly, a new type of penalty/barrier function (having a logarithmic branch glued to a quadratic branch) is introduced and used to construct an efficient algorithm. Convergence of the algorithms is proved for the case of π being a sublinear function of the dual multipliers. The algorithms are tested on large-scale quadratically constrained problems arising in structural optimization.

Key words. convex programming, augmented Lagrangian

AMS subject classification. 90C25

PII. S1052623493259215

1. Introduction.

Methods of multipliers for constrained convex programs, involving *nonquadratic augmented Lagrangians*, are getting renewed attention. New theoretical results are given for a *shifted logarithmic multiplier method* in [13] and [14]. A more general scheme is studied in [10]. Implementations reporting good numerical results are given in [4] and [8]. For the *exponential method of multipliers*, new convergence results, including rate of convergence (for the linear programming case), are obtained in [19].

Here we introduce a class of methods called *penalty/barrier multiplier* (PBM) *methods*, which are based on nonquadratic augmented Lagrangians. A member in the PBM class is specified by a *penalty/barrier function* φ and a *penalty updating function* π, responsible for updating the penalty parameters in each iteration. The requirements on φ are rather mild and allow for the inclusion as special cases, the exponential and the shifted logarithmic functions. More importantly, in section 4 we suggest a new type of penalty/barrier function made of a logarithmic branch glued smoothly to a quadratic branch. A PBM algorithm based on this log-quadratic augmented Lagrangian proved to be very efficient and capable of solving large-scale problems to a high degree of accuracy (see sections 7 and 8). The requirement on the penalty updating function π is that it is a sublinear function of the multipliers. This requirement was inspired by a suggestion in the paper by Tseng and Bertsekas [19]. We point out that an augmented Lagrangian resulting from such a choice of π is a *nonlinear function of the multipliers*. In section 4 we show that the PBM method is associated with a "proximal point" algorithm, which simultaneously solves the *dual* convex programming problem. The distance-like function appearing in the proximal term is related to the conjugate function of φ. Convergence properties of the proximal point algorithm are studied in section 5, and the results obtained

* Received by the editors November 30, 1993; accepted for publication (in revised form) January 8, 1996. The research of the first author was supported by a grant from the Israel-U.S.A. Binational Science Foundation (BSF). The research of the second author was supported by The Center for Absorption in Science, Ministry of Immigrant Absorption, State of Israel.

http://www.siam.org/journals/siopt/7-2/25921.html

[†] Faculty of Industrial Engineering and Management, Technion—Israel Institute of Technology, Haifa 32000, Israel (morbt@tx.technion.ac.il, mcib@ie.technion.ac.il).

there are instrumental in proving that the PBM algorithm produces a sequence of points which are asymptotically primal feasible. Full convergence analysis of the PBM method is contained in section 6. It is shown (see Theorem 1) that the primal–dual sequences generated by the algorithm are bounded, and each of their limit points is a pair of solutions for the primal and dual problems. In section 7 we discuss implementation issues of the PBM method and in section 8 we apply the algorithm to large-scale quadratically constrained convex problems which arise in two types of structural optimization—(1) truss topology design and (2) shape design. For the first application, the largest problem solved has 462 variables and 16290 quadratic constraints. For the second application, the largest problem solved has 6498 variables and 3136 quadratic constraints. The computational results[1] demonstrate that the PBM method solves such problems in almost a fixed number of Newton steps (typically 30) independent of the problems' dimensions.

2. Problem formulation and assumptions.

We study the *ordinary convex programming problem* (in the sense of Rockafellar [15, p. 277])

$$(P) \qquad f^* = \inf\{f(x): \ g_i(x) \le 0 \ , \qquad i = 1, \dots, m\},$$

where the functions $f,\ g_i \dots, g_m: \ \mathbb{R}^n \to R$ are closed proper convex functions. We let S denote the feasible set of P and S^* the set of optimal solutions. It is assumed throughout the paper that

$$(A1) \qquad S^* \text{ is nonempty and compact.}$$

Associated with problem (P) is its *Lagrangian*

$$L(x, u) := f(x) + \sum_{i=1}^{m} u_i g_i(x)\,,$$

its *dual function*

$$G(u) := \inf\{L(x, u): \ x \in \mathbb{R}^n\},$$

and the *dual concave problem*

$$(D) \qquad G^* = \sup\{G(u) : u \ge 0\}.$$

Our second working assumption is Slater's condition:

$$(A2) \qquad \text{there exists } \hat{x} \in \operatorname{dom} f \text{ such that } g_i(\hat{x}) < 0, \quad i = 1, \dots, m.$$

Following this assumption, it is well known that the optimal solution set of the dual problem (D) is nonempty and compact and that $f^* = G^*$. Moreover, for each $\beta' < G^*$, the level set

$$\{u \in \mathbb{R}^m : u \ge 0,\ G(u) \ge \beta'\}$$

[1] Additional computational results for the truss topology design problem were reported in an earlier version of this paper; see [4].

is compact.

Note that we do not assume differentiability of the functions $f, g_1, \dots, g_m$.

3. The penalty/barrier multiplier (PBM) method.

We transform the constraints of problem (P) using a real-valued function φ having the following properties:

$(\varphi 0)$ φ is a strictly increasing twice differentiable strictly convex function with

$$\mathrm{dom}\ \varphi = (-\infty, b),\ 0 < b \le \infty,$$

$(\varphi 1)$ $\varphi(0) = 0$,
$(\varphi 2)$ $\varphi'(0) = 1$,
$(\varphi 3)$ $\lim\limits_{t \to b} \varphi'(t) = \infty$,
$(\varphi 4)$ $\lim\limits_{t \to -\infty} \varphi'(t) = 0$,
$(\varphi 5)$ $\varphi''(t) \ge \frac{1}{M}$ for all $t \in [0, b]$ for some $M > 0$.

We next show that properties $(\varphi 1)$–$(\varphi 4)$ imply an important property of the *recession function* φ_∞ of φ. Recall [15, section 8] that

$$\varphi_\infty(s) := \lim_{\lambda \to \infty} \frac{\varphi(t + \lambda s) - \varphi(t)}{\lambda}\ \forall\ t \in\ \mathrm{dom}\ \varphi\ .$$

We further bring forth a result of Auslender, Cominetti, and Haddou [1] on the recession functions of a composite function.

LEMMA 1. *If φ possesses properties $\varphi(0)$–$\varphi(4)$ then*
$(\varphi 6)_a$ $\varphi_\infty(-1) = 0$,
$(\varphi 6)_b$ $\varphi_\infty(1) = \infty$.
Moreover, let h be a closed convex function with $\mathrm{dom}\ \varphi \cap h(\mathbb{R}^n) \ne \emptyset$ *and consider the composite function*

$$g(x) = \varphi(h(x))\ , \quad (x \in \mathrm{dom}\ f).$$

Then g is a closed convex function and its recession function is given by

$$g_\infty(d) = \begin{cases} \varphi_\infty(h_\infty(d)) & \textit{if } d \in \mathrm{dom}\ h_\infty, \\ +\infty & \textit{otherwise}\ . \end{cases} \tag{3.1}$$

Proof. For all $t \in \mathrm{dom}\ \varphi$,

$$\begin{aligned} \varphi_\infty(s) &= \lim_{\lambda \to \infty} \frac{\varphi(t + \lambda s) - \varphi'(t)}{\lambda} \\ &\ge \lim_{\lambda \to \infty} \frac{\lambda s \varphi'(t)}{\lambda} \end{aligned} \tag{3.2}$$

by the gradient inequality (valid since φ is convex); i.e.,

$$\varphi_\infty(s) \ge s\varphi'(t) \quad \forall\ t \in\ \mathrm{dom}\ \varphi = (-\infty, b). \tag{3.3}$$

Letting $t \to b$, it follows from (3.3) that

$$\varphi_\infty(1) \ge \lim_{t \to b} \varphi'(t) = \infty \ \text{ by } (\varphi 3).$$

Letting $t \to -\infty$, it follows from (3.2) that

$$\varphi_\infty(-1) \ge -\lim_{t \to -\infty} \varphi'(t) = 0 \ \text{ by } (\varphi 4).$$

Also, by (3.2) and since φ is increasing,

$$\varphi_\infty(-1) \le 0,$$

hence

$$\varphi_\infty(-1) = 0\ .$$

The formula for the recession function g_∞ of the composite function $g(x) = \varphi(h(x))$ now follows from Proposition 2.1 in [1]. □

Let p be a positive number. Then the function $t \to \varphi(t/p)$ is convex increasing and, in particular,

$$p\varphi(t/p) \le 0 \text{ iff } t \le 0\ .$$

Consequently, the constraints in problem (P) can be equivalently written as

$$p_i\varphi(g_i(x)/p_i) \le 0\ , \quad i = 1, \dots, m, \tag{3.4}$$

where $p_i > 0$ is a *penalty parameter* for the ith constraint. The Lagrangian corresponding to minimizing f subject to (3.4) is

$$F(x, u, p) := f(x) + \sum_{i=1}^{m} u_i p_i \varphi(g_i(x)/p_i)\ . \tag{3.5}$$

We say that F is the *augmented Lagrangian* for problem (P).

The family of PBM methods for solving (P) is iterative. At the $(k+1)$-iteration, the augmented Lagrangian F is minimized with respect to x:

$$x^{k+1} = \arg\min_x F(x, u^k, p^k), \tag{3.6}$$

and then the multipliers u_i^k, p_i^k $(i = 1, \dots, m)$ are updated:

$$u_i^{k+1} = u_i^k \varphi'(g_i(x^{k+1})/p_i^k), \tag{3.7}$$

$$p_i^{k+1} = \pi^k(u_i^{k+1}). \tag{3.8}$$

Initially, a positive multiplier vector is chosen; $u^0 > 0$. Here π^k is a *penalty updating function* $\pi^k : R_{++} \to R_{++}$ (here and throughout the paper R_{++} denotes the positive real line). A specific algorithm in the PBM family is determined by the particular choice of the functions φ and π^k.

The multiplier's updating formula (3.7) is motivated by the optimality condition on x^{k+1}:

$$0 \in c^{k+1} + \sum u_i^k \varphi'(g_i(x^{k+1})/p_i^k) c_i^{k+1},$$

where $c^k \in \partial f(x^{k+1})$, $c_i^k \in \partial g_i(x^{k+1})$. Thus, for u^{k+1} being chosen as in (3.7), x^{k+1} satisfies

$$0 \in \partial_x L(x^{k+1}, u^{k+1}),$$

hence $x^{k+1} \in \arg\min_x L(x, u^{k+1})$. Moreover, a lower bound for the optimal value of (P) is given by the dual objective function

$$\inf(P) \ge G(u^{k+1}) := \min_x L(x, u^{k+1}) = L(x^{k+1}, u^{k+1}). \tag{3.9}$$

The updating formula (3.7) can be explained intuitively as follows: if x^{k+1} is not feasible for the ith constraint, $g_i(x^{k+1}) > 0$, then the influence of this constraint grows since its multiplier u_i^k is increased (recall that by property (φ2), $\varphi'(g_i(x^{k+1})/p_i^k) > 1$ for positive g_i).

The updating formula (3.8) for the penalty parameter p_i^k generalizes the idea of Tseng and Bertsekas [19] for the *exponential method of multipliers*, where the following two choices of the function $\pi^k(\cdot)$ are discussed: $\pi^k(t) = c^k$ and $\pi^k(t) = c^k t \ \ (c^k > 0)$. Global convergence is proved in [19] only for the first choice. In this paper, we prove convergence for a class of PBM methods with π^k nondecreasing and *sublinear*, i.e.,

$$\forall\ t > 0 : \quad \pi^k(t) \le ct \quad \text{ for some } c > 0\ .$$

It might be noted that the exponential method of multipliers belongs to a class of multiplier methods described in Bertsekas' 1982 book [7], which was introduced as early as 1973 in two papers [11, 12] by Kort and Bertsekas. In these publications, the properties of the function φ are slightly different and π^k is a chosen constant. Algorithms of the type (3.6)–(3.8) with $\pi^k =$ constant and with specific choices of φ were also studied by Polyak [14] and Iusem et al. [10].

We give now a few examples of PBM-type methods.

1. *The classical augmented Lagrangian* [16, 17]. This PBM-type method is obtained by choosing

$$\varphi(t) = \begin{cases} t + \frac{1}{2}t^2 & \text{if} \quad t \ge -1, \\ -\frac{1}{2} & \text{if} \quad t < -1. \end{cases}$$

This function does not in fact satisfy all our basic assumptions. Indeed (φ0) is violated since φ here is neither twice differentiable, *strictly* increasing, nor *strictly* convex.

In all other examples below, φ satisfies all the properties (φ0)–(φ5).

2. *The exponential method of multipliers* [7, 19]. Here

$$\varphi(t) = e^t - 1\ .$$

3. *The modified barrier method* [14]. Here φ is a shifted logarithmic function:

$$\varphi(t) = -\log(1-t), \qquad -\infty < t < 1\ .$$

For examples 2 and 3, φ is a c^2-function, but the second derivative is not bounded throughout the domain of φ; this is a source of difficulty for applying the Newton method to the unconstrained minimization. The next two examples are new and give rise to our preferred (and implemented) multiplier method.

4. *A quadratic-logarithmic penalty function.*

$$\varphi(t) = \begin{cases} t + \frac{1}{2}\,t^2 & \text{if } t \ge -\frac{1}{2}, \\ -\frac{1}{4}\log(-2t) - \frac{3}{8} & \text{if } t < -\frac{1}{2}. \end{cases}$$

5. *A quadratic-reciprocal penalty function.*

$$\varphi(t) = \begin{cases} t + \frac{1}{2}\,t^2 & \text{if } t \ge -\frac{1}{3}, \\ \frac{32}{27}\left(\frac{1}{1-t}\right) - \frac{7}{6} & \text{if } t < -\frac{1}{3}\ . \end{cases}$$

The functions in examples 4 and 5 are twice continuously differentiable, and the second derivative $\varphi''(t)$ is bounded above for all $t \in R$. Both are made from a "barrier branch" (logarithmic or reciprocal) and a "penalty branch" (quadratic).

Remark. If $\pi^k(t) = c_k t$, then the term $u_i^k p_i^k \varphi(g_i(x)/p_i^k)$ in the augmented Lagrangian F is, for the quadratic branch in both examples 4 and 5,

$$(c_k u_i^k) u_i^k \left[\frac{g_i(x)}{c_k u_i^k} + \frac{1}{2} \left(\frac{g_i(x)}{c_k u_i^k} \right)^2 \right] = u_i^k g_i(x) + \frac{1}{2c_k} (g_i(x))^2,$$

which is precisely the corresponding term of Rockafellar's quadratic augmented Lagrangian.

For a PBM method to be well defined, the unconstrained minimization of F in step (3.6) must have a solution. In the next proposition we demonstrate that under our assumptions, this is indeed the case. The proof is essentially similar to a result in [1].

PROPOSITION 1. *Assume that Assumption* A1 *holds and that φ satisfies properties* (φ0)–(φ4). *Then for every* $p > 0,\ u > 0$, *the solution set of the unconstrained problem*

$$\min_x F(x, u, p)$$

is nonempty and compact.

Proof. Let $p > 0,\ u > 0$ be fixed and denote

$$F(x) := F(x, u, p) = f(x) + \sum_{i=1}^m u_i p_i \varphi(g_i(x)/p_i) \ .$$

We derive next a formula for the recession function F_∞ of F. By formula (3.1) in Lemma 1, the recession function of

$$p_i \varphi(g_i(x)/p_i)$$

is

$$\begin{cases} p_i \varphi_\infty((g_i)_\infty(d)/p_i) & \text{if } d \in \text{ dom } (g_i)_\infty, \\ \infty & \text{otherwise,} \end{cases}$$

which further reduces to

$$\begin{cases} \varphi_\infty((g_i)_\infty(d)) & \text{if } d \in \text{ dom } (g_i)_\infty, \\ \infty & \text{otherwise,} \end{cases}$$

since a recession function is positively homogeneous. Therefore,

$$F_\infty(d) = \begin{cases} f_\infty(d) + \sum_{i=1}^m u_i \varphi_\infty((g_i)_\infty(d)) & \text{if } d \in \cap \text{ dom } (g_i)_\infty, \\ \infty & \text{otherwise.} \end{cases}$$

Using the positive homogeneity of φ_∞ for $d \in \cap$ dom $(g_i)_\infty$, we further get

$$F_\infty(d) = f_\infty(d) + \sum_{\{i:(g_i)_\infty d > 0\}} u_i (g_i)_\infty(d) \varphi_\infty(1)$$
$$+ \sum_{\{i:(g_i)_\infty d \le 0\}} u_i |(g_i)_\infty(d)| \varphi_\infty(-1) \ .$$

Using properties $(\varphi 6)_a$ and $(\varphi 6)_b$ in Lemma 1, we finally get

$$F_\infty(d) = \begin{cases} f_\infty(d) & \text{if } (g_i)_\infty(d) \le 0 \;\; \forall\, i = 1, \dots, m, \\ \infty & \text{otherwise.} \end{cases} \tag{3.10}$$

It is well known that assumption (A1) on the compactness and nonemptiness of S^* means the following:

$$\nexists d \neq 0 \text{ such that } f_\infty(d) \le 0,\; (g_i)_\infty(d) \le 0 \;\forall\, i = 1, \dots, m.$$

Hence by (3.10),

$$\nexists d \neq 0 \text{ such that } F_\infty(d) \le 0,$$

which implies that the solution set of $\min_x F(x)$ is nonempty and compact. □

4. Dual interpretation of the PBM method. We show in this section that the PBM algorithm (3.6)–(3.8) generates the same sequence $\{u^k\}$ as an appropriate (nonquadratic) "proximal point" algorithm applied to the maximization of the dual objective function G. Such a dual interpretation is well known for the quadratic augmented Lagrangian method (see [16]) and for the entropic augmented Lagrangians introduced recently by Teboulle [18].

For the dual problem

$$\max_{u \in R^m_+} G(u) \;, \tag{4.1}$$

the prox-algorithm is given in terms of a distance-like function $D_k : R^m_+ \times R^m_+ \to R_+$ by the iterative formula

$$u^{k+1} = \arg\max_u \{G(u) - D_k(u, u^k)\} \;. \tag{4.2}$$

We consider here *separable* functions D_k,

$$D_k(u, u^k) = \sum_{i=1}^m \rho_i^k(u_i, u_i^k), \quad \rho_i^k : \; R_+ \times R_+ \to R_+. \tag{4.3}$$

Next, we show how the choice of the function φ, in the PBM method, dictates the specific form of the function ρ_i^k in (4.3) and, hence, the specific form of the distance function D_k. In the sequel, we occasionally omit the indices i and k in ρ_i^k and use simply ρ.

From the fact that $x^{k+1} = \arg\min L(x, u^{k+1})$, it follows that

$$G(u^{k+1}) = L(x^{k+1}, u^{k+1}) = f(x^{k+1}) + \sum u_i^{k+1} g_i(x^{k+1}) \;.$$

Now,

$$\begin{aligned} G(u) = \min_x \left\{ f(x) + \sum u_i g_i(x) \right\} &\le f(x^{k+1}) + \sum u_i g_i(x^{k+1}) \\ &= f(x^{k+1}) + \sum u_i^{k+1} g_i(x^{k+1}) + \sum (u_i - u_i^{k+1}) g_i(x^{k+1}) \\ &= G(u^{k+1}) + \sum (u_i - u_i^{k+1}) g_i(x^{k+1}), \end{aligned}$$

showing that

$$g(x^{k+1}) \in \partial G(u^{k+1}) \; ; \tag{4.4}$$

here $g(\cdot) = (g_i(\cdot), \ldots, g_m(\cdot))^T$, and $\partial G(u)$ is the *subgradient set* of the concave function G at u. The updating formula (3.7) can be rewritten as follows:

$$g_i(x^{k+1}) = p_i^k {\varphi'}^{-1}(u_i^{k+1}/u_i^k). \tag{4.5}$$

By choosing ρ such that its derivate with respect to the first argument $\rho_1'(\cdot,\cdot)$ is given by

$$\rho_1'(u_i^{k+1}, u_i^k) = g_i(x^{k+1}), \tag{4.6}$$

we will have by (4.4) that

$$\begin{pmatrix} \rho_1'(u_1^{k+1}, u_1^k) \\ \vdots \\ \rho_1'(u_m^{k+1}, u_m^k) \end{pmatrix} \in \partial G(u^{k+1}),$$

which is precisely the necessary and sufficient condition for u^{k+1} to satisfy (4.2). Now (4.5) and (4.6) give the following relation between φ and ρ:

$$\rho_1'(u_i^{k+1}, u_i^k) = p_i^k (\varphi')^{-1}(u_i^{k+1}/u_i^k). \tag{4.7}$$

Using the relation

$$(\varphi')^{-1} = (\varphi^*)',$$

when φ^* is the conjugate function of φ (see, e.g., [15]), then by integrating (4.7) and denoting $\psi = \varphi^*$ we get

$$\rho(u_i^{k+1}, u_i^k) = p_i^k u_i^k \psi(u_i^{k+1}/u_i^k) \; , \tag{4.8}$$

which is the promised relation between φ and ρ.

Recalling the relation (3.8), the final generic expression for ρ is

$$\rho(\alpha, \beta) = \beta\pi(\beta)\psi(\alpha/\beta), \qquad \alpha \geq 0, \; \beta > 0, \; \pi(\beta) \geq 0. \tag{4.9}$$

Property ($\varphi 0$) implies that $\psi = \varphi^*$ is an essentially smooth (hence differentiable) function on R_{++} (see [15, section 26]). Also,

$$\psi' = (\varphi')^{-1}.$$

These facts, together with properties ($\varphi 0$)–($\varphi 5$) imply that ψ inherits from φ the following properties:

($\psi 0$) ψ is a strictly convex differentiable function in $(0, \infty)$,

($\psi 1$) $\psi(1) = 0$,

($\psi 2$) $\psi'(1) = 0$,

($\psi 3$) [barrier property] $\lim_{t \to 0^+} \psi'(t) = -\infty$; $\psi(t) = \infty$ for $t \leq 0$ and $\lim_{t \to \infty} \psi(t) = \infty$,

($\psi 4$) $\psi''(t) \leq M$ for $t \geq 1$.

The properties of φ, ψ and the derivatives φ', ψ' are illustrated in Figure 1. As a

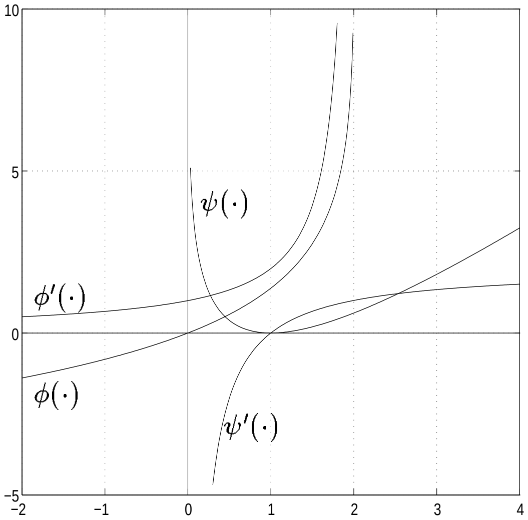


FIG. 1. *The functions φ, $\psi = \varphi^*$ and their derivatives.*

specific example, we take the logarithmic-quadratic function φ in example 4:

$$\varphi(t) = \begin{cases} t + \frac{1}{2}t^2 & \text{if } \ t \geq -\frac{1}{2}, \\ -\frac{1}{4}\log(-2t) - \frac{3}{8} & \text{if } \ t < -\frac{1}{2}. \end{cases}$$

Computing its conjugate function, we get

$$\psi(s) = \varphi^*(s) = \begin{cases} \frac{1}{2}(s-1)^2 & \text{if } s \geq \frac{1}{2}, \\ \frac{1}{8} - \frac{1}{4}\log(2s) & \text{if } 0 < s < \frac{1}{2} \ . \end{cases}$$

All the properties ($\psi 0$)–($\psi 4$) can be easily verified for this example.

From ($\psi 0$) and the fact that $\pi(\beta) > 0$ for $\beta > 0$, it follows that ($\rho 0$) $\rho(\cdot, \beta)$ is a strictly convex and differentiable function in $(0, \infty)$ for $\beta > 0$.

Since $0 = \psi(1) = \min_{t>0} \psi(t)$, it also follows that ρ is a distance-like function. For example,

($\rho 1$) $\rho(\alpha, \beta) \geq 0, \quad \rho(\alpha, \alpha) = 0, \quad \alpha > 0, \ \beta > 0.$

Also, by ($\psi 2$),

($\rho 2$) $\rho'_1(\alpha, \alpha) = 0.$

From the barrier property ($\psi 3$), a corresponding property follows for ρ.

($\rho 3$) $\forall \beta > 0 : \rho(\alpha, \beta) = \infty$ if $\alpha \leq 0 : \lim_{\alpha \to \infty} \rho(\alpha, \beta) = \infty.$

To sum up, we have shown that the sequence of multipliers $\{u^k\}$ generated by a PBM method (3.6)–(3.8), with certain penalty function φ, is the same as the sequence u^k generated by the prox-algorithm (4.2) applied to the dual problem (D), with certain distance-like functions $D_k = \sum \rho_i^k$ where each of the functions ρ_i^k is of the form (4.9) and with the function ψ in (4.9) being the conjugate of φ.

We give now a general basic result concerning convergence of the iterative prox-algorithm for solving concave maximization problems:

$$H^* := \max\{H(u) : \ u \in \mathbb{R}^m_+\} \ . \tag{4.10}$$

Proposition 2. *Let H be a concave function which is bounded above on $\mathbb{R}^m_+$. Consider a distance-like function $D : \mathbb{R}^m_+ \times \mathbb{R}^m_+ \to R$ and assume it has the following properties:*

- (D0) *$D(\cdot, v)$ is a strictly convex function on $\mathbb{R}^m_+$ with respect to its first argument, $\forall\ v > 0$.*
- (D1) *$D(\cdot, v)$ has bounded level sets $\forall\ v > 0$.*
- (D2) *$D(u, v) \geq 0,\ D(u, u) = 0$ ("distance" property).*
- (D3) *$\forall v > 0,\ D(u, v) = \infty$ if $u \not> 0$, and $\lim_{\|u\|\to\infty} D(u, v) = \infty$ (barrier property).*

For the next property, let d be a vector in the subgradient set of D taken with respect to the first argument

$$d \in \partial_1 D(u, v).$$

- (D4) *For every $\epsilon > 0$, there exists $\delta > 0$ such that if for some $i \in \{1, \dots, m\}$ $d_i > \epsilon$, $D(u, v) > \delta$.*

Then

(a) *the sequence u^k generated by the iterative process*

$$u^{k+1} = \arg\max\{H(u) - D(u, u^k)\} \ , \quad u^0 > 0, \tag{4.11}$$

is well defined, positive, and the sequence of function values $H(u^k)$ is nondecreasing;

(b) *there exists a sequence of vectors d^k such that*

$$d^k \in \partial_1 D(u^k, u^{k-1}), \tag{4.12}$$

$$d^k \in \partial H(u^k), \tag{4.13}$$

and for every such sequence,

$$\lim_{k\to\infty} (d^k_i)_+ = 0 \qquad i = 1, \dots, m. \tag{4.14}$$

Here, and henceforth, α_+ denotes the positive part of a number $\alpha \in R$:

$$\alpha_+ = \max(0, \alpha). \tag{4.15}$$

Proof. (a): Since H is bounded above and, by (D1), $D(\cdot, u^k)$ has bounded level sets, the max in (4.11) is attained. Moreover, if $u^k > 0$, then by the barrier property (D3), $u^{k+1} > 0$, also by (D0), u^{k+1} is uniquely determined. Now

$$H(u^{k+1}) - D(u^{k+1}, u^k) \geq H(u^k) - D(u^k, u^k) = H(u^k) \ \text{ by } \ \text{(D2)}.$$

Hence, using (D2) again,

$$H(u^{k+1}) - H(u^k) \geq D(u^{k+1}, u^k) \geq 0. \tag{4.16}$$

(b): The existence of d^k satisfying (4.12)–(4.13) follows from the necessary and sufficient optimality condition for u^k:

$$0 \in \partial H(u^k) - \partial_1 D(u^k, u^{k-1}).$$

Arguing by contradiction, suppose that (4.14) is *not* satisfied. Then there is an infinite set of indices $\{k_j\}$ such that for some $i \in \{1, \ldots, m\}$,

$$\frac{\partial_1 D(u^{k_j}, u^{k_j-1})}{\partial u_i^{k_j}} > \epsilon.$$

Then by property (D4) and (4.16), for some $\delta > 0$,

$$H(u^{k_j}) - H(u^{k_j-1}) \geq \delta > 0.$$

Thus, the infinite sequence $H(u^{k_j})$ increases each step at least by a positive constant; hence, it cannot be bounded above, contrary to our assumption. □

Remark. With obvious changes, Proposition 2 remains valid if the distance function is $D_k(u,v)$; i.e., it depends on the iteration number k.

5. Properties of the distance function $D(u,v)$. Consider the expression for the distance function obtained in section 3 (4.3):

$$D(u,v) = \sum_{i=1}^{m} \rho(u_i, v_i)\,, \quad u \geq 0,\ v > 0, \tag{5.1}$$

with

$$\rho(\alpha,\beta) = \beta\pi(\beta)\psi(\alpha/\beta), \quad \alpha \geq 0,\ \beta > 0, \tag{5.2}$$

$$\pi(\beta) > 0 \text{ for } \beta > 0,\ \pi \text{ increasing in } R_+, \tag{5.3}$$

and where

$$\psi = \varphi^* \text{ and } \varphi \text{ satisfies } (\varphi 0)\text{–}(\varphi 5). \tag{5.4}$$

From the properties (ψ0)–(ψ3), which follow from (5.4), and the corresponding properties (ρ0)–(ρ3) of ρ, we get the following proposition.

Proposition 3. *The distance function D described in* (5.1)–(5.4) *satisfies conditions* (D0)–(D3) *in Proposition* 2. □

It is a much more difficult task to show that D also satisfies the crucial property (D4) in Proposition 3. To this end we assume that the penalty updating function π is *sublinear*; i.e.,

$$\forall\ t > 0 : \pi(t) \text{ is increasing positive and } \pi(t) \leq ct \text{ for some } c > 0\ . \tag{5.5}$$

We first prove an important property of ρ.

Lemma 2. *If*

(i) $\alpha > \beta > 0$,

(ii) $0 < \pi(\beta)/\beta < c$,

then ρ given in (5.2) *satisfies the inequality*

$$\rho(\alpha,\beta) \geq \frac{1}{2}\ \frac{[\rho_1'(\alpha,\beta)]^2}{cM}, \tag{5.6}$$

where

$$M = \max_{t\geq 1} \psi\text{"}(t) < \infty\ .$$

Proof. Consider the quadratic function

$$q(t) = q(\alpha) + (t-\alpha)\rho_1'(\alpha,\beta) + \frac{1}{2}(t-\alpha)^2 cM.$$

Its minimizer is

$$t^* = -\frac{\rho_1'(\alpha,\beta)}{cM} \ + \ \alpha, \tag{5.7}$$

and

$$q(t^*) = q(\alpha) - \frac{1}{2}\ \frac{[\rho_1'(\alpha,\beta)]^2}{cM}\ . \tag{5.8}$$

Clearly (i) implies $\rho_1'(\alpha,\beta) > 0$, and so by (5.7),

$$t^* < \alpha\ . \tag{5.9}$$

We next show

$$t^* \geq \beta\ . \tag{5.10}$$

Indeed,

$$\rho_1'(\alpha,\beta) = \rho_1'(\alpha,\beta) - \rho_1'(\beta,\beta) = (\alpha-\beta)\rho_1''(\mu,\beta)$$

for some $\beta \leq \mu \leq \alpha$; by the mean-value theorem,

$$= (\alpha-\beta)\frac{\pi(\beta)}{\beta}\psi''\left(\frac{\mu}{\beta}\right) \leq (\alpha-\beta)cM \ \text{ by } (\psi 4) \text{ and (ii).}$$

The latter inequality is rewritten as

$$\beta \leq -\frac{\rho'(\alpha,\beta)}{cM} \ + \ \alpha = t^*,$$

so (5.10) holds.

Again by the mean-value theorem, for every $\beta \leq t < \alpha$,

$$\rho_1'(\alpha,\beta) - \rho_1'(t,\beta) = (\alpha-t)\rho_1''(\bar{t},\beta) \text{ for some } t \leq \bar{t} \leq \alpha.$$

Since

$$\rho_1''(t,\beta) = \frac{\pi(\beta)}{\beta}\psi''\left(\frac{t}{\beta}\right) \leq cM, \quad \forall\ \beta \leq t, \tag{5.11}$$

it follows that

$$\rho_1'(\alpha,\beta) \leq \rho_1'(t,\beta) + (\alpha-t)cM \quad \forall\ \beta \leq t < \alpha.$$

For example,

$$\rho_1'(t,\beta) \geq \rho_1'(\alpha,\beta) + (t-\alpha)cM \quad \forall\ \beta \leq t \leq \alpha.$$

On the other hand,

$$q'(t) = \rho_1'(\alpha,\beta) + (t-\alpha)cM;$$

hence,

$$\rho_1'(t,\beta) \geq q'(t) \qquad \forall\ \beta \leq t \leq \alpha. \tag{5.12}$$

Since $\beta \leq t^* < \alpha$ by (5.9) and (5.10), we get, by integrating (5.12) from t^* to α,

$$\rho(\alpha,\beta) - \rho(t^*,\beta) \geq q(\alpha) - q(t^*).$$

Since $\rho(t^*,\beta) \geq 0$, the latter inequality implies (using (5.8))

$$\rho(\alpha,\beta) \geq q(\alpha) - q(t^*) = \frac{1}{2}\ \frac{[\rho_1'(\alpha,\beta)]^2}{cM}. \qquad \square$$

We now establish the fact that D satisfies property (D4) previously mentioned in Proposition 3.

PROPOSITION 4. *The distance function D described in* (5.1)–(5.5) *satisfies condition* (D4) *in Proposition* 2.

Proof. Suppose $\beta > 0$. Then

$$\epsilon > 0 \text{ and } \rho_1'(\alpha,\beta) \geq \epsilon. \tag{5.13}$$

Since $\rho_1'(\alpha,\beta) > 0$ by (ρ2) and the strict convexity of $\rho(\cdot,\beta)$, $\alpha > \beta$, and since we are in the situation covered by Lemma 2, we thus conclude that

$$\rho(\alpha,\beta) \geq \delta := \frac{\epsilon^2}{2cM}. \tag{5.14}$$

Now, by (D0), which was established in Proposition 2, the subgradient $\partial_1 D(u,v)$ is the single vector whose ith component is $d_i = \rho_1'(u_i,v_i)$. Hence, the fact we just proved, that (5.13) implies (5.14), proves that (D4) holds. $\square$

6. Convergence of the PBM method.

The main theoretical result of our paper follows.

THEOREM 1. *Let the convex program* (P) *satisfy assumptions* A1 *and* A2. *Let φ satisfy properties (φ0–φ5) and let π^k satisfy the sublinearity condition* (5.5) *for each k. Then the sequences $\{x^k\}\{u^k\}$ generated by the PBM method* (3.6)–(3.8) *satisfy the following as $k \to \infty$:*

$$g_i(x^k)_+ \to 0, \qquad i = 1,\ldots,m, \tag{6.1}$$

$$f(x^k) \to f^*, \tag{6.2}$$

and

$$u_i^k g_i^k(x^k) \to 0, \qquad i = 1,\ldots,m\,. \tag{6.3}$$

Moreover, $\{x^k\}$ and $\{u^k\}$ are bounded sequences and each of their limit points is a pair of optimal solutions to (P) *and* (D), *respectively.*

Proof. We first prove the asymptotic primal feasibility result (6.1) by using Proposition 2, with $H(u) = G(u)$. Recall that by (4.4),

$$g(x^k) \in \partial G(u^k). \tag{6.4}$$

Also,

$$\begin{aligned} g_i(x^k) &= p_i^{k-1}\varphi'(u_i^k/u_i^{k-1}) && \text{by (4.5),} \\ &= \rho_1'(u_i^k, u_i^{k-1}) && \text{by (4.7),} \\ &= \tfrac{\partial_1 D(u^k,u^{k-1})}{\partial u_i^k} && \text{by (4.3).} \end{aligned}$$

Therefore, the vector $d^k = g(x^k)$ satisfies (4.12) and (4.13) of Proposition 2. Furthermore, the distance function D has properties (D0)–(D4) (this was proved in Propositions 3 and 4) and so, from conclusion (b) of Proposition 2,

$$\lim_{k\to\infty} g_i(x^k)_+ = 0,$$

proving (6.1).

We next prove the complementarity relation (6.3). We use the notation $g_i^k = g_i(x^k)$. Recall that by the Slater condition (assumption A2), the level sets of the dual objective function G are compact. Also, from conclusion (a) of Proposition 2, $G(u^k)$ is a nondecreasing sequence, so for all k, u^k belong to the compact set

$$\{u|G(u^k) \geq G(u^0)\},$$

and hence $\{u^k\}$ is a bounded sequence, say $u_i^k \leq \bar{u}$, for all i. If $g_i^k \to 0$, $u_i^k g_i^k \to 0$ also and (6.3) holds. If $g_i(x^k)$ remains bounded away from zero, then by (6.1) for some $\rho < 0$,

$$g_i^k \leq \rho < 0\ . \tag{6.5}$$

From the updating formula (3.7),

$$\begin{aligned} g_i^k(u_i^{k-1} - u_i^k) &= g_i^k u_i^{k-1}[1 - \varphi'(g_i^k/p_i^{k-1})] \\ &= g_i^k u_i^{k-1} - u_i^{k-1}p_i^{k-1}(g_i^k/p_i^{k-1})\varphi'(g_i^k/p_i^{k-1}) \\ &\leq g_i^k u_i^{k-1} - u_i^{k-1}p_i^{k-1}(g_i^k/p_i^{k-1}) = 0, \\ &\quad \text{since } t\varphi'(t) \geq t, \end{aligned} \tag{6.6}$$

hence

$$g_i^k(u_i^{k-1} - u_i^k) \leq 0. \tag{6.7}$$

From (6.4) and (6.7),

$$0 \geq \sum g_i^k(u_i^{k-1} - u_i^k) \geq G(u^{k-1}) - G(u^k), \tag{6.8}$$

but $G(u^k)$ is increasing and bounded by G^*, so $G(u^{k-1}) - G(u^k) \to 0$. From (6.8), then

$$\sum g_i^k(u_i^{k-1} - u_i^k) \to 0. \tag{6.9}$$

Moreover, by (6.7) each term in the summation is nonpositive. Thus (6.9) implies

$$g_i^k(u_i^{k-1} - u_i^k) \to 0 \qquad \forall\ i = 1, \ldots, m, \tag{6.10}$$

and by (6.6)

$$g_i^k u_i^{k-1}[1 - \varphi'(g_i^k/p_i^{k-1})] \to 0, \quad \forall\, i = 1, \ldots, m. \tag{6.11}$$

The sequence p_i^k is bounded, since u_i^k is bounded and $p_i^k \le cu_i^k$ by the sublinearity of π^k, so in particular

$$p_i^{k-1} \le c\bar{u}.$$

And so, by (6.5),

$$g_i^k/p_i^{k-1} \le \frac{\rho}{c\bar{u}} < 0. \tag{6.12}$$

The function φ' is strictly increasing and $\varphi'(0) = 1$; hence, by (6.12),

$$0 < \varphi'(g_i^k/p_i^{k-1}) < 1.$$

And so, by (6.11),

$$g_i^k u_i^{k-1} \to 0,$$

and thus, by (6.10),

$$g_i^k u_i^k \to 0,$$

proving (6.3).

Next, we prove (6.2). From (6.1) x^k is asymptotically feasible, so for all $\epsilon > 0$, $f(x^k) \ge f^* - \epsilon$ for k large enough.

From (3.9),

$$f^* \ge G(u^k) = f(x^k) + \sum_{i=1}^m u_i^k g_i(x^k). \tag{6.13}$$

Combining the last two inequalities,

$$\forall\, \epsilon > 0, \quad f^* - \epsilon \le f(x^k) \le f^* - \sum u_i^k g_i^k \text{ for } k \text{ large enough;}$$

using (6.3), we get $f(x^k) \to f^*$.

Now, by (6.1) and (6.3), there exist $\epsilon > 0$ such that for k sufficiently large,

$$g_i(x^k) \le \epsilon\,, \quad f(x^k) \le f^* + \epsilon\ . \tag{6.14}$$

Due to assumption A1 (compactness of the primal optimal set), for any α, β the set

$$\{x \in \mathbb{R}^n : g_i(x) \le \alpha,\ f(x) \le \beta\}$$

is compact [9, Cor. 20]. Hence, by (6.14) the sequence $\{x^k\}$ is bounded. We already mentioned that $\{u^k\}$ is bounded, so let $(\bar{x}, \bar{u})$ be a limit point of $\{x^k\}, \{u^k\}$. It follows from (6.1) and (6.2) that $\bar{x}$ is a primal optimal solution. By this and by using (6.13) and (6.3),

$$G(\bar{u}) = f^*.$$

But $f^* = G^*$ by strong duality (which holds due to assumption (A2). Hence, $\bar{u}$ is a dual optimal solution. □

7. Implementation. The overall efficiency of a PBM method depends mainly on the efficiency of solving the unconstrained minimization

$$x^{k+1} = \arg\min_x F(x, u^k, p^k). \tag{7.1}$$

In our implementation, (7.1) is solved by a Newton method with linesearch. It stops as soon as either the decrease of F per Newton step is less than $\alpha \cdot \min_i\{p_i^k\}$ or $\|\nabla_x F(x, u^k, p^k)\| < \alpha$ (typically $\alpha = 0.1$). The starting point for the Newton method to solve (7.1) is the last iterate x^k. Clearly, this is a reasonable starting point, provided $F(\cdot, u^k, p^k)$ is not too different from $F(\cdot, u^{k-1}, p^{k-1})$, which may occur if for some $i \in \{1, \ldots, m\}$ the ratio u_i^{k+1}/u_i^k is too large or too small. To prevent this, we impose the safeguard rule

$$\mu \le u_i^k/u_i^{k-1} \le 1/\mu, \tag{7.2}$$

where $0 < \mu < 1$ is a user-prescribed parameter (we found $\mu = 0.3$ to give consistently good results). Thus, the modified multipliers' updating rule is

$$u_i^{k+1} = \begin{cases} \mu u_i^k & \text{if } \bar{u}_i^{k+1}/u_i^k < \mu, \\ \bar{u}_i^{k+1} := u_i^k \varphi'(g_i(x^{k+1}/p_i^k)) & \text{if } \mu \le \bar{u}_i^{k+1}/u_i^k \le 1/\mu, \\ (1/\mu)u_i^k & \text{if } \bar{u}_i^{k+1}/u_i^k > 1/\mu. \end{cases}$$

The safeguard (7.2) also restricts the influence of inaccuracy in the minimization (7.1) on the values of the new multipliers and, moreover, prevents them from approaching zero too early. In our numerical experiments, after very few iterations (rarely more than three), the upper bound in (7.2) was not activated. Towards convergence, only nonbinding constraints ($u_i^* = 0$) were activating the lower bound.

Two choices of the penalty-updating functions π were implemented:

(i) $\pi^k(t) = \pi_0(\mu)^k t$.

(ii) $\pi^k(t) = \pi_0(\mu)^k$.

The parameter $0 < \mu < 1$ is the same used for (7.2), and $\pi_0 > 0$ is a parameter with typical values between 10–1000. Also, the initial choice of the multiplier vector is $u_i^0 = 0.01$ for all i. The first choice (i) of π^k agrees with the sublinear assumption

$$\frac{\pi^k(t)}{t} \le c \qquad \forall\, t > 0 \tag{7.3}$$

with $c = \pi_0\mu$. Recall that condition (7.3) was crucial for the convergence analysis, but, in fact, it is needed only for values $t = u_i^k$; i.e.,

$$\frac{\pi^k(u_i^k)}{u_i^k} \le c\ . \tag{7.4}$$

It is easy to see that the choice (ii) of π^k, together with the safeguard rule (7.2), implies

$$\frac{\pi^k(u_i^k)}{u_i^k} = \frac{\pi_0\mu^k}{u_i^k} \le \frac{\pi_0\mu^k}{u_i^0\mu^k} \le \frac{\pi_0}{\min\{u_i^0\}} \equiv c\ ,$$

so (7.4) holds for *both* choices of π^k.

The most *efficient* and *stable* implementation of the PBM algorithm was achieved with the logarithmic-quadratic penalty function φ (see example 4 in section 3); the reciprocal-quadratic φ (example 5 in section 3) was also successful and only slightly inferior. Compared with a pure (shifted) logarithmic penalty, the number of Newton steps for a logarithmic-quadratic penalty was usually reduced 2–3 times, particularly for large-scale problems.

By "stable," we mean that the algorithm's performance was not affected too much by the choice of the parameters (u^0, μ, π_0, etc.). By "efficient," we mean that the number of Newton steps grows very slowly with the dimension of the problem. This is demonstrated clearly in Tables 1 and 2 in the next section.

Empirically, we observed that after achieving an accuracy of 4–5 digits in the objective function value $f(x^k)$, every additional iteration required only one Newton step, adding typically a digit of accuracy. (An analogous fact was demonstrated in [14] for the modified barrier function (MBF) (shifted log) method in the case of linear programming.) Due to this property, the method is particularly efficient when high accuracy is required.

8. Numerical results for large-scale structural optimization problems. The PBM algorithm with a log-quadratic penalty was applied to solve two types of problems in structural optimization: (1) truss topology design and (2) shape design.

Truss topology design (TTD). The original formulation of the problem is the following (see [3, 5] for details):

$$\text{(TTD)} \qquad \begin{array}{l} \min_{x,t} f^T x \\ \text{subject to} \\ \begin{array}{rcll} A(t)x &=& f, & \\ \sum_{i=1}^m t_i &=& v, & \\ t_i &\geq& 0\,, & i = 1, \ldots, m, \end{array} \end{array} \tag{8.1}$$

where

N	=	number of nodes in the truss,
m	=	maximum number of potential bars in the truss ($m = \frac{1}{2}N(N-1)$),
$t = (t_i)$	=	m-dimensional vector of the bar volumes (design variables),
n	=	number of analysis variables $n = 2N$ (2D-trusses) or $n = 3N$ (3D-trusses),
$x = (x_j)$	=	n-dimensional vector of the displacements of the nodes (analysis variables),
f	=	n-dimensional vector of external loads,
v	=	total volume of the truss,
$A(t)$	=	symmetric positive semidefinite $n \times n$ matrix, the stiffness matrix.

The matrix $A(t)$ is given in terms of matrices A_i, which are also symmetric positive semidefinite (PSD) $n \times n$ matrices:

$$A(t) = \sum_{i=1}^m t_i A_i.$$

Each A_i contains information on the geometry of the connection of node i to the other nodes.

In [3], it was proved that problem (TTD) is equivalent to the following minimax problem:

$$\min_x \left\{ F(x) = \max_{1 \leq i \leq m} \left\{ \frac{v}{2} x^T A_i x - f^T x \right\} \right\}.$$

Table 1
Numerical results for the truss topology design problem (8.2).

# of variables	# of constraints	# of Newton steps		CPU time per Newton step
		$\pi^k(t) = \pi_0\mu^k t$ (i)	$\pi^k(t) = \pi_0\mu^k$ (ii)	
88	603	26	28	0.017
98	150	16	12	0.024
126	1234	31	28	0.051
162	2040	32	25	0.11
192	2852	27	25	0.18
242	4492	34	27	0.36
338	8744	42	38	1.00
342	8958	42	31	1.03
450	15556	90	54	2.35
462	16290	48	39	2.54

This minimax we can rewrite as the following quadratically constrained minimization problem:

$$\begin{aligned} &\min z \\ &\text{s.t. } z - \frac{v}{2} x^T A_i x + f^T x \geq 0, \quad i = 1, 2, \ldots, m, \end{aligned} \tag{8.2}$$

and we applied the PBM method to this latter formulation. The Newton steps were performed by using the routine EO4LBF from the NAG library. The results are given in Table 1. All tests were performed on an IBM RS 6000 workstation. Accuracy is six digits.

The results in Table 1 show a slightly better performance for choice (ii) of the penalty updating function π^k. The problem with 450 variables and 15556 (quadratic) constraints is particularly difficult due to a large number of "almost" active constraints in the optimal solution (many thin bars in the optimal truss). It could not be solved without the safeguard rule (7.2), due to ill-conditioning of the Newton system. The linear π^k could solve all but two of the large-scale problems without using (7.2), but the number of Newton steps could increase 1.5–2.5 times.

Shape design. In [6], a mathematical model is constructed which describes the problem of minimizing the compliance of a mechanical structure made of a given material, in which the material properties themselves appear in the role of design variables.

The final finite element discretization of the continuous problem leads to a formulation similar to (TTD) which further reduces to a quadratic minimax problem of the type

$$\min_{x \in \mathbb{R}^N} \max_{i=1,\ldots,M} \{x^T A_i x - f^T x\},$$

where M is the number of finite elements approximating the elastic continuum in question and $N \approx 2M$ is the dimension of the "displacement field" vector x. In our tests, the finite element mesh was in the range 14×14 to 56×56. All A_i's are small rank positive semidefinite matrices. The minimax problem can be reformulated as (see [5])

$$\left\{ \begin{aligned} &\qquad \min_{z \in \mathbb{R}^N} f^T z \\ &\text{subject to} \\ &\qquad z^T A_i z \leq 1, \quad 1 = 1, \ldots, M. \end{aligned} \right. \tag{8.3}$$

TABLE 2
Numerical results for the shape design problem (8.3).

# of variables	# of constraints	# of Newton steps		CPU time per Newton step
		$\pi^k(t) = \pi_0\mu^k t$	$\pi^k(t) = \pi_0\mu^k$	
450	195	20	20	0.003
800	361	25	25	0.02
1682	784	28	25	0.017
2800	1311	41	31	0.81
3000	1421	38	29	1.00
3200	1521	33	27	1.20
3600	1711	40	28	1.72
5000	2401	48	28	4.62
6000	2871	44	29	8.00
6498	3136	32	28	10.01

The Hessian matrix of augmented Lagrangian $F(\cdot, u, p)$ for the problem (8.2) is sparse and, moreover, has the same pattern of nonzero elements as the matrix $A(t)$ in the equilibrium equation (8.1). Therefore, to solve the Newton system, we used a standard solver for finite elements equilibrium equations (see Chap. 6 in [2]). Compared to a Cholesky decomposition scheme (used in the TTD problem), this solver improves computing time by a factor of 100 for very large problems.

Results for running the PBM method on problem (8.3) for different sizes are given in Table 2.

Acknowledgments. The authors are indebted to helpful discussions with Alfred Auslender, Michael Cočvara, and Marc Teboulle which improved the quality of this paper.

REFERENCES

[1] A. Auslender, R. Cominett, and M. Haddou (1997), *Asymptotic analysis of penalty and barrier methods in convex and linear programming*, Math. Oper. Res., to appear.

[2] K. J. Bathe and E. L. Wilson (1976), *Numerical Methods in Finite Element Analysis*, Prentice–Hall, Englewood Cliffs, NJ.

[3] A. Ben-Tal and M. P. Bendsøe (1993), *A new method for optimal truss topology design*, SIAM J. Optim., 3, pp. 322–358.

[4] A. Ben-Tal, I. Yuzefovich, and M. Zibulevsky (1992), *Penalty/Barrier Multiplier Methods for Minmax and Constrained Smooth Convex Programs*, Research report 9, Optimization Laboratory, Faculty of Industrial Engineering and Management, Technion—Israel Institute of Technology, Haifa, Israel.

[5] M. P. Bendsøe, A. Ben-Tal, and J. Zowe (1994), *Optimization methods for truss geometry and topology design*, Structural Optim., 7, pp. 141–159.

[6] M. P. Bendsøe, J. M. Guedes, R. B. Haber, P. Pedersen, and J. E. Taylor (1994), *An analytical model to predict optimal material properties in the context of optimal structural design*, J. App. Mech., 61, pp. 930–937.

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

[8] M. G. Breitfeld and D. F. Shanno (1993), *Computational Experience with Modified Log-Barrier Methods for Nonlinear Programming*, Rutcor research report, Rutgers University, New Brusnwick, NJ.

[9] A. V. Fiacco and G. P. McCormick (1990), *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Classics in Applied Mathematics, SIAM, Philadelphia, PA.

[10] A. N. Iusem, B. Svaiter, and M. Teboulle (1994), *Entropy-like proximal methods in convex programming*, Math. Oper. Res., 19, pp. 790–814.

[11] B. W. Kort and D. P. Bertsekas (1972), *A new penalty function method for constrained optimization*, in Proc. 1971 IEEE Decision and Control Conference, New Orleans, LA.

[12] B. W. Kort and D. P. Bertsekas (1973), *Multiplier methods for convex programming*, in Proc. 1973 IEEE Decision and Control Conference, San Diego, CA.

[13] A. Melman (1992), *Complexity Analysis for the Newton Modified Barrier Function Method*, Ph.D. thesis, Applied Mathematics, California Institute of Technology, Pasadena, CA.

[14] R. Polyak (1992), *Modified Barrier Functions: Theory and Methods*, Math. Programming, 54, pp. 177–222.

[15] R. T. Rockafellar (1970), *Convex Analysis*, Princeton University Press, Princeton, NJ.

[16] R. T. Rockafellar (1973), *A dual approach to solving nonlinear programming problems by unconstrained optimization*, Math. Programming, 5, pp. 354–373.

[17] R. T. Rockafellar (1976), *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Math. Oper. Res., 1, pp. 97–116.

[18] M. Teboulle (1992), *Entropic proximal mappings with applications to nonlinear programming*, Math. Oper. Res., 17, pp. 670–690.

[19] P. Tseng and D. P. Bertsekas (1993), *On the convergence of the exponential multiplier method for convex programming*, Math. Programming, 60, pp. 1–19.

SIAM J. Optim.
Vol. 7, No. 2, pp. 367–385, May 1997

PRACTICAL ASPECTS OF THE MOREAU–YOSIDA REGULARIZATION: THEORETICAL PRELIMINARIES*

CLAUDE LEMARÉCHAL† AND CLAUDIA SAGASTIZÁBAL†

Abstract. When computing the infimal convolution of a convex function f with the squared norm, the so-called Moreau–Yosida regularization of f is obtained. Among other things, this function has a Lipschitzian gradient. We investigate some more of its properties, relevant for optimization. The most important part of our study concerns second-order differentiability: existence of a second-order development of f implies that its regularization has a Hessian. For the converse, we disclose the importance of the decomposition of $\mathbb{R}^N$ along $\mathcal{U}$ (the subspace where f is "smooth") and $\mathcal{V}$ (the subspace parallel to the subdifferential of f).

Key words. convex optimization, mathematical programming, proximal point, second-order differentiability

AMS subject classifications. Primary, 26B05; Secondary, 65K10

PII. S1052623494267127

1. Introduction.

The motivation for this paper is to explore the possibility of introducing efficient preconditioners into the proximal-point algorithm to minimize a convex function f. This algorithm (see [2], [13], [23]) is essentially an implicit (sub)gradient method. However, it is much more fruitful to see it as the ordinary gradient method applied to a certain perturbation of f: the Moreau–Yosida regularization (see [15], [27]), whose minima coincide with those of f. The introduction of a preconditioner into this gradient method is thus natural; first steps in this direction were already made in [21], [3]. Naturally, such a preconditioner has to exploit the second-order properties of the perturbed objective function; a study of these properties is therefore a prerequisite to the development of any reasonable algorithm. We address this last, purely theoretical, question here; we also study some other properties relevant for optimization. Specifically, we relate the smoothness, behavior at infinity, and strong convexity of an objective function to the corresponding properties of its Moreau–Yosida regularization; for this, we use extensively the technical results of [11]. The companion paper [28] exploits the results obtained here to develop some related algorithms, emphasizing the implementable aspect; along these lines, we also mention the computational considerations contained in [5], [1], [9], [25], [14], [10].

Our notation follows closely that of [22] and [7]. In the space $\mathbb{R}^N$, the Euclidean product is denoted by $\langle \cdot, \cdot \rangle$, and $\| \cdot \|$ is the associated norm; $B(x, \rho)$ is the ball centered at x with radius ρ. The conjugate of a closed (i.e., lower semicontinuous) convex function φ is

$$\varphi^*(g) := \sup_{x \in \mathbb{R}^N} \{ \langle g, x \rangle - \varphi(x) \} . \tag{1}$$

Recall that the conjugacy operation is an involution; i.e., the conjugate of φ^* is φ itself. The indicator function of a closed convex set S (0 on S, $+\infty$ outside) is denoted by I_S. Given a symmetric positive definite linear operator M, we set $\langle \cdot, \cdot \rangle_M := \langle M \cdot, \cdot \rangle$;

* Received by the editors May 4, 1994; accepted for publication (in revised form) October 31, 1995.

http://www.siam.org/journals/siopt/7-2/26712.html

† INRIA, BP 105, 78153 Le Chesnay, France (claude.lemarechal@inria.fr, claudia.sagastizabal@inria.fr).

accordingly, we will shorten $\frac{1}{2}\|x\|_M^2 := \frac{1}{2}\langle x,x\rangle_M$, whose conjugate is $\frac{1}{2}\|g\|^2_{M^{-1}}$. The smallest and largest eigenvalues of M will be denoted by λ and Λ, respectively.

We denote by F the Moreau–Yosida regularization of a given closed convex function f associated to the metric defined by M:

$$F(x) := \min_{y\in\mathbb{R}^n}\left\{f(y) + \tfrac{1}{2}\|y-x\|_M^2\right\} =: \left(f \mathbin{\overset{+}{\vee}} \tfrac{1}{2}\|\cdot\|_M^2\right)(x)\,, \tag{2}$$

where $\overset{+}{\vee}$ stands for the infimal convolution. The dual relation

$$F^*(\cdot) = f^*(\cdot) + \frac{1}{2}\|\cdot\|^2_{M^{-1}}$$

will be used continually in this paper. First-order regularity of F is well known; without any further assumption, F has a Lipschitzian gradient. More precisely, for all x_1, $x_2 \in \mathbb{R}^N$,

$$\|\nabla F(x_1) - \nabla F(x_2)\|^2 \le \Lambda\,\langle\nabla F(x_1) - \nabla F(x_2), x_1 - x_2\rangle\,. \tag{3}$$

(Note that the Lipschitz property comes with Cauchy–Schwarz.) If we denote by $p(x)$ the unique minimizer in (2), called the *proximal* point of x, $\nabla F(x)$ has the following expression:

$$G := \nabla F(x) = M(x - p(x)) \in \partial f(p(x))\,. \tag{4}$$

Note in particular that f has a nonempty subdifferential at any point p of the form $p(x)$.

Our paper is organized as follows. First, we review a few elementary results on the Moreau–Yosida regularization F of (2), which are relevant when developing optimization algorithms. Some of them are easy and/or already known, at least for $M = \mathcal{I}$, the identity operator. Then in section 3 comes the main content of this paper: a study of second-order differentiability. We give a detailed answer to the question "when does F have a *Hessian*?" We also touch on second-order differentiability in the *epigraphical* sense, [24], which yields related but complementary results. This question is also addressed in [12] and [17]. In this last reference, the fairly general class of prox-regular functions, which contains lower-C^2, primal-lower-nice, and strongly amenable functions, is considered. Our present work is limited to a convex f; moreover, we will often consider the finite-valued case, $f : \mathbb{R}^N \to \mathbb{R}$. This avoids some technical difficulties and makes the reading lighter.

2. Properties of the Moreau–Yosida regularization.

We study here some properties which F of (2) inherits from f.

We first show that f and F have the same behavior at infinity. Recall that the recession (or asymptotic) function of a closed convex function φ is defined by

$$\varphi'_\infty(d) := \lim_{t\to+\infty}[\varphi(x+td) - \varphi(x)]/t$$

(a limit which does not depend on $x \in \operatorname{dom}\varphi$). This function is useful because φ has a nonempty bounded set of minima if and only if $\varphi'_\infty(d) > 0$ for all $d \neq 0$.

THEOREM 2.1. *The recession functions of f and F are identical.*

Proof. Apply Corollary 9.2.1 in [22]: the recession function of an infimal convolution is the infimal convolution of the recession functions. Because the recession function of a squared norm is clearly $\mathrm{I}_{\{0\}}$, we obtain

$$F'_\infty(d) = \left(f'_\infty \mathbin{\overset{+}{\vee}} \mathrm{I}_{\{0\}}\right)(d) = \inf_{y=0} f'_\infty(d-y) = f'_\infty(d). \qquad \square$$

Recall that a function φ is said to be *strongly convex* with modulus $c > 0$ if and only if $\varphi(\cdot) - \frac{1}{2}c\|\cdot\|^2$ is a convex function. This property plays the role of nondegenerate Hessians in smooth optimization; as such, it is fairly relevant for optimization algorithms. We show that strong convexity is transmitted between f and F. Dually, smoothness is likewise transmitted between f^* and F^*.

THEOREM 2.2. *For a finite-valued convex function f, the following statements are equivalent:*

(i) *f is strongly convex with modulus $1/\ell$;*

(ii) *f^* has a Lipschitzian gradient with Lipschitz constant ℓ;*

(iii) *F^* has a Lipschitzian gradient with Lipschitz constant L;*

(iv) *F is strongly convex with modulus $1/L$.*

Furthermore, we have the inequalities $\ell - 1/\lambda \le L \le \ell + 1/\lambda$.

Proof. Because f and F are finite valued, Theorems X.4.2.1 and X.4.2.2 in [7] can be applied to yield the equivalences (i) $\iff$ (ii) and (iii) $\iff$ (iv).

Let us prove (ii) $\iff$ (iii). Since F is the infimal convolution of f and $\frac{1}{2}\|\cdot\|_M^2$, its conjugate is the sum of the respective conjugates: $F^*(\cdot) = f^*(\cdot) + \frac{1}{2}\|\cdot\|^2_{M^{-1}}$. Actually,

$$\nabla F^*(\cdot) = \nabla f^*(\cdot) + M^{-1}(\cdot)$$

whenever one of the gradients exists. The equivalence between the Lipschitz properties is then clear; as for the relations between the constants, apply appropriate triangular inequalities. □

We now turn our attention to properties involving the proximal operator more directly. They will be useful for the study of second-order smoothness.

PROPOSITION 2.3. *For any x_1 and x_2 in $\mathbb{R}^N$,*

$$\|p(x_1) - p(x_2)\|_M^2 \le \langle x_1 - x_2, p(x_1) - p(x_2)\rangle_M \,. \tag{5}$$

It follows that the mapping $x \mapsto p(x)$ is Lipschitzian with constant Λ/λ.

Proof. For arbitrary $p_1, p_2 \in \mathbb{R}^N$ and $G_i \in \partial f(p_i)$, the convexity of f gives the monotonicity of the subgradients; $\langle G_1 - G_2, p_1 - p_2\rangle \ge 0$. Now take x_1 and x_2 in $\mathbb{R}^N$, and write the inequality for $p_i := p(x_i)$, G_i from (4):

$$\langle M(x_1 - p(x_1)) - M(x_2 - p(x_2)), p(x_1) - p(x_2)\rangle \ge 0\,,$$

which is (5). From this we can obtain

$$\lambda\|p(x_1) - p(x_2)\|^2 \le \Lambda\|x_1 - x_2\|\;\|p(x_1) - p(x_2)\|,$$

and the Lipschitz property follows immediately. □

PROPOSITION 2.4. *Assume f is a closed convex function. Then $\nabla F(\cdot)$ has directional derivatives if and only if $p(\cdot)$ has directional derivatives. The Hessian $\nabla^2 F(x)$ exists if and only if the Jacobian $\nabla p(x)$ exists:*

$$\nabla^2 F(x) = M(\mathcal{I} - \nabla p(x)) \quad \text{for all } x \in \mathbb{R}^N.$$

Proof. The proof is straightforward from (4). □

As observed in [14], a space decomposition may be important when combining quasi-Newton updates with proximal-point algorithms. Along these lines, we show that when $x \to x_0$, $p(x)$ is asymptotically close to the normal cone to $\partial f(p(x_0))$ at G. First we recall a well-known property of convex functions.

LEMMA 2.5. *Let f be a closed convex function and let $z_0 \in \operatorname{ri}\operatorname{dom} f$. Suppose $t \downarrow 0$ and $(z-z_0)/t$ has a cluster point ℓ; let $g \in \partial f(z)$ have a cluster point $g_0 \in \partial f(z_0)$. Then $\ell \in \mathcal{N} := \mathrm{N}_{\partial f(z_0)}(g_0)$, the normal cone to $\partial f(z_0)$ at g_0.*

Proof. Take any $\gamma \in \partial f(z_0)$; from convexity, $\langle g-\gamma, z-z_0\rangle \geq 0$. Dividing by $t>0$ and passing to the limit, we obtain $\langle g_0-\gamma, \ell\rangle \geq 0$. □

COROLLARY 2.6. *When $x \to x_0$, all the cluster points of $\frac{p(x)-p(x_0)}{\|x-x_0\|}$ lie in $\mathcal{N} \cap B(0, \Lambda/\lambda)$. As a result, if $p(\cdot)$ has a Jacobian $\nabla p(x)$, then $\operatorname{Im}\nabla p(x) \subset \mathcal{N}$.*

Proof. Set $g_0 = G = \nabla F(x_0) \in \partial f(p(x_0))$, $z_0 = p(x_0)$, $z = p(x)$, $g = \nabla F(x) \in \partial f(p(x))$, and $t = \|x - x_0\|$. Because F is continuously differentiable, $g \to G$. Then apply Lemma 2.5 and Proposition 2.3. □

A direct consequence is that ∇F enjoys automatically some directional differentiability.

COROLLARY 2.7. *For the closed convex function f, let G be defined by* (4), *and denote by $\mathcal{T}$ the tangent cone to $\partial f(p(x))$ at G. Then, for any d such that $Md \in \mathcal{T}$,*

$$\frac{\nabla F(x+td)-\nabla F(x)}{t} \longrightarrow Md \quad \textit{when}\, t \downarrow 0.$$

Proof. From (4), $\nabla F(x+td) - \nabla F(x) = tMd - M(p(x+td)-p(x))$; we only need to show that $[p(x+td)-p(x)]/t$ tends to 0 when $t \downarrow 0$. For this, use (5):

$$\langle M(p(x+td)-p(x)), p(x+td)-p(x)\rangle \leq t\,\langle p(x+td)-p(x), Md\rangle \;.$$

Observing that the left-hand side is minorized by $\lambda\|p(x+td)-p(x)\|^2$, divide by t^2 to obtain

$$0 \leq \lambda \frac{\|p(x+td)-p(x)\|^2}{t^2} \leq \left\langle \frac{p(x+td)-p(x)}{t}, Md \right\rangle .$$

In view of Corollary 2.6, the (bounded) right-hand side cannot have any positive cluster point; it must tend to 0 and the proof is complete. □

Of course, owing to the Lipschitz property of ∇F, a classical argument enables the following improvement of this directional result: if $x \to x_0$ in such a way that $(x-x_0)/\|x-x_0\| \to d$ with $Md \in \mathcal{T}$, then

$$\frac{\nabla F(x)-\nabla F(x_0)}{\|x-x_0\|} \longrightarrow Md.$$

To illustrate Corollary 2.7, take the bivariate function $f(\xi,\eta) = |\xi| + \frac{1}{2}\eta^2$ and $M = \mathcal{I}$. The optimality condition for the proximal point (π,ρ) of (ξ,η) close to 0 results in

$$\pi = 0 \quad \text{if } |\xi| \leq 1 \qquad \text{and} \qquad \rho = \eta/2.$$

Thus, at $x = 0$, $\partial f(x) = [-1,1] \times \{0\}$, $p(x) = 0$, $G = 0$, and $p(\cdot)$ has the Jacobian $\left(\begin{smallmatrix} 0 & 0 \\ 0 & 1/2 \end{smallmatrix}\right)$. We see that the nondifferentiability of f at 0 in the subspace $\mathcal{T} = \mathbb{R} \times \{0\}$ does not affect the second-order differentiability of F.

We conclude this section with a trivial but often forgotten observation: the proximal mapping has an explicit inverse. This may be very useful when designing algorithms; see [10].

THEOREM 2.8. *Let p be such that $\partial f(p) \neq \emptyset$ and take $G \in \partial f(p)$. Then p is the proximal point of $x := p + M^{-1}G$.*

Proof. We have $M(x-p) \in \partial f(p)$ and this characterizes the proximal point in a unique way; see (4). □

3. Second-order analysis. The aim of this section is to relate second-order derivatives of F and f. For the continuously differentiable F, there is no need of generalizing the classical notion of Hessian. For f, however, the multivalued ∂f calls for a special concept. We will say that the finite-valued convex function f admits at z_0 a generalized Hessian $\mathrm{H}f(z_0)$ when

(i) the gradient $\nabla f(z_0)$ exists,

(ii) there exists a symmetric positive semidefinite operator $\mathrm{H}f(z_0)$ such that

$$f(z_0+h) = f(z_0) + \langle \nabla f(z_0), h\rangle + \frac{1}{2}\langle \mathrm{H}f(z_0)h, h\rangle + o(\|h\|^2)\,. \tag{6}$$

An important result of [6] is that (6) is equivalent to

$$\partial f(z_0+h) \subset \nabla f(z_0) + \mathrm{H}f(z_0)h + o(\|h\|)B\,, \tag{7}$$

where B is the unit ball. Note also that when ∂f is single valued in a neighborhood of z_0, $\mathrm{H}f(z_0)$ is the classical Hessian $\nabla^2 f(z_0)$.

We present our study in several steps. First, we consider f strongly convex and differentiable; next, we eliminate the differentiability assumption. Finally, we take a general convex finite-valued function. We are interested in relating the existence of $\nabla^2 F(x_0)$ and $\mathrm{H}f(p_0)$, with $p_0 = p(x_0)$. The following growth condition plays a central role for most of our results:

$$f(p_0+h) \le f(p_0) + f'(p_0;h) + \frac{1}{2}C\|h\|^2 \quad \text{for some } C>0 \text{ and all } h\in B(0,\varepsilon). \tag{8}$$

3.1. Differentiable case. The following result has been proved independently by J.-B. Hiriart-Urruty and L. Q. Qi.

THEOREM 3.1. *Let the finite-valued convex function f have a generalized Hessian at* $p(x_0)$. *Then the Hessian of F exists at* x_0; *more precisely,*

$$\nabla^2 F(x_0) = M - M[\mathrm{H}f(p(x_0)) + M]^{-1}M\,.$$

Proof. In view of Proposition 2.4, we only need to exhibit $\nabla p(x_0)$. Write (7) with z_0 and z_0+h replaced by $p(x_0)$ and $p(x_0+h)$, respectively:

$$\partial f(p(x_0+h)) \subset \nabla f(p(x_0)) + \mathrm{H}f(p(x_0))(p(x_0+h)-p(x_0)) + o(\|p(x_0+h)-p(x_0)\|)B\,.$$

Because $p(\cdot)$ is Lipschitzian (Proposition 2.3), $o(\|p(x_0+h)-p(x_0)\|) = o(\|h\|)$. Multiply by M^{-1} and add $p(x_0+h)$ to both sides to obtain the following:

$$\begin{aligned} M^{-1}\partial f(p(x_0+h)) + p(x_0+h) \quad \subset \quad & M^{-1}\nabla f(p(x_0)) + p(x_0+h) \\ & + M^{-1}\mathrm{H}f(p(x_0))(p(x_0+h)-p(x_0)) + o(\|h\|)B \\ = \quad & M^{-1}\nabla f(p(x_0)) + p(x_0) \\ & \quad + [\mathcal{I} + M^{-1}\mathrm{H}f(p(x_0))](p(x_0+h)-p(x_0)) \\ & \quad\quad + o(\|h\|)B\,. \end{aligned}$$

Now recall Theorem 2.8: the left-hand side contains x_0+h; likewise, $M^{-1}\nabla f(p(x_0)) + p(x_0)$ is the singleton x_0. Thus, we have proved

$$h \in [\mathcal{I} + M^{-1}\mathrm{H}f(p(x_0))](p(x_0+h)-p(x_0)) + o(\|h\|)B\,.$$

Knowing that $\mathcal{I} + M^{-1}\mathrm{H}f(p(x_0))$ is invertible, this can also be written

$$p(x_0 + h) - p(x_0) \in [\mathcal{I} + M^{-1}\mathrm{H}f(p(x_0))]^{-1}h + o(\|h\|)B\,,$$

which means that $[\mathcal{I}+M^{-1}\mathrm{H}f(p(x_0))]^{-1}$ is exactly the gradient of p at x_0. Therefore, $\nabla^2 F(x_0) = M(\mathcal{I} - \nabla p(x_0)) = M - M[\mathcal{I} + M^{-1}\mathrm{H}f(p(x_0))]^{-1}$. ☐

COROLLARY 3.2. *Let the finite-valued convex function f have a generalized Hessian at $p(x_0)$. Then $\nabla^2 F(x_0)$ exists and*

$$\ker \nabla^2 F(x_0) = \ker \mathrm{H}f(p(x_0)).$$

Proof. Use the notation $H := \mathrm{H}f(p(x_0))$ and $H' := \nabla^2 F(x_0)$. From Theorem 3.1, $M^{-1}H' = \mathcal{I} - [H + M]^{-1}M = \mathcal{I} - [M^{-1}H + \mathcal{I}]^{-1}$; hence,

$$\mathcal{I} - M^{-1}H' = (\mathcal{I} + M^{-1}H)^{-1}\,.$$

If $H'v = 0$, then $(\mathcal{I} + M^{-1}H)v = v$ and $Hv = 0$. Taking inverses, $(\mathcal{I} - M^{-1}H')^{-1} = \mathcal{I} + M^{-1}H$ and we show, likewise, that $Hv = 0$ implies $H'v = 0$. ☐

The converse part of Theorem 3.1 is not so simple; it will be stated in Theorems 3.14 and 3.15 below. First, the next geometrical result is crucial.

PROPOSITION 3.3. *Let f be a finite-valued strongly convex function satisfying* (8) *at a given $p_0 = p(x_0)$. If $\nabla^2 F(x_0)$ exists, then G of* (4) *lies in the relative interior of $\partial f(p_0)$.*

Proof. Let F have a Hessian at x_0; hence, by Proposition 2.4, $\nabla p(x_0)$ exists.

Assume for contradiction $G \in \operatorname{rbd} \partial f(p_0)$; by Proposition 2.2 in [11], the normal cone $\mathcal{N} = \mathrm{N}_{\partial f(p_0)}(G)$ is not a subspace. Now use Proposition 2.1 in [11] and the notation therein. We can take a unitary $\nu_0 \in \mathcal{N} \cap \mathcal{M}^\perp$ (with $\mathcal{M} := \mathcal{N} \cap -\mathcal{N}$) such that

$$\nu \in \mathcal{N} \text{ and } \langle \nu_0, \nu\rangle \neq 0 \quad\Longrightarrow\quad -\nu \notin \mathcal{N}\,. \tag{9}$$

Take $G_t := G + t\nu_0$ with $t > 0$; calling c the modulus of strong convexity of f, Theorem 2.2 guarantees that $p_t := \nabla f^*(G_t)$ satisfies the Lipschitz condition

$$\|p_t - p_0\| \leq \frac{1}{c}\|G_t - G\| = \frac{1}{c}t\,.$$

By Theorem 2.8, this p_t is the proximal point of $x_t := p_t + M^{-1}G_t$ which, therefore, satisfies

$$\|x_t - x_0\| \leq \|p_t - p_0\| + \frac{1}{\lambda}\|G_t - G\| \leq \left(\frac{1}{c} + \frac{1}{\lambda}\right)t. \tag{10}$$

Furthermore, since (8) holds, we can apply Corollary 3.3 in [11], with $\varphi = f$, $r = \frac{1}{2}C\|\cdot\|^2$, $z_0 = p_0$, $x = p_t$, $g_0 = G$, and $s = t\nu_0 \in \mathcal{N}$: whenever $t \in [0, \varepsilon C/2]$,

$$\langle G_t - G, p_t - p_0\rangle \geq \frac{1}{2C}\|G_t - G\|^2 = \frac{1}{2C}t^2\,.$$

Combine this equation with (10):

$$\frac{\lambda c}{2C(c+\lambda)} \leq \left\langle \nu_0, \frac{p_t - p_0}{\|x_t - x_0\|}\right\rangle .$$

Let $t \downarrow 0$. By (10), $x_t \to x_0$; extracting a subsequence if necessary, $[p_t - p_0]/\|x_t - x_0\| \to \nu \in \mathcal{N}$ (Corollary 2.6). Clearly, $\langle \nu_0, \nu \rangle > 0$; hence, by (9), $-\nu \notin \mathcal{N}$. This shows that $\mathrm{Im}\nabla p(x_0)$ is not a symmetric set; $\nabla p(x_0)$ cannot be a linear operator, which is the required contradiction. □

We can now establish two second-order results, a local and a global one, valid for strongly convex functions.

PROPOSITION 3.4. *Let f be a finite-valued strongly convex function satisfying* (8) *at a given $p_0 = p(x_0)$. If $\nabla^2 F(x_0)$ and $\nabla f(p_0)$ exist, then* $\mathrm{H}f(p_0)$ *exists.*

Proof. We have from (4) $p_0 = x_0 - M^{-1}G$ with $G = \nabla f(p_0)$. Apply Corollary 3.3 in [11] with $\varphi = f$, $r = \frac{1}{2}C\|\cdot\|^2$, $z_0 = p_0$, and $g_0 = G$; since $\partial f(p_0) = \{G\}$, $G + s$ is projected onto G for all s:

$$f^*(G+s) \geq f^*(G) + \langle s, p_0 \rangle + \frac{1}{2C}\|s\|^2 \quad \text{for } \|s\| \leq \varepsilon C/2\,. \tag{11}$$

By Corollary X.4.2.9 in [7], the existence of $\nabla^2 F(x_0)$ (positive definite, recall Theorem 2.2) implies the existence of $\nabla^2 F^*(G) = \nabla^2 f^*(G) + M^{-1}$. Therefore, $\nabla^2 f^*(G)$ exists and, by (11), is positive definite. Again, by Corollary X.4.2.9 in [7], f has a generalized Hessian at p_0. □

PROPOSITION 3.5. *Let f be a finite-valued strongly convex function satisfying* (8) *at $p_0 = p(x_0)$ for all $x_0 \in \mathbb{R}^N$. If $\nabla^2 F$ exists on the whole of $\mathbb{R}^N$, then f has a classical Hessian $\nabla^2 f$ on the whole of $\mathbb{R}^N$.*

Proof. We claim that f is differentiable at every $p \in \mathbb{R}^N$. Indeed, if $\partial f(p_0)$ is not a singleton, take a subgradient G in the relative boundary of $\partial f(p_0)$ (Proposition 2.3 in [11]). Because of Theorem 2.8, p_0 is the proximal point of $x_0 := p_0 + M^{-1}G$ and, by assumption, $\nabla^2 F(x_0)$ exists. From Proposition 3.3 we get the desired contradiction: G lies in the relative interior of $\partial f(p_0)$.

Then $\nabla^2 F$ and ∇f exist on the whole of $\mathbb{R}^N$ and Proposition 3.4 applies. □

3.2. Nondifferentiable case: The partial proximal operator. In this section, we consider a fixed x_0 such that f has no gradient at $p(x_0)$. In such a situation, can we relate the existence of $\nabla^2 F(x_0)$ with some smoothness of f at $p(x_0)$? To get an idea of what can happen, perturb the bivariate example $f(\xi, \eta) = |\xi| + \frac{1}{2}\eta^2$ at the end of section 2. Add to f an arbitrary convex univariate function $n(\xi)$, as nasty as desired, the extreme cases being $n \equiv 0$ and $n = \mathrm{I}_{\{0\}}$. It is easy to see that F remains the same near 0. In other words, F is totally blind to the behavior of f in the tangent cone $\mathcal{T}$.

We already know that the existence of $\nabla^2 F(x_0)$ implies $G \in \mathrm{ri}\,\partial f(p(x_0))$ (Proposition 3.3). As a result, the normal and tangent cones to $\partial f(p(x_0))$ at G are two complementary subspaces (Proposition 2.2 in [11]). Things are easier to visualize if the following notation is used: we set $\mathcal{U} = \mathcal{N} = \mathrm{N}_{\partial f(p(x_0))}(G)$ and $\mathcal{V} = \mathcal{T} = \mathrm{T}_{\partial f(p(x_0))}(G)$. The reason is that f is smooth or "U-shaped" in $p(x_0) + \mathcal{U}$ and kinky in $p(x_0) + \mathcal{V}$. Accordingly, we use the matrix-like decomposition $H = \left(\begin{smallmatrix} H_{\mathcal{U}\mathcal{U}} & H_{\mathcal{U}\mathcal{V}} \\ H_{\mathcal{V}\mathcal{U}} & H_{\mathcal{V}\mathcal{V}} \end{smallmatrix}\right)$ for linear operators. We denote indifferently by $\mathrm{Proj}_{\mathcal{U}}(s)$ or $s_{\mathcal{U}}$ the projection of s onto $\mathcal{U}$, similarly for $\mathcal{V}$. Note, incidentally, that the important subspace is really $\mathcal{U}$, which is completely defined by f alone; by contrast, $\mathcal{V}$ depends on the scalar product. Furthermore, these two subspaces do not depend on the particular $G \in \mathrm{ri}\,\partial f(p(x_0))$: $\mathcal{V}$ is the subspace parallel to $\mathrm{aff}\,\partial f(p(x_0))$.

LEMMA 3.6. *Let f be a finite-valued strongly convex function with modulus c, satisfying* (8) *at a given p_0. If $\nabla^2 f^*(G)$ exists, then it has the form $\left(\begin{smallmatrix} H^* & 0 \\ 0 & 0 \end{smallmatrix}\right)$, with*

$H^* : \mathcal{U} \to \mathcal{U}$ positive definite:

$$f^*(G+s) = f^*(G) + \langle s, p_0 \rangle + \frac{1}{2}\Big\langle s, \begin{pmatrix} H^* & 0 \\ 0 & 0 \end{pmatrix} s \Big\rangle + o(\|s\|^2)\,. \tag{12}$$

Proof. The existence of $\nabla f^*(G)$ follows from Theorem 2.2. Then write the development (6) for f^* and apply Corollary 3.7 in [11] with $\varphi = f$, $r = \frac{1}{2}c\|\cdot\|^2$, $z_0 = p_0$, and $g_0 = G$. We obtain $\big\langle s, \nabla^2 f^*(G)s \big\rangle \le \frac{1}{2c}\|s_{\mathcal{U}}\|^2$, which implies $\operatorname{Im} \nabla^2 f^*(G) \subset \mathcal{U}$. The symmetric operator $\nabla^2 f^*(G)$ therefore has the stated form.

This establishes (12); let us now prove that H^* is positive definite. Because (8) holds, Corollary 3.4 in [11] applies with $\varphi = f$, $r = \frac{1}{2}C\|\cdot\|^2$, $z_0 = p_0$, and $g_0 = G$:

$$f^*(G+s) \ge f^*(G) + \langle s, p_0 \rangle + \frac{1}{2C}\|s_{\mathcal{U}}\|^2$$

for all $s \in B(0, \varepsilon C/2)$. In particular, if $s \in \mathcal{U} \cap B(0, \varepsilon C/2)$, then $s_{\mathcal{U}} = s$ and we obtain, with (12),

$$f^*(G) + \langle s, p_0 \rangle + \frac{1}{2}\Big\langle s, \begin{pmatrix} H^* & 0 \\ 0 & 0 \end{pmatrix} s \Big\rangle + o(\|s\|^2) \ge f^*(G) + \langle s, p_0 \rangle + \frac{1}{2C}\|s\|^2\,.$$

This clearly implies $\big\langle s, \left(\begin{smallmatrix} H^* & 0 \\ 0 & 0 \end{smallmatrix}\right) s \big\rangle \ge \frac{1}{C}\|s\|^2$ for all $s \in \mathcal{U}$. □

While (12) describes the structure of $\nabla^2 f^*$, it also kills the proof technique used in Proposition 3.4: $\nabla^2 f^*$ is no longer invertible. At this stage we use the cure of section 4 in [11]: we consider the perturbation

$$\phi_{\mathcal{V}}(x) = \min_{y \in x + \mathcal{V}} \{ f(y) - \langle G, y \rangle + \tfrac{1}{2}\|x - y\|^2 \}\,, \quad \phi^*_{\mathcal{V}}(s) := f^*(G+s) + \frac{1}{2}\|s_{\mathcal{V}}\|^2\,. \tag{13}$$

LEMMA 3.7. *Let f be a finite-valued convex function. Take the Moreau–Yosida regularization of $\phi_{\mathcal{V}}$, defined in (13): $\Phi_{\mathcal{V}}(x) := \min_{y \in \mathbb{R}^N}\{\phi_{\mathcal{V}}(y) + \frac{1}{2}\|y - x\|^2_M\}$, and denote by $\pi(x)$ the associated proximal point. Then the following holds:*

(i) $\pi(p_0) = p_0$.

(ii) *$\phi_{\mathcal{V}}$ is strongly convex if and only f is strongly convex.*

(iii) *If f is strongly convex and satisfies (8) at a given p_0, then $\mathrm{H}\phi_{\mathcal{V}}(p_0)$ exists if and only if $\nabla^2 \Phi_{\mathcal{V}}(p_0)$ exists. In this case, $\mathrm{H}\phi_{\mathcal{V}}(p_0) = \left(\begin{smallmatrix} H^{*-1} & 0 \\ 0 & \mathcal{I}_{\mathcal{V}} \end{smallmatrix}\right)$.*

Proof. (i): Use Theorem 2.8 at $p = p_0$, with f replaced by $\phi_{\mathcal{V}}$: $p_0 = \pi(p_0 + M^{-1}\gamma)$ for any $\gamma \in \partial\phi_{\mathcal{V}}(p_0)$. But Proposition 4.1 of [11] used with $z_0 = p_0$ gives $\partial\phi_{\mathcal{V}}(p_0) = \nabla\phi_V(p_0) = 0$. Thus, $\pi(p_0) = p_0$.

(ii): Theorem 2.2 yields the following chain of equivalences: f strongly convex $\iff$ ∇f^* Lipschitzian $\iff$ $\nabla(f^*(G + \cdot) + 1/2\|\operatorname{Proj}_{\mathcal{V}}(\cdot)\|^2)$ Lipschitzian $\iff$ $(f^*(G + \cdot) + 1/2\|\operatorname{Proj}_{\mathcal{V}}(\cdot)\|^2)^*$ strongly convex. The result follows from (13).

(iii): Because f is strongly convex, f^* is finite valued. Furthermore, due to Corollary 3.3 in [11], the assumptions of Proposition 4.2 in [11] hold and we have, for h small enough, $\phi_{\mathcal{V}}(p_0 + h) \le \phi_{\mathcal{V}}(p_0) + 1/2C'\|h\|^2$. This is the growth condition (8) for $\phi_{\mathcal{V}}$ (recall $\nabla\phi_{\mathcal{V}}(p_0) = 0$). Write Theorem 3.1 and Proposition 3.4, with f, F, x_0, p_0 replaced by $\phi_{\mathcal{V}}, \Phi_{\mathcal{V}}, p_0, p_0$, to obtain the stated equivalence.

Finally, when $\mathrm{H}\phi_{\mathcal{V}}(p_0)$ exists, it is positive definite and its inverse is $\nabla^2\phi^*_{\mathcal{V}}(0)$ (Corollary X.4.2.9 of [7]). Because of (13), $\nabla^2\phi^*_{\mathcal{V}}(0) = \nabla^2 f^*(G) + \operatorname{Proj}_{\mathcal{V}}$, and because of Lemma 3.6, the diagonal form follows. □

This enables us to state the key relation for the present nondifferentiable case.

PROPOSITION 3.8. *Let f be a finite-valued strongly convex function satisfying* (8) *at a given $p_0 = p(x_0)$. Then $\nabla^2 F(x_0)$ exists if and only if* $\mathrm{H}\phi_{\mathcal{V}}(p_0)$ *exists.*

Proof. In view of Lemma 3.7(iii), we have to prove the equivalence "$\nabla^2 F(x_0)$ exists $\iff$ $\nabla^2 \Phi_{\mathcal{V}}(p_0)$ exists." From Theorem 2.2, F is strongly convex; hence, by using Corollary X.4.2.9 in [7],

$$\exists \nabla^2 F(x_0) \iff \exists \nabla^2 F^*(G) \iff \exists \nabla^2 f^*(G) = \nabla^2 F^*(G) - M^{-1}\,.$$

Because of (13), this is further equivalent to (recall $\nabla\phi_{\mathcal{V}}(p_0) = \nabla\Phi_{\mathcal{V}}(p_0) = 0$)

$$\exists \nabla^2 \phi_{\mathcal{V}}^*(0) = \nabla^2 f^*(G) + \mathrm{Proj}_{\mathcal{V}} \iff \exists \nabla^2 \Phi_{\mathcal{V}}^*(0) = \nabla^2 \phi_{\mathcal{V}}^*(0) + M^{-1}\,.$$

Finally, $\nabla^2 \Phi_{\mathcal{V}}^*(0)$ is positive definite; by Corollary X.4.2.9 in [7], this last existence is equivalent to the existence of $\nabla^2 \Phi_{\mathcal{V}}(p_0)$. ◻

We devote the end of the section to interpretations of the above result in terms of the original function f. They crucially rely on the partial proximal operator associated to (13):

$$p_{\mathcal{V}}(x) := \operatorname*{argmin}_{y \in x + \mathcal{V}} \{f(y) - \langle G, y\rangle + \tfrac{1}{2}\|x - y\|^2\}.$$

Remember from Proposition 4.1 of [11] the useful characterization

$$\exists g \in \partial f(p_{\mathcal{V}}(x)) \quad \text{such that} \quad p_{\mathcal{V}}(x) - x = \mathrm{Proj}_{\mathcal{V}}(G - g), \tag{14}$$

as well as

$$\partial \phi_{\mathcal{V}}(x) = -G + \{g \in \partial f(p_{\mathcal{V}}(x)) \;:\; \mathrm{Proj}_{\mathcal{V}}(G - g) = p_{\mathcal{V}}(x) - x\}\,. \tag{15}$$

First of all, the definition (7) of $\mathrm{H}\phi_{\mathcal{V}}(p_0)$ can be translated as follows.

COROLLARY 3.9. *Let f be a finite-valued strongly convex function satisfying* (8) *at a given $p_0 = p(x_0)$. Existence of $\nabla^2 F(x_0)$ is equivalent to the following property. Let $x \to p_0$ and let $g \in \partial f(p_{\mathcal{V}}(x))$ be such that* $\mathrm{Proj}_{\mathcal{V}}(g - G) = x - p_{\mathcal{V}}(x)$. *Then* $g = G + \mathrm{H}\phi_{\mathcal{V}}(p_0)(x - p_0) + o(\|x - p_0\|)$.

Proof. At each x, apply (15): with g as stated, $g - G$ describes $\partial\phi_{\mathcal{V}}(x)$. The result follows from Proposition 3.8, remembering that $\nabla\phi_{\mathcal{V}}(p_0) = 0$. ◻

This result concerns approximations of particular subgradients of f near p_0. Function values can also be approximated along the surface described by $p_{\mathcal{V}}(\cdot)$. In what follows, we study second-order developments of f with respect to the variable $p_{\mathcal{V}}(x)$ rather than x itself.

THEOREM 3.10. *Let f be a finite-valued strongly convex function satisfying* (8) *at a given p_0 such that $H := \mathrm{H}\phi_{\mathcal{V}}(p_0)$ exists. Taking $x = p_0 + h_{\mathcal{U}} + h_{\mathcal{V}}$ with $h_{\mathcal{U}} \to 0$ and $h_{\mathcal{V}} = O(\|h_{\mathcal{U}}\|)$, set $d(x) := p_{\mathcal{V}}(x) - p_0$. Then we have*

$$\begin{gathered} f(p_0 + d(x)) = \\ f(p_0) + \langle G, d(x)\rangle + \tfrac{1}{2}\langle d(x), Hd(x)\rangle - \langle p_{\mathcal{V}}(x) - x, d(x)\rangle + o(\|d(x)\|^2)\,. \end{gathered} \tag{16}$$

Proof. By definition (13) and remembering that $\phi_{\mathcal{V}}(p_0) = f(p_0) - \langle G, p_0\rangle$ (Proposition 4.1 of [11]), the second-order development of $\phi_{\mathcal{V}}$ gives

$$f(p_{\mathcal{V}}(x)) - \langle G, p_{\mathcal{V}}(x)\rangle + \frac{1}{2}\|p_{\mathcal{V}}(x) - x\|^2 = f(p_0) - \langle G, p_0\rangle + \frac{1}{2}\langle h, Hh\rangle + o(\|h\|^2). \tag{17}$$

Now, H has the special form given in Lemma 3.7(iii), so the property $x - p_{\mathcal{V}}(x) \in \mathcal{V}$ implies $H(x - p_{\mathcal{V}}(x)) = x - p_{\mathcal{V}}(x)$; hence, by writing $h = (x - p_{\mathcal{V}}(x)) + (p_{\mathcal{V}}(x) - p_0) = (x - p_{\mathcal{V}}(x)) + d(x)$, we obtain $\langle h, Hh\rangle = \langle d(x), Hd(x)\rangle - 2\,\langle p_{\mathcal{V}}(x) - x, d(x)\rangle + \|p_{\mathcal{V}}(x) - x\|^2$. Plugging this equality into (17), we get

$$f(p_0 + d(x)) = f(p_0) + \langle G, d(x)\rangle + \frac{1}{2}\,\langle d(x), Hd(x)\rangle - \langle p_{\mathcal{V}}(x) - x, d(x)\rangle + o(\|h\|^2)\,.$$

Finally, $\|h\|^2 = \|h_{\mathcal{U}}\|^2 + \|h_{\mathcal{V}}\|^2 = O(\|h_{\mathcal{U}}\|^2)$, but $h_{\mathcal{U}}$ is just the component on $\mathcal{U}$ of $d(x)$; altogether, $\|h\|^2 = O(\|d(x)\|^2)$. □

Beware that (16) is not a regular development of f near p_0. First, it is valid only for special increments $d(\cdot)$ and, at this point, we have not even proved that they tend to 0. Second, what happens to the "extra" term $p_{\mathcal{V}}(x) - x \in \mathcal{V}$? To clarify this situation, we need to bound the difference $p_{\mathcal{V}}(x) - x$.

THEOREM 3.11. *Let f be a finite-valued strongly convex function, satisfying* (8) *at a given p_0 and such that $H := \mathrm{H}\phi_{\mathcal{V}}(p_0)$ exists. For $d = d_{\mathcal{U}} + d_{\mathcal{V}}$ tending to 0 in such a way that*

(i) $\|d_{\mathcal{V}}\| = o(\|d_{\mathcal{U}}\|)$,

(ii) $\exists g \in \partial f(p_0 + d)$ *such that* $\mathrm{Proj}_{\mathcal{V}}(g - G) = O(\|d\|)$,

we have the second-order development

$$f(p_0 + d) = f(p_0) + \langle G, d\rangle + \frac{1}{2}\,\langle d, Hd\rangle + o(\|d\|^2)\,. \tag{18}$$

Proof. With g as in (ii), define $x := p_0 + d + \mathrm{Proj}_{\mathcal{V}}(g - G)$; then, (14) shows that $p_{\mathcal{V}}(x) = p_0 + d$ and $h := x - p_0$ satisfies $h_{\mathcal{U}} = d_{\mathcal{U}}$ and $h_{\mathcal{V}} = d_{\mathcal{V}} + \mathrm{Proj}_{\mathcal{V}}(g - G) = O(\|h_{\mathcal{U}}\|)$. Write (16) and observe that $\langle p_{\mathcal{V}}(x) - x, d\rangle = \langle \mathrm{Proj}_{\mathcal{V}}(g - G), d_{\mathcal{V}}\rangle$. The assumptions (i), (ii) clearly imply that this is $o(\|d\|^2)$. □

This result describes points at which f behaves like a quadratic function. We now show a way of constructing such points.

COROLLARY 3.12. *Let f be a finite-valued strongly convex function satisfying* (8) *at a given p_0 such that $H := \mathrm{H}\phi_{\mathcal{V}}(p_0)$ exists. Let $h \to 0$ with $\|h_{\mathcal{V}}\| = O(\|h_{\mathcal{U}}\|)$ and set $x := p_0 + h$. Then $p_{\mathcal{V}}(x)$ from* (14) *tends to p_0 and* (18) *holds for $d := p_{\mathcal{V}}(x) - p_0 \to 0$.*

Proof. Proceeding as in the proof of Lemma 3.7(iii), we get the assumptions of Corollary 4.3 in [11] (with $z_0 = p_0$): $p_{\mathcal{V}}(\cdot)$ is radially Lipschitzian at p_0, hence $d := p_{\mathcal{V}}(x) - p_0 \to 0$. On the other hand, for some constant $\theta > 0$, $\|h\| \le \theta\|h_{\mathcal{U}}\| = \theta\|d_{\mathcal{U}}\| \le \theta\|d\|$. Because of (14), there is some $g \in \partial f(p_{\mathcal{V}}(x))$ such that $\|\mathrm{Proj}_{\mathcal{V}}(g - G)\| = \|p_{\mathcal{V}}(x) - x\| = O(\|h\|) = O(\|d\|)$. We are in the framework of Theorem 3.11: $d \to 0$, assumption (ii) holds; let us prove that assumption (i) is also satisfied. Any limit point of g lies in $\partial f(p_0)$ (graph closedness of ∂f). Since $\mathcal{V} = \mathrm{aff}\,\partial f(p_0) - G$, the property $\|\mathrm{Proj}_{\mathcal{V}}(g - G)\| = O(\|d\|)$ actually implies $\|g - G\| = O(\|d\|)$. Using Lemma 2.5 with $g_0 = G$, $z_0 = p_0$, $z = p_{\mathcal{V}}(x)$, and $t = \|p_{\mathcal{V}}(x) - p_0\|$, we deduce that any limit point of $\frac{p_{\mathcal{V}}(x) - p_0}{\|p_{\mathcal{V}}(x) - p_0\|}$ lies in $\mathcal{U}$: assumption (i) of Theorem 3.11 holds. □

Let us summarize our results: appropriate assumptions (strong convexity, growth condition, existence of $\nabla^2 F(x_0)$) provide the following second-order information:

(i) $\nabla^2 F(x_0)$ is positive definite (Theorem 2.2); $\nabla^2 f^*(G) = \nabla^{-2} F(x_0) - M^{-1}$ exists and is completely characterized by its $\mathcal{UU}$-block (Lemma 3.6).

(ii) $\nabla p(x_0)$ exists and, in view of Corollary 2.6, has the form $\nabla p(x_0) = \left(\begin{smallmatrix} P & 0 \\ T & 0 \end{smallmatrix}\right)$. From Proposition 2.4 we have $\nabla^{-2} F(x_0) = \nabla^2 F^*(G) = [\mathcal{I} - \nabla p(x_0)]^{-1} M^{-1}$.

(iii) A "partial" generalized Hessian of f at p_0, as described by (18), exists in $\mathcal{U}$. It is the $\mathcal{UU}$-block of the (diagonal) operator $[\nabla^{-2} F(x_0) - M^{-1} + \mathrm{Proj}_{\mathcal{V}}]^{-1}$ and we denote it by $\mathrm{H}_{\mathcal{U}} f(p_0)$.

This last block turns out to have the simple expression

$$\mathrm{H}_{\mathcal{U}} f(p_0) = M_{\mathcal{U}\mathcal{U}}(P^{-1} - \mathcal{I}_{\mathcal{U}})\,. \tag{19}$$

To see this, we need two results from linear algebra, stated without proof.

- For $M = \left(\begin{smallmatrix} M_{\mathcal{U}\mathcal{U}} & M_{\mathcal{U}\mathcal{V}} \\ M_{\mathcal{U}\mathcal{V}}^T & M_{\mathcal{V}\mathcal{V}} \end{smallmatrix}\right)$ and $M^{-1} = \left(\begin{smallmatrix} W_{\mathcal{U}\mathcal{U}} & W_{\mathcal{U}\mathcal{V}} \\ W_{\mathcal{U}\mathcal{V}}^T & W_{\mathcal{V}\mathcal{V}} \end{smallmatrix}\right)$, it holds that
$$M_{\mathcal{U}\mathcal{U}}^{-1} = W_{\mathcal{U}\mathcal{U}} - W_{\mathcal{U}\mathcal{V}} W_{\mathcal{V}\mathcal{V}}^{-1} W_{\mathcal{U}\mathcal{V}}^T\,.$$
- Let P be such that $\mathcal{I} - P$ and $(\mathcal{I} - P)^{-1} - \mathcal{I}$ are both invertible. Then P is invertible and
$$[(\mathcal{I} - P)^{-1} - \mathcal{I}]^{-1} = P^{-1} - \mathcal{I}\,. \tag{20}$$

Let us now compute H^* of (12): $\left(\begin{smallmatrix} H^* & 0 \\ 0 & 0 \end{smallmatrix}\right)$ is equal to

$$\nabla^2 f^*(G) = \nabla^2 F^*(G) - M^{-1} = \nabla^{-2} F(x_0) - M^{-1} = [(\mathcal{I} - \nabla p(x_0))^{-1} - \mathcal{I}]M^{-1}.$$

A straightforward computation gives

$$(\mathcal{I} - \nabla p(x_0))^{-1} = \begin{pmatrix} (\mathcal{I}_{\mathcal{U}} - P)^{-1} & (\mathcal{I}_{\mathcal{U}} - P)^{-1} T \\ 0 & \mathcal{I}_{\mathcal{V}} \end{pmatrix};$$

the $\mathcal{U}\mathcal{V}$-block of $\nabla^2 f^*(G)$ is therefore $[(\mathcal{I}_{\mathcal{U}} - P)^{-1} - \mathcal{I}_{\mathcal{U}}]W_{\mathcal{U}\mathcal{V}} + (\mathcal{I}_{\mathcal{U}} - P)^{-1} T W_{\mathcal{V}\mathcal{V}} = 0$. This serves to compute T and we obtain the $\mathcal{U}\mathcal{U}$-block

$$H^* = [(\mathcal{I}_{\mathcal{U}} - P)^{-1} - \mathcal{I}_{\mathcal{U}}](W_{\mathcal{U}\mathcal{U}} - W_{\mathcal{U}\mathcal{V}} W_{\mathcal{V}\mathcal{V}}^{-1} W_{\mathcal{U}\mathcal{V}}^T) = [(\mathcal{I}_{\mathcal{U}} - P)^{-1} - \mathcal{I}_{\mathcal{U}}]M_{\mathcal{U}\mathcal{U}}^{-1}\,.$$

This is precisely the inverse of $\mathrm{H}_{\mathcal{U}} f(p_0)$; then, (19) follows using (20).

3.3. Getting rid of strong convexity. Our final goal will be to eliminate the strong convexity assumption in the preceding second-order results. For this we perturb f to a strongly convex function f_τ, and we study the effect of this perturbation on the proximal point.

PROPOSITION 3.13. *Let f be a finite-valued convex function satisfying* (8) *at a given p_0. Take $z_0 \in \mathbb{R}^N$, $\tau \in]0, 1[$ and define $f_\tau := f + \frac{1}{2}\tau\|\cdot - z_0\|_M^2$. Consider the Moreau–Yosida regularization of f_τ associated to the metric defined by $(1 - \tau)M$:*

$$F_\tau(x) := \min_{y \in \mathbb{R}^n} \left\{ f_\tau(y) + \tfrac{1}{2}(1 - \tau)\|y - x\|_M^2 \right\}. \tag{21}$$

Denote by $q_\tau(x)$ the unique minimizer of (21); *then, the following statements hold:*

(i) *the function f_τ is strongly convex and satisfies* (8) *at p_0 with C replaced by $C + \tau\Lambda$;*

(ii) *for all x, $q_\tau(x) = p(\tau z_0 + (1 - \tau)x)$.*

Proof. The strong convexity of f_τ is clear. To prove that (8) holds for f_τ, add $\frac{1}{2}\tau\|p_0 + h - z_0\|_M^2$ to both sides of (8) written for f at p_0. Then use the properties $f'_\tau(p_0; h) = f'(p_0; h) + \frac{1}{2}\tau \langle M(p_0 - z_0), h\rangle$ and $\frac{1}{2}\tau\|\cdot\|_M^2 \le \frac{1}{2}\tau\Lambda\|\cdot\|^2$.

For proving (ii), write the optimality conditions for $p(\tau z_0 + 1 - \tau x)$ and $q_\tau(x)$:

$$\begin{array}{lll} p(\tau z_0 + (1 - \tau)x) \text{ solves} & M(\tau z_0 + (1 - \tau)x - p) \in \partial f(p) & \text{and} \\ q_\tau(x) \text{ solves} & (1 - \tau)M(x - p) \in \partial f_\tau(p)\,. & \end{array}$$

Since $\partial f_\tau(p) = \partial f(p) + \{\tau M(p - z_0)\}$, they have the same solutions. □

Thus, passing from f to f_τ can be absorbed by a perturbation of M in the Moreau–Yosida regularization of (2) and a (smooth) change of variables. Then the wording "strongly convex" can be removed in our second-order results. Note that our proof technique below will use f_τ and F_τ with arbitrary $\tau \in]0, 1[$.

THEOREM 3.14. *Let f be a finite-valued convex function such that for a given $x_0 \in \mathbb{R}^n$,* (8) *holds at $p_0 = p(x_0)$. Assume $\nabla f(p_0)$ exists. Then $\nabla^2 F(x_0)$ exists if and only if $\mathrm{H}f(p_0)$ exists.*

Proof. The "only if" part is Theorem 3.1. As for the "if" part, suppose $\nabla^2 F(x_0)$ exists; hence, from Proposition 2.4, $\nabla p(x_0)$ exists. Then, consider f_τ as in Proposition 3.13, with $z_0 = x_0$: we have $q_\tau(x) = p(\tau x_0 + (1-\tau)x)$ for all x, therefore $\nabla q_\tau(x_0) = (1-\tau)\nabla p(x_0)$ exists. Again, using Proposition 2.4, F_τ has a Hessian at x_0. Since f_τ is strongly convex and satisfies (8) at p_0, Proposition 3.4 applies: $\mathrm{H}f_\tau(p_0)$ exists. Thus $\mathrm{H}f(p_0) = \mathrm{H}f_\tau(p_0) - \tau M$ exists as well. □

THEOREM 3.15. *Let f be a finite-valued convex function such that for all $x_0 \in \mathbb{R}^N$,* (8) *holds at $p_0 = p(x_0)$. Then $\nabla^2 F$ exists on the whole of $\mathbb{R}^N$ if and only if $\nabla^2 f$ exists on the whole of $\mathbb{R}^N$.*

Proof. Recall that the existence of $\mathrm{H}f$ on the whole space implies the existence of $\nabla^2 f$ on the whole space. Then the "only if" part is Theorem 3.1. As for the "if" part, suppose $\nabla^2 F$ exists everywhere; hence, from Proposition 2.4, $p(\cdot)$ has a Jacobian everywhere. Then consider f_τ as in Proposition 3.13 with $z_0 = 0$; we have $\nabla q_\tau(x) = (1-\tau)\nabla p((1-\tau)x)$ for all x. Proceeding as in the proof of Theorem 3.14 but applying, this time, Proposition 3.5, we conclude that $\nabla^2 f_\tau$ (and hence $\nabla^2 f$) exists everywhere. □

For the nondifferentiable case, we again will use f_τ as in Proposition 3.13 with $z_0 = x_0$ and F_τ of (21). Then, because $G_\tau := \nabla F_\tau(x_0) = (1-\tau)G$, it follows that

$$\partial f_\tau(\cdot) - G_\tau = \partial f(\cdot) - G + \tau M(\cdot - p_0)\,. \tag{22}$$

We will also consider $\phi_{\mathcal{V},\tau}$, obtained by replacing f and G in (13) by f_τ and G_τ. The associated partial proximal operator $q_{\mathcal{V},\tau}$ is characterized by

$$\begin{gathered} \exists g \in \partial f(q_{\mathcal{V},\tau}(x)) \quad \text{such that} \\ q_{\mathcal{V},\tau}(x) - x = \operatorname{Proj}_{\mathcal{V}}(G - g) + \tau \operatorname{Proj}_{\mathcal{V}}(M(q_{\mathcal{V},\tau}(x) - p_0))\,. \end{gathered} \tag{23}$$

THEOREM 3.16. *Let f be a finite-valued convex function satisfying* (8) *at a given $p_0 = p(x_0)$. Assume $\nabla^2 F(x_0)$ exists. For $d = d_\mathcal{U} + d_\mathcal{V}$ tending to 0 in such a way that*

(i) $\|d_\mathcal{V}\| = o(\|d_\mathcal{U}\|)$,

(ii) $\exists g \in \partial f(p_0 + d)$ *such that* $\operatorname{Proj}_\mathcal{V}(g - G) = O(\|d\|)$,

we have the second-order development

$$f(p_0 + d) = f(p_0) + \langle G, d\rangle + \frac{1}{2}\langle d, H'd\rangle + o(\|d\|^2)\,, \tag{24}$$

where $H' = \begin{pmatrix} M_{\mathcal{U}\mathcal{U}}(P^{-1} - \mathcal{I}_\mathcal{U}) & -\tau M_{\mathcal{U}\mathcal{V}} \\ -\tau M_{\mathcal{U}\mathcal{V}}^T & \mathcal{I}_\mathcal{V} - \tau M_{\mathcal{V}\mathcal{V}} \end{pmatrix}$.

Proof. Consider again f_τ as in Proposition 3.13 with $z_0 = x_0$; F_τ of (21) has at x_0 a Hessian $\nabla^2 F_\tau(x_0) = (1-\tau)^2 \nabla^2 F(x_0) + \tau(1-\tau)M$. By Proposition 3.8, this is equivalent to the existence of $\mathrm{H}\phi_{\mathcal{V},\tau}(p_0)$. Take d satisfying (i) and apply (ii) together with (22): there is $g_\tau \in \partial f_\tau(p_0 + d)$ such that $\operatorname{Proj}_\mathcal{V}(g_\tau - G_\tau) = O(\|d\|) + \tau M d = O(\|d\|)$. Then Theorem 3.11 holds for the perturbed functions: mutatis mutandis, we write

$$f(p_0 + d) + \frac{1}{2}\tau\|p_0 + d - x_0\|_M^2 = f(p_0) + \frac{1}{2}\tau\|p_0 - x_0\|_M^2 + \langle G_\tau, d\rangle + \frac{1}{2}\langle d, H_\tau d\rangle + o(\|d\|^2),$$

where H_τ is the corresponding generalized Hessian. By rearranging terms we obtain

$$f(p_0 + d) = f(p_0) + \langle G, d\rangle + \frac{1}{2}\langle d, (H_\tau - \tau M)d\rangle + o(\|d\|^2)\,.$$

To finish the proof we give the expression of $H' := H_\tau - \tau M$. Indeed, by Lemma 3.7(iii), H_τ has the diagonal form $\left(\begin{smallmatrix} H_\tau^{*-1} & 0 \\ 0 & \mathcal{I}_\mathcal{V}\end{smallmatrix}\right)$. This, together with (19) and Proposition 3.13(ii), gives

$$H_\tau = \begin{pmatrix} \mathrm{H}_\mathcal{U} f_\tau(p_0) & 0 \\ 0 & \mathcal{I}_\mathcal{V} \end{pmatrix} = \begin{pmatrix} (1-\tau)M_{\mathcal{U}\mathcal{U}}(\frac{1}{1-\tau}P^{-1} - \mathcal{I}_\mathcal{U}) & 0 \\ 0 & \mathcal{I}_\mathcal{V} \end{pmatrix}.$$

Finally,

$$H' = H_\tau - \tau M = \begin{pmatrix} M_{\mathcal{U}\mathcal{U}}(P^{-1} - \mathcal{I}_\mathcal{U}) & -\tau M_{\mathcal{U}\mathcal{V}} \\ -\tau M_{\mathcal{U}\mathcal{V}}^T & \mathcal{I}_\mathcal{V} - \tau M_{\mathcal{V}\mathcal{V}} \end{pmatrix} \qquad \square$$

Of course, the $\mathcal{U}\mathcal{U}$-block of H' does not depend on τ: it *has to* be $\mathrm{H}_\mathcal{U} f(p_0)$ of (19).

COROLLARY 3.17. *Let f be a finite-valued convex function satisfying* (8) *at a given $p_0 = p(x_0)$. Assume $\nabla^2 F(x_0)$ exists. Let $h \to 0$ with $\|h_\mathcal{V}\| = O(\|h_\mathcal{U}\|)$ and set $x := p_0 + h$. For arbitrary $\tau \in]0, 1[$, $q_{\mathcal{V},\tau}(x)$ from* (23) *tends to p_0 and* (24) *holds for $d := q_{\mathcal{V},\tau}(x) - p_0 \to 0$.*

Proof. Proceed as before; $q_{\mathcal{V},\tau}(\cdot)$ of (23) enjoys the same properties as $p_\mathcal{V}(\cdot)$. Also, the existence of $\nabla^2 F(x_0)$ implies the existence of $\nabla^2 F_\tau(x_0)$, which in turn is equivalent to the existence of $\mathrm{H}\phi_{\mathcal{V},\tau}(p_0)$. Thus Corollary 3.12 applies. $\square$

3.4. The epi-convergence approach.

So far, our study has been dealing with rather classical derivatives: the Hessian for F and its (slight) generalization (6) for f. Another object appears as fairly handy when used in conjunction with Moreau–Yosida regularizations: the so-called epi-derivative. We proceed to explain in a few informal words what it is, referring to [26], [4], [24] for more detailed explanations.

- Let $\{E_t\}$ be a family of sets indexed by t. Form the set E of all possible clusterpoints of all possible sequences of elements $e_t \in E_t$ when $t \downarrow 0$. Under certain conditions which we do not specify here, we say that E is the *limit* of E_t and we write $E_t \to E$.
- Recall that the *epigraph* of a (convex) function φ is the set $\operatorname{epi}\varphi := \{(x, r) \in \mathbb{R}^n \times \mathbb{R} : r \geq \varphi(x)\} \subset \mathbb{R}^n \times \mathbb{R}$. The *graph* of its subdifferential is the set $\operatorname{gr}\partial\varphi := \{(z, g) : g \in \partial\varphi(z)\} \subset \mathbb{R}^n \times \mathbb{R}^n$.
- Several classical meanings can be given to a statement like "the function φ_t converges to the function φ" (pointwise convergence, uniform convergence,...). Here, we use the following concept: φ_t *epi-converges* to φ when $\operatorname{epi}\varphi_t \to \operatorname{epi}\varphi$ in the sense of the above set-convergence. We will then use the notation $\varphi = \operatorname{epi}\lim\varphi_t$.
- For closed convex functions, a fundamental property of epi-convergence is its stability under conjugacy and differentiation. More precisely, the statements $\varphi = \operatorname{epi}\lim\varphi_t$ and $\varphi^* = \operatorname{epi}\lim\varphi_t^*$ are equivalent. They are further equivalent to the statement $\operatorname{gr}\partial\varphi = \lim\operatorname{gr}\partial\varphi_t$, provided that pointwise convergence holds at least at one point (to fix the constant of integration).
- A (classical) second-order derivative can be viewed as a quadratic form. In the present theory, it is convenient to accept the value $+\infty$ for such an object. Accordingly, given a positive semidefinite operator H and a subspace S, we call *generalized quadratic form* characterized by H and S the closed convex function

$q(x) := 1/2 \langle x, Hx \rangle + \mathrm{I}_S$. Note that $\partial q(x) = Hx + S^\perp$ for $x \in S$. This class of functions is invariant under conjugacy: the conjugate of a generalized quadratic form is another generalized quadratic form, characterized by K and T, say.

• Let φ be a closed convex function. For given z_0 and $g_0 \in \partial\varphi(z_0)$, we form the second-order difference quotient

$$\Delta_t(d) := \frac{\varphi(z_0 + td) - \varphi(z_0) - t \langle g_0, d \rangle}{t^2} .$$

This is a closed convex function of d, indexed by $t \downarrow 0$. Direct calculations give its subdifferential

$$\partial\Delta_t(d) = \frac{\partial\varphi(z_0 + td) - g_0}{t}$$

and its conjugate

$$\Delta_t^*(s) = \frac{\varphi^*(g_0 + ts) - \varphi^*(g_0) - t \langle s, x_0 \rangle}{t^2}.$$

Observe that $\Delta_t(0) = \Delta_t^*(0) = 0$ and $\partial\Delta_t(0) \ni 0$.

• Then we say that φ has a second *epi-derivative* q at z_0, relative to g_0, when the function Δ_t epi-converges to q (a generalized quadratic form). Equivalently, φ^* has at g_0, relative to z_0, a second epi-derivative q^*. A further equivalence, probably the most useful, is $\operatorname{gr} \partial\Delta_t \to \operatorname{gr} \partial q$. In plain words, the clusterpoints of the difference quotients $[\partial\varphi(z_0 + td_t) - g_0]/t$, when $t \downarrow 0$ and $d_t \to d$, form some affine manifold $Hd + S$.

Now the stage is set and we can use these concepts in our Moreau–Yosida framework. Here again, we will not go into details, referring to [17] and also [12] for a more accurate analysis. Our aim here is to somehow "explain" our second-order results by heuristic observations rather than rigorous statements.

(a): We start with the following observation. Let the growth condition (8) hold at some p_0, and let f have a gradient at p_0. Then ∂f has at p_0 the radially Lipschitz behavior (see Corollary 3.5 in [11]); the two concepts of second epi-derivative and of generalized Hessian (7) coincide. Likewise, since ∇F is Lipschitzian, the two concepts of second epi-derivative and of classical Hessian coincide for F.

(b): Now suppose that f has at p_0 a second epi-derivative (H, S), relative to some subgradient $G \in \partial f(p_0)$. Equivalently, f^* has at G a second epi-derivative (K, T) relative to p_0. Clearly enough, $F^* = f^* + 1/2 \|\cdot\|^2_{M-1}$ also has a second epi-derivative $(K + M^{-1}, T)$ at G, which is relative to $p_0 + M^{-1}G \in \partial F^*(G)$. Dualizing again, F has (at $p_0 + M^{-1}G =: x_0$ and relative to $G = \nabla F(x_0)$) a second epi-derivative (H', S'). Naturally, since $K + M^{-1}$ is positive definite, this last epi-derivative is an ordinary quadratic function: $S' = \mathbb{R}^n$. Indeed, as already mentioned, the difference quotients $[\nabla F(x_0 + td) - G]/t$ (which are bounded) converge uniformly to a linear function $H'd$. This explains and actually completes Theorem 3.1.

(c): Conversely, suppose that F has a second epi-derivative at some x_0 (relative to $G = M(x_0 - p(x_0)) = \nabla F(x_0)$; of course, it is actually a Hessian). Then F^* has a second epi-derivative (K', T') at G, relative to x_0. Here again, $f^* = F^* - 1/2 \|\cdot\|^2_{M-1}$ has a second epi-derivative (K, T) at G, relative to $x_0 - M^{-1}G = p(x_0) \in \partial f^*(G)$. Finally, because f^* is convex, f itself has a second epi-derivative (H, S) at $p(x_0)$, relative to $G \in \partial f(p(x_0))$.

We conclude that as far as epi-derivatives are concerned, second differentiability of f and of F are always equivalent properties.

(c$'$): When F has a Hessian at x_0, suppose that the growth condition (8) holds and that $\nabla f(p(x_0))$ exists. As seen in (a) above, the second epi-derivative of f at $p(x_0)$ (which exists) is actually a generalized Hessian; this explains Proposition 3.4.

(d): Suppose f has a second epi-derivative at p_0, relative to $G \in \partial f(p_0)$ (for example, $\nabla^2 F(p_0 + M^{-1}G)$ exists), and consider curves $(g_t - G)/t$ with $g_t \in \partial f(p_0 + td_t)$ and $d_t \to d$. Their clusterpoints form the set $Hd + S^{\perp}$, where H is a symmetric positive semidefinite operator and S is a subspace.

• Such clusterpoints can exist only for $d \in \mathrm{N}_{\partial f(p_0)}(G)$ (otherwise, $g_t - G$ does not even tend to 0).

• To obtain these clusterpoints, one can in particular take $d_t \equiv 0$ and g_t arbitrary in $\partial f(p_0)$; this generates $\mathrm{T}_{\partial f(p_0)}(G)$, which is therefore contained in $S^{\perp}$.

• On the other hand, let the growth condition (8) hold. Then it takes some work (based on Corollary 3.3 in [11]) to realize that $S^{\perp}$ exactly reduces to $\mathrm{T}_{\partial f(p_0)}(G)$. As a result, $S^{\perp} = \mathrm{T}_{\partial f(p_0)}(G) = \mathcal{V}$. This explains Proposition 3.3; it also explains the extra term $p_{\mathcal{V}}(x) - x \in \mathcal{V} = S^{\perp}$ in (16).

• Finally, since these clusterpoints cover the whole of $Hd + S = Hd + \mathcal{U}$, some of them are exactly Hd; among the latter, we have those described by Theorem 3.11 and Corollary 3.12.

Let us summarize this section. Epi-derivatives are an elegant and powerful tool to relate second-order behaviors of f and F. They yield the essence of our results in sections 3.1–3.3 at practically no cost. This, however, is paid by the high degree of abstraction imposed by the concept. By contrast, our approach requires the heavy material developed in [11], but we deal with natural objects such as Taylor developments and ordinary (point-) convergence. We believe that the two approaches are in fact complementary and beneficial to each other: epi-derivatives give insightful guesses of the kind of result to be expected; standard convex analysis gives a more intuitive meaning to these "epi-results" and makes a closer description of $f(p_0 + h)$ for actual values of h. This last point becomes particularly useful when coming to numerical algorithms.

4. Concluding remarks. The very first motivation for the Moreau–Yosida regularization was to solve ill-conditioned systems of linear equations ([2], Chap.V). In fact, suppose f is quadratic, its Hessian H having extreme eigenvalues c and C. From Theorem 3.1, F is also quadratic with Hessian $M + M(H + M)^{-1}M$. Taking $M = \lambda\mathcal{I}$, a quick calculation shows that the condition number C/c of H is divided by $(\lambda + C)/(\lambda + c)$. This is a clear beneficial effect of the Moreau–Yosida regularization.

Consider now a general objective function. Barring all implementation considerations, assume that the proximal point $p(x)$ can be computed for each x (perhaps approximately, but for a negligible computation cost). Then the question arises whether such a computation is any good to minimize F (i.e., f). More specifically, what can be said about a superlinear algorithm minimizing F as compared to the ordinary proximal algorithm minimizing f?

When f is differentiable on the whole space, Theorem 3.15 and Corollary 3.2 tell us that such an approach brings exactly nothing. Either F still does not enjoy the necessary properties of smoothness and nondegeneracy or a superlinear algorithm could have been applied to f at the first place (ordinary Newton, quasi-Newton, or nonsmooth Newton as in [20]). For example, take an augmented Lagrangian

$$f(x) := f_0(x) + \frac{\pi}{2}\left[\max\left(0, f_1(x) + \frac{\mu}{\pi}\right)\right]^2,$$

which is associated to the nonlinear program: minimize f_0 subject to $f_1 \leq 0$. The minimization of the corresponding F (for given μ, π, and M) is not easier than the minimization of f: $\nabla^2 F$ exists and is positive definite only when $\nabla^2 f$ enjoys the same properties.

When f is differentiable just at an optimum point $\bar{x}$, the situation is less clear. On the one hand, existence of a positive definite $\nabla^2 F(\bar{x})$ gives hope for an efficient nonsmooth Newton algorithm; the pending question is semismoothness of ∇F, a question which is investigated in [19]. On the other hand, the existence alone of a (positive definite) generalized Hessian $\mathrm{H}f(\bar{x})$ is probably not quite enough to obtain superlinearly convergent algorithms applied directly to f. Here we mention a technical question. As far as quasi-Newton methods are concerned, an important property is the *strict* differentiability of ∇F at an optimum point (see [3], [8]). It would be interesting to examine the consequences of such a property on the behavior of f; following [24], some useful insight might be provided by the epi-derivative approach.

The real issue is when f is not differentiable at $\bar{x}$, a situation which does not preclude the existence of a positive definite $\nabla^2 F(\bar{x})$. In this case, any kind of Newtonian method will minimize F rapidly but will not even be locally convergent when applied to f. Existence of $\nabla^2 F$ at an optimum point $\bar{x}$ implies some interesting properties for f. First of all, $0 \in \operatorname{ri} \partial f(\bar{x})$, a property which can be compared to the strict complementarity slackness in constrained optimization. Furthermore, f enjoys a partial second-order behavior, via the existence of $\mathrm{H}_{\mathcal{U}} f(\bar{x})$; see (19). In our analogy with constrained optimization, $\mathcal{U}$ is the subspace tangent to the active constraints. The $\mathcal{UV}$-decomposition appears as an important tool from the theoretical point of view; this observation assesses the algorithmic approach of [14].

Take for illustration the bivariate function $f(x) := \max\{\frac{1}{2}\|x\|^2 - \alpha \langle e, x \rangle, \langle e, x \rangle\}$; here, $e := (0,1)^T$ and α is a nonnegative parameter.

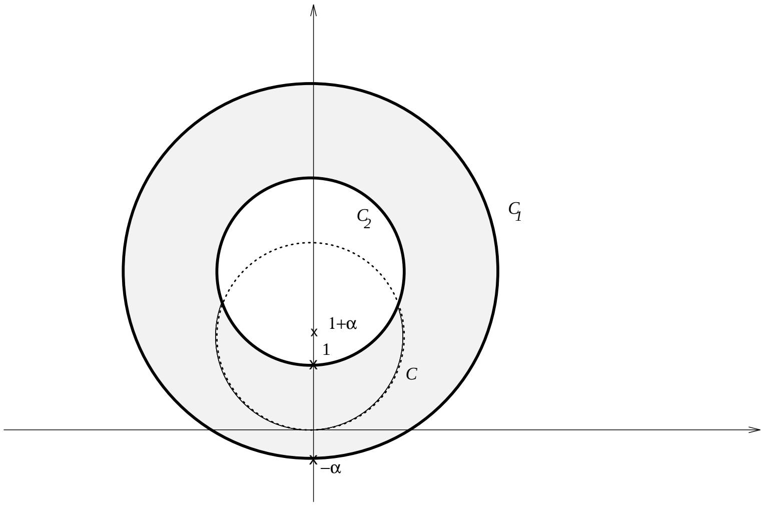


FIG. 1. *Moreau–Yosida regularization without Hessian.*

The kinks of f form a circle, denoted by C in Fig. 1. The subdifferential of f at 0 is the segment $[-\alpha e, e]$; hence, f is minimized at $0 = p(0)$ for all $\alpha \geq 0$. For $M = \mathcal{I}$, let us compute the proximal point of $x \neq 0$:

$$x - p = \begin{cases} p - \alpha e & \text{if } \frac{1}{2}\|p\|^2 - \alpha \langle e, p\rangle > \langle e, p\rangle\,, \\ \mu(p - \alpha e) + (1-\mu)e & \text{for some } \mu \in [0,1] \text{ if } \frac{1}{2}\|p\|^2 - \alpha \langle e, p\rangle = \langle e, p\rangle\,, \\ e & \text{if } \frac{1}{2}\|p\|^2 - \alpha \langle e, p\rangle < \langle e, p\rangle\;. \end{cases}$$

Working out the calculations, we find that

$$p(x) = \begin{cases} \dfrac{x + \alpha e}{2} & \text{if } \|x - (\alpha+2)e\| > 2(\alpha+1), \\ \dfrac{x + (\alpha\mu + \mu - 1)e}{1+\mu} & \text{if } \alpha + 1 \leq \|x - (\alpha+2)e\| \leq 2(\alpha+1), \\ x - e & \text{if } \|x - (\alpha+2)e\| < \alpha + 1\,, \end{cases}$$

where

$$\mu = \mu(x) := \frac{\|x - (\alpha+2)e\|}{\alpha+1} - 1\,. \tag{25}$$

In a condensed form,

$$p(x) = \frac{x - e + (\alpha+1)\nu(x)e}{1 + \nu(x)}\,, \tag{26}$$

where $\nu(x)$ is the projection of $\mu(x)$ in (25) onto $[0,1]$.

In Fig. 1, C_1 (respectively, C_2) is the boundary of the region where the first (respectively, second) function prevails at $p(x)$. The dashed crown is the locus of those x such that $p(x)$ is a kink. The point is that C_2 is always far from 0, while C_1 does contain 0 when $\alpha = 0$. As a result, $\nabla^2 F(0)$ exists if $\alpha > 0$ but not if $\alpha = 0$. To show this, we consider two cases.

(i): When $\alpha = 0$, the origin is on C_1. Analytically, $\nu(x) = 1$ in (26) whenever $\|x - 2e\| > 2$. From this observation, the directional derivatives are easy to compute:

$$p'(0; d) = \begin{cases} \frac{1}{2}(d_1, 0)^T & \text{for } d_2 > 0, \\ \frac{1}{2}(d_1, d_2)^T & \text{for } d_2 \leq 0\,. \end{cases}$$

Here, the nonexistence of $\nabla p(0)$ illustrates Proposition 3.3: $G = 0$ is on the relative boundary of $\partial f(0) = [0,1]$.

(ii): When $\alpha > 0$, we have $\mu \in [0,1]$ in (25) for small $\|x\|$. This comes from $\mu(0) = 1/(\alpha+1)$, together with the continuity of $\mu(\cdot)$. In this region, which includes the origin in its interior,

$$p(x) = (\alpha+1)\frac{x - (\alpha+2)e}{\|x - (\alpha+2)e\|} + (\alpha+1)e\,.$$

A mere differentiation gives

$$\nabla p(0) = \frac{\alpha+1}{\alpha+2}\left(\mathcal{I} - ee^T\right) = \frac{\alpha+1}{\alpha+2}\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

Let us turn now to the $\mathcal{UV}$-analysis developed in section 3.2. Here $\mathcal{V} = 0 \times \mathbb{R}$ is the vertical axis. The partial (generalized) Hessian (19) is $\mathrm{H}_{\mathcal{U}} f(0) = (\frac{\alpha+2}{\alpha+1} - 1)\mathcal{I}_{\mathcal{U}} = \frac{1}{\alpha+1}\mathcal{I}_{\mathcal{U}}$.

We claim that for small $\|x\|$, the partial proximal operator is

$$p_{\mathcal{V}}(x) = (x_1, \alpha + 1 - \sqrt{D})^T \,,$$

where we have set $D := (\alpha+1)^2 - x_1^2$. Note: $\sqrt{D} = \alpha+1-\frac{x_1^2}{2(\alpha+1)}+o(x_1^2)$ and $[p_{\mathcal{V}}(x)]_2 = \frac{x_1^2}{2(\alpha+1)} + o(x_1^2)$. Geometrically, $p_{\mathcal{V}}(x)$ is obtained by intersecting $x + \mathcal{V}$ with C. To prove the formula analytically, plug $\partial f(p_{\mathcal{V}}(x)) = \{\lambda p_{\mathcal{V}}(x) + (1-(\alpha+1)\lambda)e : \lambda \in [0,1]\}$ and $G = 0$ in the characterization (14) to obtain

$$[p_{\mathcal{V}}(x) =]p(\lambda) := \left(x_1, \frac{x_2 - 1 + \lambda(\alpha+1)}{1+\lambda} \right)^T \tag{27}$$

for some $\lambda \in [0,1]$. With this change of variables, $p_{\mathcal{V}}(x)$ can be rewritten as

$$p_{\mathcal{V}}(x) = \underset{\lambda}{\operatorname{argmin}} \{ f(p(\lambda)) + \tfrac{1}{2}\|x - p(\lambda)\|^2 \} \,.$$

It takes some calculations to see that this minimum is attained at

$$\lambda := \frac{\alpha + 2 - \sqrt{D} - x_2}{\sqrt{D}} = \frac{1 - x_2}{\alpha + 1} + o(x_1^2) \to \frac{1}{\alpha+1} \in]0,1[\,.$$

With this value of λ in (27), the claim follows.

This confirms that $p_{\mathcal{V}}(x)$ is a kink, as explained at the end of [11]. Then the function $\phi_{\mathcal{V}}$ of (13) has the expression

$$\phi_{\mathcal{V}}(x) = [p_{\mathcal{V}}(x)]_2 + \frac{1}{2}\|[p_{\mathcal{V}}(x)]_2 - x_2\|^2 = \frac{x_1^2}{2(\alpha+1)} + \frac{1}{2}x_2^2 + o(\|x\|^2) \,.$$

It can be differentiated directly, or (15) can be used: $\partial\phi_{\mathcal{V}}(x)$ is made up of those $g \in \partial f(p_{\mathcal{V}}(x))$ whose second coordinate is the same as $x - p_{\mathcal{V}}(x)$. We obtain just one vector: $\lambda p_{\mathcal{V}}(x) + (1 - (\alpha+1)\lambda)e$, where λ takes the above value. Thus,

$$\nabla\phi_{\mathcal{V}}(x) = \begin{pmatrix} \frac{\alpha+2-\sqrt{D}x_2}{\sqrt{D}} x_1 \\ x_2 - [p_{\mathcal{V}}(x)]_2 \end{pmatrix} = \begin{pmatrix} \frac{x_1}{\alpha+1} \\ x_2 \end{pmatrix} + o(\|x\|) \,.$$

From there, (generalized) Hessians follow easily. Alternatively, the above expression of $\nabla p(0)$ can be plugged into the calculations made at the end of section 3.2: $P = \frac{\alpha+1}{\alpha+2}\mathcal{I}_{\mathcal{U}}$, $T = 0$, and we obtain

$$\mathrm{H}^* = (\mathcal{I}_{\mathcal{U}} - P)^{-1} - \mathcal{I}_{\mathcal{U}} = (\alpha+1)\mathcal{I}_{\mathcal{U}}, \quad \nabla^2\phi_{\mathcal{V}}(0) = \begin{pmatrix} \frac{1}{\alpha+1} & 0 \\ 0 & 1 \end{pmatrix}$$

and

$$\mathrm{H}_{\mathcal{U}} f(0) = P^{-1} - \mathcal{I}_{\mathcal{U}} = \left(\frac{\alpha+2}{\alpha+1} - 1 \right) \mathcal{I}_{\mathcal{U}} = \frac{1}{\alpha+1}\mathcal{I}_{\mathcal{U}} \,.$$

We have a final comment. In this paper we focused our attention on the Fréchet point of view; as far as algorithms are concerned, this is well suited to the quasi-Newton pattern. For the Newton pattern (possibly approximate, see [18]), the directional point of view may be more relevant; see [20]. Likewise, the Moreau–Yosida regularization could be generalized to the resolvent of a maximal monotone operator in the framework of variational inequalities; for this, see [16].

We are indebted to J.-B. Hiriart-Urruty, L.Q. Qi, and R.T. Rockafellar, as well as three anonymous referees, for careful reading and many helpful suggestions.

REFERENCES

[1] A. Auslender, *Numerical methods for nondifferentiable convex optimization*, Math. Programming Study, 30 (1987), pp. 102–126.

[2] R. Bellman, R. Kalaba, and J. Lockett, *Numerical Inversion of the Laplace Transform*, Elsevier, New York, 1966.

[3] J. Bonnans, J. Gilbert, C. Lemaréchal, and C. Sagastizábal, *A family of variable metric proximal methods*, Math. Programming, 68 (1995), pp. 15–48.

[4] S. Dolecki, G. Salinetti, and R. Wets, *Convergence of functions: Equi-semi-continuity*, Trans. Amer. Math. Soc., 276 (1983), pp. 409–429.

[5] M. Fukushima, *A descent algorithm for nonsmooth convex programming*, Math. Programming, 30 (1984), pp. 163–175.

[6] J.-B. Hiriart-Urruty, *The approximate first-order and second-order directional derivatives for a convex function*, in Mathematical Theories of Optimization, J.-P. Cecconi and T. Zolezzi, eds., Lecture Notes in Mathematics 979, Springer-Verlag, Berlin, New York, 1983, pp. 154–166.

[7] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, New York, 1993.

[8] C.-M. Ip and J. Kyparisis, *Local convergence of quasi-Newton methods for B-differentiable equations*, Math. Programming, 56 (1992), pp. 71–89.

[9] K. Kiwiel, *Proximity control in bundle methods for convex nondifferentiable minimization*, Math. Programming, 46 (1990), pp. 105–122.

[10] C. Lemaréchal and C. Sagastizábal, *An approach to variable metric bundle methods*, in Systems Modelling and Optimization, J. Henry and J.-P. Yvon, eds., Lecture Notes in Control and Inform. Sci. 197, Springer-Verlag, Berlin, New York, 1994, pp. 144–162.

[11] C. Lemaréchal and C. Sagastizábal, *More than first-order developments of convex functions: Primal-dual relations*, J. Convex Anal., (1996), pp. 1–14.

[12] Y. Lucet, *Formule explicite du hessien de la régularisée de Moreau-Yosida d'une fonction convexe f en fonction de l'épi-différentielle seconde de f*, Report # LAO95-08, University of Toulouse, France, 1995.

[13] B. Martinet, *Régularisation d'inéquations variationnelles par approximations successives*, Revue Française d'Informatique et Recherche Opérationnelle, R-3 (1970), pp. 154–179.

[14] R. Mifflin, *A Quasi-Second-Order Proximal Bundle Algorithm*, Technical report 93-3, University of Washington, Pullman, WA, 1993.

[15] J. Moreau, *Proximité et dualité dans un espace hilbertien*, Bull. Soc. Math. France, 93 (1965), pp. 273–299.

[16] J. Pang and L. Qi, *Nonsmooth equations: Motivation and algorithms*, SIAM J. Control Optim., 3 (1993), pp. 443–465.

[17] R. Poliquin and R. Rockafellar, *Generalized hessian properties of regularized nonsmooth functions*, SIAM J. Optim., 6 (1996), pp. 1121–1137.

[18] L. Qi, *Superlinearly convergent approximate Newton methods for LC^1 optimization problems*, Math. Programming, 64 (1994), pp. 277–294.

[19] L. Qi, *Second-Order Analysis of the Moreau-Yosida Approximation of a Convex Function*, Applied Mathematics Report AMR94/20, School of Mathematics, The University of New South Wales, Sydney, Australia, 1994.

[20] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Programming, 58 (1993), pp. 353–367.

[21] M. Qian, *The Variable Metric Proximal Point Algorithm: Application to Optimization*, Department of Mathematics, Manuscript GN-50, University of Washington, Seattle, WA 98195, 1992.

[22] R. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[23] R. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.

[24] R. Rockafellar, *Maximal monotone relations and the second derivatives of nonsmooth functions*, Annales de l'Institut Henri Poincaré, Analyse non linéaire, 2 (1985), pp. 167–186.

[25] 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.

[26] R. Wets, *Convergence of convex functions, variational inequalities and convex optimization problems*, in Variational Inequalities and Complementarity Problems, F. G. R. Cottle and J.-L. Lions, eds., Wiley, New York, 1980, pp. 405–419.

[27] K. Yosida, *Functional Analysis*, Springer-Verlag, Berlin, New York, 1964.

[28] C. Lemaréchal and C. Sagastizábal, *Variable metric bundle methods: From conceptual to implementable forms*, Math. Programming, to appear.

SIAM J. OPTIM.
Vol. 7, No. 2, pp. 386–402, May 1997

AN INFEASIBLE PATH-FOLLOWING METHOD FOR MONOTONE COMPLEMENTARITY PROBLEMS*

PAUL TSENG†

Abstract. We propose an infeasible path-following method for solving the monotone complementarity problem. This method maintains positivity of the iterates and uses two Newton steps per iteration—one with a centering term for global convergence and one without the centering term for local superlinear convergence. We show that every cluster point of the iterates is a solution, and if the underlying function is affine or is sufficiently smooth and a uniform nondegenerate function on $\Re^n_{++}$, then the convergence is globally Q-linear. Moreover, if every solution is strongly nondegenerate, the method has local quadratic convergence. The iterates are guaranteed to be bounded when either a Slater-type feasible solution exists or when the underlying function is an R_0-function.

Key words. monotone complementarity problem, infeasible path-following method, global Q-linear convergence, local quadratic convergence

AMS subject classifications. 49M45, 90C25, 90C33

PII. S105262349427409X

1. Introduction.

The complementarity problem (CP) is the problem of finding an $(x^*, y^*) \in \Re^{2n}$ satisfying

$$x^* \geq 0, \quad y^* \geq 0, \quad (x^*)^T y^* = 0, \quad F(x^*) - y^* = 0, \tag{1}$$

where $F = (F_1, ..., F_n)$ is a given continuous function from $\Re^n_+$ to $\Re^n$. This problem is well known in optimization, as is surveyed in [1, 21], and in the case where F is affine, it reduces to the linear complementarity problem [2].

Many solution approaches for CP have been proposed. One approach is based on modifying the projection method [12, 23]. A second approach is based on reformulating the CP as a differentiable minimization problem with simple constraints and then applying a descent method to the latter [4, 16, 19]. A third approach is based on reformulating the CP as a system of smooth nonlinear equations and applying a Newton-type method to solve the system [3, 7, 15, 24, 28]. An alternative to this approach is to reformulate the CP as a system of nonsmooth nonlinear equations and then use solution techniques from nonsmooth analysis (see [6, 20] and references therein). A fourth approach is based on approximating the CP by a parameterized system of smooth nonlinear equations and, after solving the system inexactly using a few Newton steps, adjusting the parameter to refine the approximation. The path-following interior-point methods (see [8, 9, 11, 25, 29]) are essentially based on this approach, whereby the approximating system, parameterized by an $\omega \in [0, \infty)$, is of the form $H(x, y) = (\omega e, 0)$ with $H : \Re^{2n} \to \Re^{2n}$, the function given by (see [13])

$$H(x, y) = (Xy, F(x) - y). \tag{2}$$

Here e denotes the vector of 1's and ωe is a centering term that keeps (x, y) away from the boundary of $\Re^n_+$. Related approaches using potential reduction are presented in [18, 22, 27].

* Received by the editors September 12, 1994; accepted for publication (in revised form) December 11, 1995. The author was supported by NSF grant CCR-9311621.
http://www.siam.org/journals/siopt/7-2/27409.html

† Department of Mathematics, Box 354350, University of Washington, Seattle, WA 98195 (tseng@math.washington.edu).

Motivated by the recent interests in infeasible interior-point methods, in this paper we study a new infeasible interior-point method for solving CPs when F is monotone and continuously differentiable on $\Re^n_{++}$. (Our results appear also to extend to the case where $F'(x)$ is a $P^*(\kappa)$-matrix in the sense of [10] for all $x \in \Re^n_{++}$, with κ a nonnegative constant, but for simplicity we will not consider this more general case.) The method maintains at each iteration an (x, y, ω) in a neighborhood around the central path given by

$$\mathcal{N}_{\rho,\beta} = \{\ (x, y, \omega) \in \Re^{2n+1}_{++} \mid \|H(x, y) - (\omega e, 0)\|_\rho \le \omega\beta\ \}$$

for some $\beta \in (0, 1)$, where for a fixed $\rho \in (0, \infty)$, we define the weighted norm $\|(u, v)\|_\rho = \sqrt{\|u\|^2 + \rho^2\|v\|^2}$. (The parameter ρ determines the magnitude of the infeasibility term $\|F(x) - y\|$ relative to the centering term $\|\omega e - Xy\|$.) The method updates (x, y, ω) by computing simultaneously the Newton direction (u, v) satisfying

$$H'(x, y) \begin{bmatrix} u \\ v \end{bmatrix} + H(x, y) = \begin{bmatrix} \omega e \\ (1 - \gamma)(F(x) - y) \end{bmatrix} \tag{3}$$

with $\gamma \in (0, 1]$ suitably chosen and the Newton direction $(\hat{u}, \hat{v})$ satisfying

$$H'(x, y) \begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix} + H(x, y) = 0 \tag{4}$$

and then moving (x, y) along either (u, v) or a convex combination of (u, v) and $(\hat{u}, \hat{v})$, with the aim of decreasing ω while maintaining $(x, y, \omega) \in \mathcal{N}_{\rho,\beta}$.

The proposed method is an infeasible path-following method in the spirit of [11, 29] in that it maintains (x, y) as positive and as an inexact solution of $H(x, y) = (\omega e, 0)$, while it decreases ω towards 0. However, this method differs from the methods of [11, 29] in at least two significant ways: (i) it, like the method of [5], uses the two Newton directions in combination rather than in alternation; (ii) it moves y along a straight line, rather than along an arc (so it does not maintain $F(x) - y$ to be a scalar multiple of its initial value). The method is relatively simple and has nice global and local convergence properties (see Theorem 4.2 and Lemma 4.3). In particular, every cluster point of the iterates generated by this method is a solution of the CP and, if in addition F is affine or is sufficiently smooth and a uniform nondegenerate function on $\Re^n_{++}$, the convergence is globally Q-linear. (To our knowledge, this is the first global linear convergence result for an infeasible interior-point method when F is not affine.) If every solution of CP is strongly nondegenerate, then the convergence is locally quadratic. (This result is analogous to one in [29].) Finally, the iterates are guaranteed to be bounded if either there exists an $\bar{x} \in \Re^n_+$ with $F(\bar{x})$ sufficiently positive or F is an R_0-function.

We assume throughout that F is monotone and continuously differentiable on $\Re^n_{++}$. This implies that the Jacobian $F'(x) = [F_1'(x) \ \cdots \ F_n'(x)]^T$ is positive semidefinite (not necessarily symmetric) for all $x \in \Re^n_{++}$. For parts of the global convergence rate analysis, we will further assume that F is a *uniform nondegenerate function* on $\Re^n_{++}$ in the sense that there exists a continuous function $\eta : \Re^n_+ \mapsto (0, \infty)$ such that

$$\|F'(x)^{-1}_{II}\| \le \eta(x) \quad \forall I \subset \{1, ..., n\}\ \forall x \in \Re^n_{++}. \tag{5}$$

(This assumption is satisfied when F is a uniform P-function [6, 19, 27] on $\Re^n_{++}$, as shown in Lemma 4.6, and, in particular, when $F(x) = Mx + q$ for all x with M being an $n \times n$ positive semidefinite nondegenerate matrix [2].) We denote by S the set of solutions (possibly empty) of the CP, i.e.,

$$S = \{(x^*, y^*) \in \Re^{2n} \mid (x^*, y^*) \text{ satisfies } (1)\}.$$

We say that an $(x^*, y^*) \in S$ is *strongly nondegenerate* if $(x_I^*, y_J^*) > 0$ and $F'(x^*)_{II}$ is nonsingular for some partition I, J of $\{1, \ldots, n\}$ (i.e., $I \cup J \in \{1, \ldots, n\}$ and $I \cap J = \emptyset$) (cf. [9, Condition 7.1], [29, Assumption 2]). We also denote the remainder term

$$r(x, z) = \|F(x+z) - F(x) - F'(x)z\| \quad \forall (x, z) \in \Re_+^n \times \Re^n \text{ with } F(x+z) \text{ defined.}$$

(Notice that $r(x, z)/\|z\| \to 0$ as $\|z\| \to 0$.) For the global convergence rate analysis, we will assume that there is a continuous function $L : \Re_{++}^n \mapsto [0, \infty)$ such that

$$r(x, z) \le L(x)\|z\|^2 \quad \forall (x, z) \in \Re_{++}^n \times \Re^n \text{ with } \|X^{-1}z\| \le 1. \tag{6}$$

For the local convergence rate analysis, we will assume that for a given $x^* \in \Re_+^n$, there exist scalars $\mu > 0$ and $\epsilon > 0$ such that

$$r(x, z) \le \mu\|z\|^2 \quad \forall (x, z) \in \Re_+^n \times \Re^n \text{ with } \max\{\|z\|, \|x - x^*\|\} \le \epsilon. \tag{7}$$

Both assumptions are quite mild and hold whenever F is defined and twice continuously differentiable on an open set containing $\Re_+^n$. (This can be seen by letting $L(x) := \sum_{i=1}^n \max_z\{\|F_i''(x+z)\| \mid \|X^{-1}z\| \le 1\}$ and by letting $\mu := \sum_{i=1}^n \max_z\{\|F_i''(x+z)\| \mid \max\{\|z\|, \|x - x^*\|\} \le \epsilon/2\}$ and ϵ be any positive scalar such that the closed Euclidean ball of radius ϵ around x^* is contained in the aforementioned open set.) The first assumption also holds when F' is Lipschitz continuous on $\Re_{++}^n$ (in which case L can be chosen to be a constant function).

In our notation, all vectors are column vectors and superscript T denotes transpose. We denote by $\Re^m$ the m-dimensional real vector space and by $\Re_+^m$ and $\Re_{++}^m$, respectively, the nonnegative orthant and the strictly positive orthant in $\Re^m$. We denote by e the vector whose components are all 1 (with its dimension inferred from the context). For any $x \in \Re^m$, we denote by x_i the ith component of x, by X the $m \times m$ diagonal matrix whose ith diagonal entry is x_i for all i, and by $\|x\|_1$, $\|x\|$, $\|x\|_\infty$ the 1-norm, the 2-norm, and the ∞-norm, respectively, of x. For any $I \subset \{1, \ldots, m\}$, we denote by x_I the vector with components x_i, $i \in I$. For any $m \times m$ real matrix M, we denote $\|M\| = \max_{\|x\|=1} \|Mx\|$. For any $I, J \subset \{1, \ldots, m\}$, we denote by M_I the submatrix of M obtained by removing all rows not indexed by I and by M_{IJ} the submatrix of M_I obtained by removing all columns not indexed by J. Finally, we will frequently use in our analysis the following observation:

$$\|\omega e - Xy\| \le \omega\beta_1, \qquad \rho\|F(x) - y\| \le \omega\beta_1 \tag{8}$$

for any $(x, y, \omega) \in \mathcal{N}_{\rho,\beta_1}$ and any $\beta_1 \in (0, 1)$.

2. Centering Newton step and its properties. We describe below the Newton step based on (3). Fix any scalars β_1, β_2 satisfying $0 < \beta_1/(1 - \beta_1) < \beta_2 < 1$ and any $\rho \in (0, \infty)$ and $\psi \in (0, 1)$. For any $(x, y) \in \Re_{++}^{2n}$, let $\Omega(x, y)$ denote the smallest $\omega \in [0, \infty)$ satisfying $(x, y, \omega) \in \mathcal{N}_{\rho,\beta_1}$, with $\Omega(x, y) := \infty$ if no such ω exists.

Centering Newton step. *For a given $(x, y, \omega) \in \mathcal{N}_{\rho,\beta_1}$, we choose the largest $\gamma \in [0, 1]$ satisfying*

$$\|\omega e - Xy - \gamma X(F(x) - y)\| \le \beta_2(\omega - \|\omega e - Xy\|) \tag{9}$$

and let (u,v) *be the vector in* $\Re^{2n}$ *satisfying* (3); *that is,*

$$\begin{bmatrix} Y & X \\ F'(x) & -I \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \omega e - Xy \\ \gamma(y - F(x)) \end{bmatrix}; \tag{10}$$

then we choose the largest $\lambda \in \{1, \psi, \psi^2, \ldots\}$ *satisfying*

$$(x+\lambda u, y+\lambda v) > 0, \quad \Omega(x+\lambda u, y+\lambda v) \le \omega(\beta_1\sqrt{1-\lambda\gamma} + \sqrt{n})/(\beta_1+\sqrt{n}), \tag{11}$$

and let

$$(x_+, y_+) := (x+\lambda u, y+\lambda v), \qquad \omega_+ := \Omega(x_+, y_+). \tag{12}$$

Roughly speaking, γ is chosen to ensure a balanced decrease in the centering term $\|\omega e - Xy\|$ and the infeasibility term $\|F(x) - y\|$; λ is chosen to minimize (approximately) the minimum ω for which $(x+\lambda u, y+\lambda v, \omega) \in \mathcal{N}_{\rho,\beta_1}$. Note that both γ and, for each λ, $\Omega(x+\lambda u, y+\lambda v)$ can be computed very easily by solving a quadratic equation in one variable. Also, we can more generally choose λ from $\{\tau, \tau\psi, \tau\psi^2, \ldots\}$ for some $\tau \in (0,\infty)$, but, for simplicity, we will not consider this more general stepsize rule.

The following key lemma shows that γ is bounded away from 0 and that λ is well defined and not too small. (γ is well defined since, by $(x,y,\omega) \in \mathcal{N}_{\rho,\beta_1}$ (so the first inequality in (8) holds) and $\beta_1/(1-\beta_1) < \beta_2$, (9) is satisfied by $\gamma = 0$, so there is a largest $\gamma \in [0,1]$ satisfying (9).) Moreover, in cases where F is affine or F is sufficiently smooth and a uniform nondegenerate function on $\Re^n_{++}$, a certain quantity $\sigma(\lambda)$ is bounded above by a positive continuous function of x. Alternatively, if (x,y) is near a strongly nondegenerate solution and ω is near 0, the quantity $\sigma(\lambda)$ is bounded above by a constant and (cf. [29, Lemma 5.2]) (u,v) is in the order of ω.

LEMMA 2.1. *Fix any scalars* β_1, β_2 *with* $0 < \beta_1/(1-\beta_1) < \beta_2 < 1$ *and any* $\rho \in (0,\infty)$. *For any* $(x,y,\omega) \in \mathcal{N}_{\rho,\beta_1}$, *if* γ *denotes the largest scalar in* $[0,1]$ *satisfying* (9) *and* (u,v) *denotes the vector satisfying* (10), *then the following hold:*

(a) $\|X^{-1}u\| < 1$ *and*

$$\gamma \ge \min\{1, (\beta_2(1-\beta_1) - \beta_1)\rho/(\|x\|_\infty\beta_1)\}. \tag{13}$$

(b) *Any* $\lambda \in [0,\bar\lambda]$ *satisfies* (11), *where* $\bar\lambda$ *denotes the smallest* $\lambda \in [0,1]$ *satisfying* $\lambda\sigma(\lambda) = \gamma$, *with*

$$\sigma(\lambda) := (1+\beta_3/\beta_1)^2 + 2\rho r(x,\lambda u)/(\lambda^2\beta_1\omega) + \rho^2 r(x,\lambda u)^2/(\lambda\beta_1\omega)^2 \tag{14}$$

and $\beta_3 = (\beta_1 + \beta_2(1-\beta_1))\beta_2$. *For any* $\lambda \in [0,1]$ *satisfying* (11), (x_+, y_+, ω_+) *given by* (12) *is in* $\mathcal{N}_{\rho,\beta_1}$.

(c) *If there exist continuous functions* $\eta : \Re^n_+ \mapsto (0,\infty)$ *and* $L : \Re^n_{++} \mapsto [0,\infty)$ *such that* (5) *and* (6) *hold, respectively, then for all* $\lambda \in [0,1]$, $\sigma(\lambda)$ *given by* (14) *is bounded above by a continuous function of* x *depending on* β_1, β_2, ρ, F', η, L *only.*

(d) *If* F *is affine, then for all* $\lambda \in [0,1]$, $\sigma(\lambda)$ *given by* (14) *is bounded above by a constant depending on* β_1 *and* β_2 *only.*

(e) *If there exists an* $(x^*, y^*) \in S$ *that is strongly nondegenerate and for which there exist scalars* $\mu > 0$ *and* $\epsilon > 0$ *satisfying* (7), *then there exist positive constants* δ *and* C *(depending on* β_1, β_2, ρ, $F', (x^*, y^*)$, μ, ϵ*) such that* $\|(x,y) - (x^*,y^*)\| \le \delta$ *implies that* $\|(u,v)\| \le C\omega$ *and if, in addition,* $\omega \le \epsilon/C$, *implies that for all* $\lambda \in [0,1]$, $\sigma(\lambda)$ *given by* (14) *is bounded above by* C.

Proof. Let $M = F'(x)$, $r = \omega e - Xy$, $s = F(x) - y$, $d = X^{-1}u$, and $\beta = \|r\|/\omega$. Then (10) may be rewritten as

$$YXd + Xv = r, \qquad MXd - v = -\gamma s.$$

Since $XMX + YX$ is positive definite and hence invertible, we can solve for d to obtain

$$d = (XMX + YX)^{-1}(r - \gamma Xs).$$

This together with the positive semidefinite property of M yields

$$d^TYXd \le d^T(XMX + YX)d = d^T(r - \gamma Xs) \le \|d\|\|r - \gamma Xs\|, \tag{15}$$

and it readily follows that

$$\|d\| \le \|r - \gamma Xs\|/\min_i x_iy_i \le \|r - \gamma Xs\|/\omega(1 - \beta), \tag{16}$$

where the second inequality follows from $\|r\| = \beta\omega$ so $Xy \ge (1 - \beta)\omega e$.

(a) By (9), the right-hand side of (16) is below β_2, so

$$\|X^{-1}u\| = \|d\| \le \beta_2 < 1. \tag{17}$$

We have either $\gamma = 1$ or (9) is satisfied with equality. In the latter case, we have

$$\begin{aligned} \beta_2(1 - \beta) &= \|\omega e - Xy - \gamma Xs\|/\omega \\ &\le \|\omega e - Xy\|/\omega + \gamma\|x\|_\infty\|s\|/\omega \\ &= \beta + \gamma\|x\|_\infty\|F(x) - y\|/\omega \\ &\le \beta + \gamma\|x\|_\infty\beta_1/\rho, \end{aligned}$$

where the last inequality uses the second equation in (8). This together with $\beta \le \beta_1$ yields

$$\gamma \ge (\beta_2(1 - \beta_1) - \beta_1)\rho/(\|x\|_\infty\beta_1).$$

Thus, (13) holds.

(b) Fix any $\lambda \in [0, \bar{\lambda}]$, and we show below that (11) holds. (Note that $\bar{\lambda}$ is well defined and positive. This is because $r(x, \lambda u)/\lambda \to 0$ as $\lambda \to 0$, so $\lambda\sigma(\lambda) \to 0$ as $\lambda \to 0$ while, by (14), $\lambda\sigma(\lambda)$ exceeds 1 when $\lambda \ge 1$.) Let $(x', y') := (x + \lambda u, y + \lambda v)$. Now, $u = Xd$ implies $X' = X + \lambda DX$, which together with $y' = y + \lambda v$ yields

$$\begin{aligned} \omega e - X'y' &= \omega e - (I + \lambda D)X(y + \lambda v) \\ &= (1 - \lambda)(\omega e - Xy) - \lambda^2 D(\omega e - Xy) + \lambda^2 D(\omega e - Xy - Xv) \\ &= (1 - \lambda)(\omega e - Xy) - \lambda^2 D(\omega e - Xy) + \lambda^2 DYXd, \end{aligned}$$

where both the second and the third equality follow from the first equation in (10). Thus

$$\begin{aligned} \|\omega e - X'y'\| &\le (1 - \lambda)\|\omega e - Xy\| + \lambda^2\|D(\omega e - Xy)\| + \lambda^2\|DYXd\| \\ &\le (1 - \lambda)\|\omega e - Xy\| + \lambda^2\|d\|\|\omega e - Xy\| + \lambda^2\|DYXd\|_1 \\ &= (1 - \lambda)\|r\| + \lambda^2\|d\|\|r\| + \lambda^2 d^TYXd \\ &\le (1 - \lambda)\|r\| + \lambda^2(\|r\| + \|r - \gamma Xs\|)\|r - \gamma Xs\|/\omega(1 - \beta) \\ &\le (1 - \lambda)\|r\| + \lambda^2(\|r\| + \beta_2(1 - \beta)\omega)\beta_2 \\ &= \left[1 - \lambda + \lambda^2(1 + \beta_2(1 - \beta)/\beta)\beta_2\right]\|r\| \\ &\le \left(1 - \lambda + \lambda^2\tau_1\right)\|r\|, \end{aligned} \tag{18}$$

where the second inequality follows from properties of the 1-norm and the 2-norm; the third inequality follows from (15), and (16), the fourth inequality follows from (9), and the last equality follows from $\beta\omega = \|r\|$, and the last inequality follows from using $\beta \le \beta_1$ and letting $\tau_1 := \beta_3/\beta$. Also, from the second equation in (10) we have that

$$\begin{aligned}
\|F(x') - y'\| &= \|F(x') - F(x) - \lambda F'(x)u + F(x) + \lambda F'(x)u - y'\| \\
&\le \|F(x') - F(x) - \lambda F'(x)u\| + \|F(x) + \lambda F'(x)u - y'\| \\
&= r(x, \lambda u) + (1 - \lambda\gamma)\|F(x) - y\| \\
&= (1 - \lambda\gamma + \lambda\tau_2)\|F(x) - y\|,
\end{aligned} \tag{19}$$

where we let $\tau_2 := r(x, \lambda u)/(\lambda\|F(x) - y\|)$. Combining (18) and (19), we have

$$\begin{aligned}
&\|H(x', y') - (\omega e, 0)\|_\rho^2 \\
&= \|\omega e - X'y'\|^2 + \rho^2\|F(x') - y'\|^2 \\
&\le (1 - \lambda + \lambda^2\tau_1)^2\|r\|^2 + (1 - \lambda\gamma + \lambda\tau_2)^2\rho^2\|F(x) - y\|^2 \\
&= \left[\|r\|^2 + \rho^2\|F(x) - y\|^2\right] - 2\lambda\left[\|r\|^2 + \gamma\rho^2\|F(x) - y\|^2\right] \\
&+ \lambda^2\left(1 + 2(1 - \lambda)\tau_1 + \lambda^2\tau_1^2\right)\|r\|^2 \\
&+ \lambda^2\rho^2\left(\gamma^2 + 2(1 - \lambda\gamma)\tau_2/\lambda + \tau_2^2\right)\|F(x) - y\|^2 \\
&\le \left[\|r\|^2 + \rho^2\|F(x) - y\|^2\right] - 2\lambda\gamma\left[\|r\|^2 + \rho^2\|F(x) - y\|^2\right] + \lambda^2(1 + \tau_1^2)\|r\|^2 \\
&+ 2\lambda^2\tau_1\omega\beta_1\|r\| + \lambda^2\rho^2(1 + \tau_2^2)\|F(x) - y\|^2 + 2\lambda\rho\tau_2\omega\beta_1\|F(x) - y\| \\
&= (1 - 2\lambda\gamma + \lambda^2)\|H(x, y) - (\omega e, 0)\|_\rho^2 + \lambda^2(\sigma(\lambda) - 1)(\omega\beta_1)^2 \\
&\le (1 - \lambda\gamma)(\omega\beta_1)^2,
\end{aligned} \tag{20}$$

where the second inequality follows from $0 \le \gamma, \lambda \le 1$, $r = \omega e - Xy$, and (8); the third equality follows from $\tau_1\|r\| = \beta_3\omega$, $\tau_2\|F(x) - y\| = r(x, \lambda u)/\lambda$, (2), and (14); the last inequality follows from $\|H(x, y) - (\omega e, 0)\|_\rho \le \omega\beta_1$ (since $(x, y, \omega) \in \mathcal{N}_{\rho,\beta_1}$) and $\lambda\sigma(\lambda) \le \gamma$ (since $\lambda \le \bar{\lambda}$). Thus, for any $\omega' \in [0, \omega]$, we have from (20) that

$$\begin{aligned}
\|H(x', y') - (\omega' e, 0)\|_\rho &= \|H(x', y') - (\omega e, 0) + (\omega - \omega')(e, 0)\|_\rho \\
&\le \|H(x', y') - (\omega e, 0)\|_\rho + (\omega - \omega')\sqrt{n} \\
&\le \sqrt{1 - \lambda\gamma}(\omega\beta_1) + (\omega - \omega')\sqrt{n}.
\end{aligned}$$

Since $\Omega(x', y')$ is the smallest $\omega' \in [0, \infty)$ such that the left-hand side is below $\omega'\beta_1$, $\Omega(x', y')$ is below the smallest $\omega' \in [0, \omega]$ such that the right-hand side is below $\omega'\beta_1$. This yields the second inequality in (11). By (2) and (20), we have

$$\|\omega e - X'y'\| \le \|H(x', y') - (\omega e, 0)\|_\rho \le \omega\beta_1 < \omega$$

and, by (17) and $0 \le \lambda \le 1$ and $x > 0$, we have $x' = x + \lambda u > 0$. This implies $y' > 0$ and the first inequality in (11) follows. Finally, it follows from the definition of $\Omega(\cdot, \cdot)$ that for any $\lambda \in [0, 1]$ satisfying (11), (x_+, y_+, ω_+) given by (12) is in $\mathcal{N}_{\rho,\beta_1}$.

(c) Assume there exist continuous functions $\eta : \Re^n_+ \mapsto (0, \infty)$ and $L : \Re^n_{++} \mapsto [0, \infty)$ such that (5) and (6) hold, respectively. Let $I = \{\ i \in \{1, \ldots, n\} \mid x_i \ge \sqrt{\omega}\ \}$ and $J = \{1, \ldots, n\} \backslash I$. We have from (17) that

$$\|u_J\| \le \|X_J\|\|(X_J)^{-1}u_J\| \le \sqrt{\omega}\beta_2. \tag{21}$$

Also, it is easily seen that u satisfies $(M + YX^{-1})u = X^{-1}(r - \gamma Xs)$, so

$$(M_{II} + Y_I X_I^{-1})u_I = X_I^{-1}(r - \gamma Xs)_I - M_{IJ}u_J.$$

By assumption, M_{II} is nonsingular, so multiplying both sides by M_{II}^{-1} and using (5) yields

$$\begin{aligned}
\|u_I\| &= \|M_{II}^{-1}\left[X_I^{-1}(r - \gamma Xs)_I - M_{IJ}u_J - Y_I X_I^{-1}u_I\right]\| \\
&\le \|M_{II}^{-1}\|\left[\|X_I^{-1}\|\|(r - \gamma Xs)_I\| + \|M_{IJ}\|\|u_J\| + \|y_I\|_\infty \|X_I^{-1}u_I\|\right] \\
&\le \eta(x)\left[\|X_I^{-1}\|\|(r - \gamma Xs)_I\| + \|M_{IJ}\|\|u_J\| + \|y_I\|_\infty \beta_2\right] \\
&\le \eta(x)\left[\|(r - \gamma Xs)_I\|/\sqrt{\omega} + \|M_{IJ}\|\sqrt{\omega}\beta_2 + \|y_I\|_\infty \beta_2\right] \\
&\le \eta(x)\left[\sqrt{\omega}\beta_2 + \|M_{IJ}\|\sqrt{\omega}\beta_2 + \|y_I\|_\infty \beta_2\right] \\
&\le \eta(x)\left[\sqrt{\omega} + \|M_{IJ}\|\sqrt{\omega} + \sqrt{\omega}(1+\beta_1)\right]\beta_2, \qquad (22)
\end{aligned}$$

where the second inequality also uses (17), the third inequality uses (21), the fourth inequality follows from (9), and the last inequality uses the fact that $Xy \le \omega(1+\beta_1)e$ (see (8)), so $y_I \le \omega(1+\beta_1)X_I^{-1}e \le \sqrt{\omega}(1+\beta_1)e$. Also, for any $\lambda \in [0,1]$, we have from (17) that (6) holds with $z = \lambda u$. This implies $\sigma(\lambda)$ given by (14) can be bounded above as

$$\sigma(\lambda) \le (1 + \beta_3/\beta_1)^2 + 2\rho L(x)\|u\|^2/(\beta_1\omega) + \rho^2 L(x)^2\|u\|^4/(\beta_1\omega)^2. \qquad (23)$$

Combining (21) with (22), we see that the right-hand side of (23) is bounded above by a continuous function of x depending on $\beta_1, \beta_2, \rho, F', \eta, L$ only.

(d) If F is affine, then $r(x, \lambda u) = 0$ for all $\lambda \in [0,1]$, so $\sigma(\lambda)$ given by (14) is bounded above by a constant depending on β_1 and β_2 only.

(e) Assume there exists an $(x^*, y^*) \in S$ that is strongly nondegenerate and for which there exist scalars $\mu > 0$ and $\epsilon > 0$ satisfying (7). The former implies $(x_I^*, y_J^*) > 0$ and $F'(x^*)_{II}$ is nonsingular for some partition I, J of $\{1, \ldots, n\}$, so the Jacobian

$$\begin{bmatrix} Y^* & X^* \\ F'(x^*) & -I \end{bmatrix}$$

is nonsingular, implying that there exists constant $\kappa > 0$ (depending on F' and (x^*, y^*)) such that

$$\left\| \begin{bmatrix} Y & X \\ F'(x) & -I \end{bmatrix}^{-1} \right\| \le \kappa \qquad (24)$$

whenever $\|(x,y) - (x^*, y^*)\| \le 1/\kappa$. Assume (x,y) satisfies $\|(x,y) - (x^*, y^*)\| \le 1/\kappa$ and let $C := \kappa(1 + 1/\rho)\beta_1$. Then, (10) and (24) yield

$$\begin{aligned}
\|(u,v)\| &= \left\| \begin{bmatrix} Y & X \\ F'(x) & -I \end{bmatrix}^{-1} \begin{bmatrix} \omega e - Xy \\ \gamma(y - F(x)) \end{bmatrix} \right\| \\
&\le \kappa \, \|(\omega e - Xy, \gamma(y - F(x)))\| \\
&\le \kappa(\|\omega e - Xy\| + \gamma\|y - F(x)\|) \\
&\le \kappa(1 + 1/\rho)\omega\beta_1 \;=\; C\omega,
\end{aligned}$$

where the last inequality follows from (8) and $\gamma \le 1$. (The above argument is based on the proof of Lemma 5.2 in [29].) If, in addition, $\|x - x^*\| \le \epsilon$ and $\omega \le \epsilon/C$, we have by a similar argument as in the proof of (c) (but with (6) replaced by (7)) that for any $\lambda \in [0,1]$, the relation (23) holds (with $L(x)$ replaced by μ) so that $\sigma(\lambda)$ is bounded above by a constant (depending on $\beta_1, \beta_2, \rho, F', (x^*, y^*), \mu$). □

3. Pure Newton step and its properties. We describe below the Newton step based on (4). Fix any scalar $\beta_1 \in (0,1)$ and any $\rho \in (0,\infty)$.

PURE NEWTON STEP. *For a given* $(x,y,\omega) \in \mathcal{N}_{\rho,\beta_1}$*, we let* $(\hat{u},\hat{v})$ *be the vector in* $\Re^{2n}$ *satisfying* (4)*, that is,*

$$\begin{bmatrix} Y & X \\ F'(x) & -I \end{bmatrix} \begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix} = \begin{bmatrix} -Xy \\ y - F(x) \end{bmatrix}, \tag{25}$$

and let

$$(\hat{x}_+, \hat{y}_+) := (x + \hat{u}, y + \hat{v}). \tag{26}$$

The following lemma roughly says that whenever (x,y) is near a strongly nondegenerate solution, $(\hat{u},\hat{v})$ is in the order of ω. The proof of this lemma is nearly identical to that of Lemma 2.1(e) (which in turn is based on the proof of Lemma 5.2 in [29]) and, for simplicity, is omitted.

LEMMA 3.1. *Fix any* $\beta_1 \in (0,1)$ *and* $\rho \in (0,\infty)$*. Assume there exists an* $(x^*,y^*) \in S$ *that is strongly nondegenerate and for which there exist scalars* $\mu > 0$ *and* $\epsilon > 0$ *satisfying* (7)*. Then there exist positive constants* δ *and* C *(depending on* β_1, ρ, F', (x^*,y^*), μ, ϵ*) such that for any* $(x,y,\omega) \in \mathcal{N}_{\rho,\beta_1}$ *satisfying* $\|(x,y)-(x^*,y^*)\| \le \delta$*, the vector* $(\hat{u},\hat{v})$ *satisfying* (25) *also satisfies* $\|(\hat{u},\hat{v})\| \le C\omega$*.*

4. Algorithm and convergence analysis. In this section we describe our algorithm for solving CP and use the results from previous sections to analyze its convergence. The algorithm uses either the centering Newton direction alone or a convex combination of the centering Newton direction and the pure Newton direction (also called the affine-scaling direction), whichever yields the greater decrease in ω.

ALGORITHM 4.1. *Choose any* β_1, β_2 *with* $0 < \beta_1/(1-\beta_1) < \beta_2 < 1$ *and any* $\rho \in (0,\infty)$ *and* $\psi \in (0,1)$*. Also choose any* $(x^0,y^0,\omega^0) \in \mathcal{N}_{\rho,\beta_1}$*. For any* $(x,y,\omega) \in \Re^{2n+1}_+$ *and any* $(\hat{x},\hat{y}) \in \Re^{2n}$*, let*

$$\Theta_\omega(x,y,\hat{x},\hat{y}) := \sup_{\theta\in[0,1]} \{\theta | (\theta\hat{x} + (1-\theta)x, \theta\hat{y} + (1-\theta)y, (1-\theta)\omega) \in \mathcal{N}_{\rho,\beta_1}\}, \tag{27}$$

with $\Theta_\omega(x,y,\hat{x},\hat{y}) := -\infty$ *if no* θ *satisfies the constraints of* (27)*. For* $t = 0,1,\ldots,$ *if* $\omega^t = 0$*, we let*

$$(x^{t+1}, y^{t+1}, \omega^{t+1}) := (x^t, y^t, \omega^t); \tag{28}$$

otherwise, we generate $(x^{t+1}, y^{t+1}, \omega^{t+1})$ *from* (x^t,y^t,ω^t) *as follows:*

(1a) *Apply centering Newton step (see* (9)–(12)*) with* $(x,y,\omega) = (x^t,y^t,\omega^t)$ *and denote the resulting* (γ,u,v,λ) *and* (x_+,y_+,ω_+) *by, respectively,* $(\gamma^t,u^t,v^t,\lambda^t)$ *and* $(x^t_+, y^t_+, \omega^t_+)$*.*

(1b) *Apply pure Newton step (see* (25)–(26)*) with* $(x,y) = (x^t,y^t)$ *and denote the resulting* $(\hat{u},\hat{v})$ *and* $(\hat{x}_+,\hat{y}_+)$ *by, respectively,* $(\hat{u}^t,\hat{v}^t)$ *and* $(\hat{x}^t_+,\hat{y}^t_+)$*.*

(2) *Let* $\theta^t := \Theta_{\omega^t}(x^t_+, y^t_+, \hat{x}^t_+, \hat{y}^t_+)$*, and let*

$$(x^{t+1}, y^{t+1}, \omega^{t+1}) := \begin{cases} (x^t_+, y^t_+, \omega^t_+) & \text{if } \omega^t_+ \le (1-\theta^t)\omega^t, \\ (\theta^t\hat{x}^t_+ + (1-\theta^t)x^t_+, \\ \quad \theta^t\hat{y}^t_+ + (1-\theta^t)y^t_+, (1-\theta^t)\omega^t) & \text{otherwise} \end{cases}. \tag{29}$$

Note 1. One choice of (x^0,y^0,ω^0) that satisfies $(x^0,y^0,\omega^0) \in \mathcal{N}_{\rho,\beta_1}$ is

$$x^0 = \alpha_x e, \quad y^0 = \alpha_y e, \quad \omega^0 = \alpha_x\alpha_y,$$

with α_x chosen large enough so $\rho\sqrt{n}/\alpha_x < \beta_1$ and then with α_y chosen large enough so $\rho\|F(\alpha_x e)/\alpha_y - e\|/\alpha_x \le \beta_1$. Alternatively, we can start with any $(x^0, y^0) \in \Re^{2n}_{++}$ provided we modify Algorithm 4.1 as follows: let $z^0 = X^0 y^0$ and choose $\omega^0 = \min_i z_i^0$ and $\rho = \omega^0\beta_1/\|F(x^0) - y^0\|$; replace the vector e everywhere by z^0/ω^0 and replace $\sqrt{n}$ in (11) by $\|z^0\|/\omega^0$. Our convergence results (Theorem 4.2) also extend to this modified algorithm.

Note 2. If F is affine, then θ^t can be computed exactly by solving a quartic equation in one variable; otherwise, we must solve an optimization problem in one variable. Alternatively, we can estimate θ^t. One such estimate, suggested by Lemma 4.1 to follow, is

$$1 - C^t\omega^t/[1 - (1 - \lambda^t\gamma^t)^2],$$

where $C^t := \max\{\|(u^t, v^t)\|/\omega^t, \|(\hat{u}^t, \hat{v}^t)\|/\omega^t\}^2(1+\rho\mu^t)(1+\sqrt{2})/\beta_1$ and μ^t is an upper estimate of $\limsup_{z\to 0} r(x^t, z)/\|z\|^2$. In practice, we use the maximum of $1 - \omega_+^t/\omega^t$ and this estimate as the lower endpoint in a binary search procedure.

Note 3. The pure Newton step is needed only for local quadratic convergence and may be removed (by setting $(x^{t+1}, y^{t+1}, \omega^{t+1}) := (x_+^t, y_+^t, \omega_+^t)$ for all t) without affecting the global convergence behavior of the algorithm as stated in Theorem 4.2(a)–(c). Also, the idea of taking a convex combination of the centering Newton direction and the pure Newton direction to achieve superlinear convergence is not new and, as noted in [5], traces back to a work of McShane [14]. However, our mechanism for switching from the centering Newton direction to the convex combination, as given in (29), appears to be new. Also, we consider an infeasible interior-point method for solving monotone CP rather than a feasible interior-point method for solving monotone linear CP as considered in [5, 14].

To analyze the local convergence rate of Algorithm 4.1, we further need the following technical lemma.

LEMMA 4.1. *Assume there exists an $x^* \in \Re^n_+$ and scalars $\mu > 0$ and $\epsilon > 0$ such that* (7) *holds. Fix any $\beta_1 \in (0,1)$ and $\rho \in (0,\infty)$. For any positive scalar C, any $(x, y, \omega) \in \mathcal{N}_{\rho,\beta_1}$, and any $\gamma \in [0,1]$ and $\lambda \in [0,1]$ satisfying*

$$(30)\quad \|x - x^*\| \le \epsilon,\ \|(u,v)\| \le C\omega,\ \|(\hat{u},\hat{v})\| \le C\omega,\ \omega \le \epsilon/C,\ C'\omega \le 1 - (1-\lambda\gamma)^2,$$

where $C' := C^2(1+\rho\mu)(1+\sqrt{2})/\beta_1$ and (u,v) and $(\hat{u},\hat{v})$ satisfy (10) *and* (25)*, respectively, we have*

$$\Theta_\omega(x+\lambda u, y+\lambda v, x+\hat{u}, y+\hat{v}) \ge 1 - C'\omega/[1-(1-\lambda\gamma)^2].$$

Proof. Consider any C, (x,y,ω), γ, λ, and C', (u,v), $(\hat{u},\hat{v})$ satisfying the hypothesis of the lemma. For any $\theta \in [0,1]$, we have upon letting

$$(x', y') := \theta(x+\hat{u}, y+\hat{v}) + (1-\theta)(x+\lambda u, y+\lambda v)$$

that

$$\begin{aligned}
&\|(1-\theta)\omega e - X'y'\| \\
&= \|(1-\theta)(1-\lambda)(\omega e - Xy) + \theta(1-\theta)\lambda(V\hat{u} + U\hat{v}) + \theta^2\hat{U}\hat{v} + (1-\theta)^2\lambda^2 Uv\| \\
&\le (1-\theta)(1-\lambda)\|\omega e - Xy\| + \theta(1-\theta)\lambda[\|v\|\|\hat{u}\| + \|u\|\|\hat{v}\|] \\
&\quad + \theta^2\|\hat{u}\|\|\hat{v}\| + (1-\theta)^2\lambda^2\|u\|\|v\| \\
&\le (1-\theta)(1-\lambda\gamma)\|\omega e - Xy\| + 2\theta(1-\theta)\lambda(C\omega)^2 + \theta^2(C\omega)^2 + (1-\theta)^2\lambda^2(C\omega)^2 \\
&= (1-\theta)(1-\lambda\gamma)\|\omega e - Xy\| + (\theta + (1-\theta)\lambda)^2(C\omega)^2,
\end{aligned}$$

where the first equality uses the first equation in (10) and in (25) and the last inequality uses (30). Also, we have from the second equation in (10) and in (25) that

$$\begin{aligned}\|F(x')-y'\| &= \|F(x)+F'(x)(x'-x)-y'+F(x')-F(x)-F'(x)(x'-x)\|\\ &\le \|F(x)+F'(x)(x'-x)-y'\|+\|F(x')-F(x)-F'(x)(x'-x)\|\\ &= (1-\theta)(1-\lambda\gamma)\|F(x)-y\|+r(x,\theta\hat{u}+(1-\theta)\lambda u)\\ &\le (1-\theta)(1-\lambda\gamma)\|F(x)-y\|+\mu\|\theta\hat{u}+(1-\theta)\lambda u\|^2\\ &\le (1-\theta)(1-\lambda\gamma)\|F(x)-y\|+\mu(\theta+(1-\theta)\lambda)^2(C\omega)^2,\end{aligned}$$

where the second inequality follows from using (30) (so $\|x-x^*\|\le\epsilon$ and $\|\theta\hat{u}+(1-\theta)\lambda u\|\le\theta\|\hat{u}\|+(1-\theta)\lambda\|u\|\le C\omega\le\epsilon$) and (7); the last inequality uses (30). Combining the above two relations and using $\theta+(1-\theta)\lambda\le 1$, we obtain

$$\begin{aligned}&\|H(x',y')-((1-\theta)\omega e,0)\|_\rho^2\\ &=\|(1-\theta)\omega e-X'y'\|^2+\rho^2\|F(x')-y'\|^2\\ &\le\left[(1-\theta)(1-\lambda\gamma)\|\omega e-Xy\|+(C\omega)^2\right]^2\\ &+\rho^2\left[(1-\theta)(1-\lambda\gamma)\|F(x)-y\|+\mu(C\omega)^2\right]^2\\ &=(1-\theta)^2(1-\lambda\gamma)^2\|H(x,y)-(\omega e,0)\|_\rho^2+(C\omega)^4(1+\rho^2\mu^2)\\ &+2(1-\theta)(1-\lambda\gamma)(C\omega)^2\left[\|\omega e-Xy\|+\rho^2\mu\|F(x)-y\|\right]\\ &\le(1-\theta)^2(1-\lambda\gamma)^2\beta_1^2\omega^2+C^4(1+\rho^2\mu^2)\omega^4\\ &+2(1-\theta)(1-\lambda\gamma)C^2(1+\rho\mu)\beta_1\omega^3,\end{aligned}\tag{31}$$

where the last inequality follows from (8). The right-hand side of (31) is below $[(1-\theta)\omega\beta_1]^2$ for all $\theta\in[0,1-\xi]$, where ξ solves the quadratic equation $a\xi^2-2b\omega\xi-c\omega^2=0$ with

$$a=[1-(1-\lambda\gamma)^2]\beta_1^2,\quad b=(1-\lambda\gamma)C^2(1+\rho\mu)\beta_1,\quad c=C^4(1+\rho^2\mu^2).$$

(Note that $\xi=(b+\sqrt{b^2+ac})\omega/a\le C'\omega/[1-(1-\lambda\gamma)^2]$, so (30) yields $\xi\le 1$.) Thus, the left-hand side of (31) is below $[(1-\theta)\omega\beta_1]^2$ for all $\theta\in[0,1-\xi]$. Moreover, we have $(x',y')\ge 0$ for all such θ (since if any component of (x',y') is below 0, then $\|(1-\theta)\omega e-X'y'\|>(1-\theta)\omega$ and the left-hand side of (31) would be greater than $[(1-\theta)\omega]^2$). Since $\Theta_\omega(x+\lambda u,y+\lambda v,x+\hat{u},y+\hat{v})$ is the largest $\theta\in[0,1]$ such that the left-hand side of (31) is below $[(1-\theta)\omega\beta_1]^2$ and $(x',y')\ge 0$, this yields

$$\Theta_\omega(x+\lambda u,y+\lambda v,x+\hat{u},y+\hat{v})\ge 1-\xi\ge 1-C'\omega/[1-(1-\lambda\gamma)^2].\qquad\square$$

By using properties of the two Newton steps (see Lemmas 2.1 and 3.1) and the preceding lemma, we have the following main result, giving sufficient conditions for global (Q-linear) convergence and local quadratic convergence of Algorithm 4.1.

THEOREM 4.2. *Let* β_1, β_2, ρ, ψ *and* $\{(x^t,y^t,\omega^t,\gamma^t,u^t,v^t,\lambda^t,x_+^t,y_+^t,\hat{u}^t,\hat{v}^t,\hat{x}_+^t,\hat{y}_+^t,\theta^t)\}_{t=0,1,\ldots}$ *be generated by Algorithm* 4.1. *Then either* $\omega^t=0$ *for some* t *or the following hold:*

(a) *For all* t, *we have* $(x^t,y^t,\omega^t)\in\mathcal{N}_{\rho,\beta_1}$ *and* $\|(X^t)^{-1}u^t\|<1$ *and* $\omega^{t+1}\le\omega_+^t\le\omega^t$.

(b) *If* $\{x^t\}$ *has a convergent subsequence, then* $\{\omega^t\}\to 0$ *and every cluster point of* $\{(x^t,y^t)\}$ *is in* S.

(c) *Assume* $\{x^t\}$ *is bounded. If there also exist continuous functions* $\eta : \Re^n_+ \mapsto (0,\infty)$ *and* $L : \Re^n_{++} \mapsto [0,\infty)$ *such that* (5) *and* (6) *hold, respectively, then there exists a* $c \in (0,1)$ *(depending on* β_1, β_2, ρ, F', η, L, $\sup_t \|x^t\|$*) such that* $\omega^{t+1} \le c\omega^t$ *for all* t*. Alternatively, if every* $(x^*, y^*) \in S$ *is strongly nondegenerate and for which there exist scalars* $\mu > 0$ *and* $\epsilon > 0$ *satisfying* (7)*, then* $\limsup_{t\to\infty} \omega^{t+1}/(\omega^t)^2 < \infty$*.*

(d) *Assume* F *is affine and* $\{x^t\}$ *is bounded. Then there exists a* $c \in (0,1)$ *(depending on* β_1, β_2, ρ, $\sup_t \|x^t\|$*) such that* $\omega^{t+1} \le c\omega^t$ *for all* t*. If every element of* S *is strongly nondegenerate, then* $\limsup_{t\to\infty} \omega^{t+1}/(\omega^t)^2 < \infty$*.*

Proof. (a): We argue by induction on t that $(x^t, y^t, \omega^t) \in \mathcal{N}_{\rho,\beta_1}$ for all t. This clearly holds for $t = 0$. Suppose it holds for some $t \ge 0$. Then, we have from Lemma 2.1(b) that $(x^t_+, y^t_+, \omega^t_+) \in \mathcal{N}_{\rho,\beta_1}$ and from the definition of θ^t that $(\theta^t \hat{x}^t_+ + (1-\theta^t)x^t_+, \theta^t \hat{y}^t_+ + (1-\theta^t)y^t_+, (1-\theta^t)\omega^t) \in \mathcal{N}_{\rho,\beta_1}$ whenever $\theta^t \in [0,1)$. Since $\omega^{t+1} \ne 0$ by assumption so that $\theta^t \ne 1$, (29) yields $(x^{t+1}, y^{t+1}, \omega^{t+1}) \in \mathcal{N}_{\rho,\beta_1}$. The remaining results readily follow from Lemma 2.1(a) and (11)–(12).

(b): Assume $\{x^t\}$ has a convergent subsequence $\{x^t\}_{t\in T}$. We argue that $\{\omega^t\} \to 0$ by contradiction. If $\{\omega^t\} \not\to 0$, then since $\{\omega^t\}$ is nonincreasing (see (a)), we have $\omega^t \ge C$ for all t for some scalar $C > 0$. Then $\omega^t \ge \omega^t_+ \ge \omega^{t+1}$ for all t (see (a)) implies $\{\omega^t_+/\omega^t\} \to 1$, which together with

$$\omega^t_+/\omega^t = \Omega(x^t + \lambda^t u^t, y^t + \lambda^t v^t)/\omega^t \le (\beta_1\sqrt{1-\lambda^t\gamma^t} + \sqrt{n})/(\beta_1 + \sqrt{n}) \quad \forall t$$

(see (11) and (12)) implies $\{\lambda^t\gamma^t\} \to 0$. Since $\{x^t\}_{t\in T}$ converges so $\{\gamma^t\}_{t\in T} \not\to 0$ (see (13)), this implies $\{\lambda^t\}_{t\in T} \to 0$. Thus, for all $t \in T$ sufficiently large, we have $\lambda^t \ne 1$, implying $\lambda = \lambda^t/\psi$ does not satisfy (11) (with $(x, y, u, v, \gamma) = (x^t, y^t, u^t, v^t, \gamma^t)$). Then, by Lemma 2.1(b), it must be that $\lambda^t/\psi > \bar{\lambda}^t$, where $\bar{\lambda}^t$ is the smallest $\lambda \in [0,1]$ satisfying $\lambda\sigma^t(\lambda) = \gamma^t$ with (see (14))

$$\sigma^t(\lambda) := (1 + \beta_3/\beta_1)^2 + 2\rho r(x^t, \lambda u^t)/(\lambda^2\beta_1\omega^t) + \rho^2 r(x^t, \lambda u^t)^2/(\lambda\beta_1\omega^t)^2.$$

Since $\{\lambda^t\}_{t\in T} \to 0$, this implies $\{\bar{\lambda}^t\}_{t\in T} \to 0$ and, since $\{x^t\}_{t\in T}$ converges and $\|(X^t)^{-1}u^t\| < 1$ for all t (see (a)), we also have that $\{u^t\}_{t\in T}$ is bounded. Moreover, the limit point of $\{x^t\}_{t\in T}$ is in $\Re^n_{++}$. (This is because $(x^t, y^t, \omega^t) \in \mathcal{N}_{\rho,\beta_1}$ for all t so, by using (8), we have $X^t y^t \ge \omega^t(1-\beta_1)e$ and $\rho\|F(x^t) - y^t\| \le \omega^t\beta_1$. The latter implies $\{y^t\}_{t\in T}$ is bounded (since $\{F(x^t)\}_{t\in T}$ converges and $\{\omega^t\} \downarrow$), so the former together with $\omega^t \ge C$ for all t implies $\{x^t\}_{t\in T}$ is componentwise bounded away from zero.) Then the continuous differentiability of F on $\Re^n_{++}$ implies $\{r(x^t, \bar{\lambda}^t u^t)/\bar{\lambda}^t\}_{t\in T} \to 0$, which together with $\omega^t \ge C$ for all t yields $\{\bar{\lambda}^t\sigma^t(\bar{\lambda}^t)\}_{t\in T} \to 0$. Since $\{\gamma^t\}_{t\in T} \not\to 0$, this contradicts $\bar{\lambda}^t\sigma^t(\bar{\lambda}^t) = \gamma^t$ for all t. Thus, $\{\omega^t\} \to 0$. Since $(x^t, y^t, \omega^t) \in \mathcal{N}_{\rho,\beta_1}$ for all t (see (a)), it readily follows that every cluster point of $\{(x^t, y^t)\}$ is in S.

(c): Assume $\{x^t\}$ is bounded. By (13), there is a constant $C_1 > 0$ such that $\gamma^t \ge C_1$ for all t.

Suppose there also exist continuous functions $\eta : \Re^n_+ \mapsto (0,\infty)$ and $L : \Re^n_{++} \mapsto [0,\infty)$ such that (5) and (6) hold, respectively. Then, by Lemma 2.1(c) and boundedness of $\{x^t\}$, there is a constant $C_2 > 0$ such that $\sigma^t(\bar{\lambda}^t) \le C_2$ for all t, where $\sigma^t(\cdot)$ and $\bar{\lambda}^t$ are as defined in the proof of (b). Then, by $\bar{\lambda}^t\sigma^t(\bar{\lambda}^t) = \gamma^t$, we have $\bar{\lambda}^t \ge \gamma^t/C_2 \ge C_1/C_2$. Also, we argued in the proof of (b) that if $\lambda^t \ne 1$, then $\lambda^t/\psi > \bar{\lambda}^t$, and hence $\lambda^t \ge C_3 := \min\{1, \psi C_1/C_2\}$ for all t. This implies (see (11) and (12)) that

$$\omega^t_+/\omega^t = \Omega(x^t + \lambda^t u^t, y^t + \lambda^t v^t)/\omega^t \le (\beta_1\sqrt{1 - C_1C_3} + \sqrt{n})/(\beta_1 + \sqrt{n}),$$

so $\omega^{t+1} \le \omega_+^t \le c\omega^t$ for some constant $c \in (0,1)$.

Alternatively, suppose every element (x^*, y^*) of S is strongly nondegenerate and for which there exist scalars $\mu > 0$ and $\epsilon > 0$ satisfying (7). First, we claim that $\{(x^t, y^t)\}$ converges. To see this, let (x^*, y^*) be any cluster point of $\{(x^t, y^t)\}$. By (b), (x^*, y^*) is in S so (x^*, y^*) is strongly nondegenerate, i.e., $(x_I^*, y_J^*) > 0$ and $F'(x^*)_{II}$ is nonsingular for some partition I, J of $\{1, \ldots, n\}$. By $\{\omega^t\} \to 0$ (see (b)) and by using an argument similar to the one above (but with Lemma 2.1(c) replaced by Lemma 2.1(e)), we have the existence of constants $\delta_1 > 0$, $C_2 > 0$, and $c \in (0,1)$ such that

$$\|(u^t, v^t)\| \le C_2\omega^t \quad \text{and} \quad \omega^{t+1} \le \omega_+^t \le c\omega^t \quad \text{and} \quad \lambda^t \ge 1/C_2 \tag{32}$$

for all t with $\|(x^t, y^t) - (x^*, y^*)\| \le \delta_1$ and $\omega^t \le \delta_1$. By taking C_2 larger and δ_1 smaller if necessary, we also have that $C_2 \ge 1/2$ and from Lemma 3.1 that

$$\|(\hat{u}^t, \hat{v}^t)\| \le C_2\omega^t \tag{33}$$

for all t with $\|(x^t, y^t) - (x^*, y^*)\| \le \delta_1$ and $\omega^t \le \delta_1$. Let

$$\delta_2 := \delta_1(1-c)/(2C_2) \tag{34}$$

(so, by $C_2 \ge 1/2$, we have $\delta_2 \le \delta_1$) and consider any $\bar{t}$ such that $\|(x^{\bar{t}}, y^{\bar{t}}) - (x^*, y^*)\| \le \delta_1/2$ and $\omega^{\bar{t}} \le \delta_2$. We claim that

$$\|(x^t, y^t) - (x^*, y^*)\| \le C_2 \left(\sum_{k=0}^{t-\bar{t}-1} c^k \right) \delta_2 + \delta_1/2, \qquad \omega^t \le c^{t-\bar{t}}\delta_2 \tag{35}$$

for all $t \ge \bar{t}$. Clearly, (35) holds for $t = \bar{t}$. Suppose (35) holds for some $t \ge \bar{t}$. Then (34) and $c \in (0,1)$ yield $\|(x^t, y^t) - (x^*, y^*)\| \le \delta_2 \le \delta_1$ and $\omega^t \le \delta_1$, so (32) and (33) hold. Hence, (29) together with (12) and (26) yield that either

$$\|(x^{t+1}, y^{t+1}) - (x^t, y^t)\| = \|(x_+^t, y_+^t) - (x^t, y^t)\| = \lambda^t\|(u^t, v^t)\| \le \; C_2\omega^t$$

or

$$\begin{aligned} \|(x^{t+1}, y^{t+1}) - (x^t, y^t)\| &= \|\theta^t(\hat{x}_+^t, \hat{y}_+^t) + (1-\theta^t)(x_+^t, y_+^t) - (x^t, y^t)\| \\ &= \|\theta^t(\hat{u}^t, \hat{v}^t) + (1-\theta^t)\lambda^t(u^t, v^t)\| \\ &\le \theta^t\|(\hat{u}^t, \hat{v}^t)\| + (1-\theta^t)\lambda^t\|(u^t, v^t)\| \le C_2\omega^t, \end{aligned}$$

where the inequalities also use $\lambda^t \le 1$. Then (35) yields

$$\begin{aligned} \|(x^{t+1}, y^{t+1}) - (x^*, y^*)\| &\le \|(x^{t+1}, y^{t+1}) - (x^t, y^t)\| + \|(x^t, y^t) - (x^*, y^*)\| \\ &\le C_2 c^{t-\bar{t}}\delta_2 + C_2 \left(\sum_{k=0}^{t-\bar{t}-1} c^k \right) \delta_2 + \delta_1/2 \\ &= C_2 \left(\sum_{k=0}^{t-\bar{t}} c^k \right) \delta_2 + \delta_1/2. \end{aligned}$$

We also have from (32) and (35) that

$$\omega^{t+1} \le c\omega^t \le c^{t+1-\bar{t}}\delta_2.$$

Thus, (35) holds when t is replaced by $t+1$. Then, by induction, (35) holds for all $t \geq \bar{t}$ and hence, by (34),

$$\|(x^t, y^t) - (x^*, y^*)\| \leq C_2\delta_2/(1-c) + \delta_1/2 < \delta_1$$

for all $t \geq \bar{t}$. This holds for any δ_1 sufficiently small, so, since (x^*, y^*) is a cluster point of $\{(x^t, y^t)\}$ and $\{\omega^t\} \to 0$ (see (b)), we have $\{(x^t, y^t)\} \to (x^*, y^*)$. Moreover, we have that (32) and (33) hold for all t sufficiently large, so Lemma 4.1 and $\gamma^t \geq C_1$ for all t yield

$$\theta^t \geq 1 - C_3\omega^t/[1-(1-\lambda^t\gamma^t)^2] \geq 1 - C_3\omega^t/[1-(1-C_1/C_2)^2]$$

for all t sufficiently large, with $C_3 := (C_2)^2(1+\rho\mu)(1+\sqrt{2})/\beta_1$. This, together with $\omega^{t+1} \leq (1-\theta^t)\omega^t$ (see (29)), implies that

$$\omega^{t+1} \leq C_3(\omega^t)^2/[1-(1-C_1/C_2)^2]$$

for all t sufficiently large, and hence $\{\omega^t\}$ has local quadratic convergence.

(d): Since F is affine and $\{x^t\}$ is bounded, we have, by an argument analogous to that for (c) (with Lemma 2.1(c) replaced by Lemma 2.1(d)), that $\{\lambda^t\gamma^t\}$ is bounded below by some positive constant (depending on β_1, β_2, ρ, $\sup_t \|x^t\|$), implying $\{\omega^t\} \downarrow 0$ globally Q-linearly. If every element of S is strongly nondegenerate, the result follows from (c). □

It was pointed out to the author by S. J. Wright that the assumption that every element of S be nondegenerate implies that the elements of S are isolated; hence, by a result of Minty [6, Proposition 3.1] (which says that for the monotone CP, the projection of S onto the x-space is convex), S has at most one element. This fact may be used to shorten the proof of Theorem 4.2(c) somewhat.

A careful analysis shows that the constant c in Theorem 4.2(c) satisfies

$$c \geq 1 - \frac{C\rho^2}{\sqrt{n}} \left[\sup_t \left\{ \|x^t\|_\infty \left(1 + \rho L(x^t)\eta(x^t)(1 + \|F'(x^t)\|)^2\right) \right\} \right]^{-2},$$

where C is a constant depending on β_1 and β_2. Also, the convergence result in Theorem 4.2 depends on $\{x^t\}$ being bounded (at least when F is not affine). The following lemma gives conditions on F under which $\{x^t\}$ is guaranteed to be bounded.

LEMMA 4.3. *Fix any $\beta_1 \in (0,1)$, any $\rho \in (0,\infty)$, and any $\omega^0 \in [0,\infty)$. The set $\{\ (x,y) \in \Re^{2n} \mid (x,y,\omega) \in \mathcal{N}_{\rho,\beta_1}$ for some $\omega \in [0,\omega^0]\ \}$ is bounded if any of the following two conditions holds:*

(a) *There exists $\bar{x} \in \Re^n_+$ satisfying $F(\bar{x}) > (\beta_1\omega^0/\rho)e$.*

(b) *F is an R_0-function in the sense that for any sequence $x^1, x^2, \ldots$ in $\Re^n_{++}$ satisfying*

$$\{\|x^t\|\} \to \infty \quad \textit{and} \quad \liminf_{t\to\infty} (\min_i F_i(x^t))/\|x^t\| \geq 0, \tag{36}$$

we have $\{(x^t)^T F(x^t)/\|x^t\|\} \to \infty$.

Proof. Assume condition (a) holds and consider any $(x,y,\omega) \in \mathcal{N}_{\rho,\beta_1}$ with $\omega \in [0,\omega^0]$. Since F is monotone, we have

$$0 \leq (x-\bar{x})^T(F(x) - F(\bar{x})),$$

which can be rewritten as

$$\bar{x}^T y + x^T F(\bar{x}) \leq x^T y + x^T(F(x) - y) + \bar{x}^T(y - F(x)) + \bar{x}^T F(\bar{x}).$$

Also, we have from $(x, y, \omega) \in \mathcal{N}_{\rho,\beta_1}$ that (8) holds, so that

$$x^T y \leq (1 + \beta_1)n\omega \quad \text{and} \quad \rho\|F(x) - y\| \leq \beta_1\omega. \tag{37}$$

The above two relations together with $\bar{x}^T y \geq 0$, $x^T F(\bar{x}) \geq \|x\|_1(\min_i F_i(\bar{x}))$, and $\omega \leq \omega^0$ yield

$$\begin{aligned}\|x\|_1(\min_i F_i(\bar{x})) &\leq x^T y + \|x\|\|F(x) - y\| + \|\bar{x}\|\|y - F(x)\| + \bar{x}^T F(\bar{x}) \\ &\leq (1 + \beta_1)n\omega^0 + \|x\|_1\beta_1\omega^0/\rho + \|\bar{x}\|\beta_1\omega^0/\rho + \bar{x}^T F(\bar{x}),\end{aligned}$$

and hence

$$\|x\|_1 \leq \left[(1 + \beta_1)n\omega^0 + \|\bar{x}\|\beta_1\omega^0/\rho + \bar{x}^T F(\bar{x})\right] / \left[\min_i F_i(\bar{x}) - \beta_1\omega^0/\rho\right].$$

Assume condition (b) holds. We argue by contradiction. Suppose there exists a sequence $\{(x^t, y^t, \omega^t)\}$ in $\Re^{2n+1}$ such that $(x^t, y^t, \omega^t) \in \mathcal{N}_{\rho,\beta_1}$ and $\omega^t \in [0, \omega^0]$ for all t and yet $\{\|x^t\|\} \to \infty$. The first two relations yield (see (37))

$$(x^t)^T y^t \leq (1 + \beta_1)n\omega^0 \quad \text{and} \quad \rho\|F(x^t) - y^t\| \leq \beta_1\omega^0,$$

which together with $y^t \geq 0$ for all t yields

$$\liminf_{t\to\infty} (x^t)^T F(x^t)/\|x^t\| < \infty, \ \ \{\|F(x^t) - y^t\|/\|x^t\|\} \to 0, \ \ \liminf_{t\to\infty} (\min_i y^t)/\|x^t\| \geq 0,$$

contradicting the assumption of F being an R_0-function. □

COROLLARY 4.4. *If there exists a constant function $\eta : \Re^n_+ \mapsto (0, \infty)$ such that* (5) *holds and F is twice continuously differentiable on an open set containing $\Re^n_+$, then $\{(x^t, y^t)\}$ is bounded and $\{\omega^t\}$ converges to zero globally Q-linearly and, if every $(x^*, y^*) \in S$ satisfies $x^* + y^* > 0$, locally quadratically.*

Proof. The assumptions imply F is an R_0-function and there exists a continuous function $L : \Re^n_{++} \mapsto [0, \infty)$ such that (6) holds. Moreover, every $(x^*, y^*) \in S$ satisfying $x^* + y^* > 0$ is strongly nondegenerate and for which there exist scalars $\mu > 0$ and $\epsilon > 0$ satisfying (7). The result then follows from Lemma 4.3(b) and Theorem 4.2(c). □

The notion of an R_0-function may be viewed as a generalization of the notion of a uniform P-function (see [6, 19, 27]) and, more generally, of a function that is coercive in the Hadamard sense [18]. When F is affine of the form $F(x) = Mx + q$, F being an R_0-function reduces to M being an R_0-matrix [2]. This is shown in the lemma below.

LEMMA 4.5. *If F is a uniform P-function on $\Re^n_{++}$ or if $F(x) = Mx + q$ for all $x \in \Re^n_{++}$, where M is an $n \times n$ R_0-matrix and $q \in \Re^n$, then F is an R_0-function in the sense of Lemma* 4.3(b).

Proof. Suppose F is a uniform P-function on $\Re^n_{++}$; then there exists a scalar $\nu > 0$ such that

$$\max_{i=1,\ldots,n} (y_i - x_i)(F_i(y) - F_i(x)) \geq \nu\|y - x\|^2 \quad \forall x, y \in \Re^n_{++}. \tag{38}$$

Consider any sequence $x^1, x^2, \ldots$ in $\Re^n_{++}$ satisfying (36). Fix any $\bar{x} \in \Re^n_{++}$. Then, for each t, by letting $x = x^t$ and $y = \bar{x}$ in (38) and dividing both sides by $\|x^t\|^2$, we have

$$\max_{i=1,\ldots,n} \left\{ \left(\bar{x}_i F_i(\bar{x}) - x^t_i F_i(\bar{x}) - \bar{x}_i F_i(x^t) + x^t_i F_i(x^t)\right)/\|x^t\|^2 \right\} \geq \nu\|\bar{x} - x^t\|^2/\|x^t\|^2.$$

Upon letting $t \to \infty$ and using $\bar{x} \geq 0$ and (36), the above relation yields

$$\liminf_{t\to\infty} \max_{i=1,\ldots,n} \left\{ x_i^t F_i(x^t)/\|x^t\|^2 \right\} \geq \nu.$$

We also have from (36) that

$$\liminf_{t\to\infty} \left\{ x_i^t F_i(x^t)/\|x^t\|^2 \right\} \geq 0, \quad i = 1, \ldots, m.$$

The above two relations imply that $\liminf_{t\to\infty}\{(x^t)^T F(x^t)/\|x^t\|^2\} \geq \nu$ and, hence, $\{(x^t)^T F(x^t)/\|x^t\|\} \to \infty$.

Suppose $F(x) = Mx + q$ for all $x \in \Re^n_{++}$, where M is an $n \times n$ R_0-matrix and $q \in \Re^n$. We argue by contradiction. Suppose there exists a sequence $x^1, x^2, \ldots$ in $\Re^n_{++}$ satisfying (36) and yet $\{(x^t)^T(Mx^t + q)/\|x^t\|\} \not\to \infty$. By passing to a subsequence if necessary, we can assume that

$$\limsup_{t\to\infty} \{(x^t)^T(Mx^t + q)/\|x^t\|\} < \infty.$$

Let y denote any cluster point of $\{x^t/\|x^t\|\}$ (so that $y \neq 0$). Then, (36) and the above relation imply $y \geq 0$, $My \geq 0$, and $y^T My = 0$, contradicting M being an R_0-matrix. ▯

The next lemma shows that the assumptions of Corollary 4.4 hold if F is twice continuously differentiable on an open set containing $\Re^n_+$ and is a uniform P-function on $\Re^n_{++}$.

Lemma 4.6. *If F is continuously differentiable and a uniform P-function on $\Re^n_{++}$, then there exists a constant function $\eta : \Re^n_+ \mapsto (0, \infty)$ such that* (5) *holds.*

Proof. Since F is a uniform P-function on $\Re^n_{++}$, there exists a scalar $\nu > 0$ such that (38) holds. Consider any $x \in \Re^n_{++}$ and any $I \subset \{1, \ldots, n\}$. For every nonzero $d \in \Re^n$, we have $x + \lambda d > 0$ for all $\lambda > 0$ sufficiently small so that (38) yields

$$\max_{i=1,\ldots,n} \{\lambda d_i(F_i(x + \lambda d) - F_i(x))\} \geq \nu\lambda^2\|d\|^2.$$

Dividing both sides by λ^2 and letting $\lambda \to 0$, we obtain

$$\max_{i=1,\ldots,n} d_i M_i d \geq \nu\|d\|^2,$$

where we let $M = F'(x)$. This holds for all nonzero d and, in the case where $d_i = 0$ for all $i \notin I$, we obtain

$$\max_{i\in I} d_i M_{iI} d_I \geq \nu\|d_I\|^2.$$

Since the left-hand side is bounded above by $\|d_I\|\|M_{II}d_I\|$, this yields $\|M_{II}d_I\| \geq \nu\|d_I\|$. Thus, M_{II} is invertible and, for every $y \in \Re^n$, we have upon letting $d_I = (M_{II})^{-1}y_I$ that

$$\|(M_{II})^{-1}y_I\| = \|d_I\| \leq \|M_{II}d_I\|/\nu = \|y_I\|/\nu,$$

implying $\|(M_{II})^{-1}\| \leq 1/\nu$. Since the preceding choice of $x \in \Re^n_{++}$ and $I \subset \{1, \ldots, n\}$ was arbitrary, this shows that (5) holds with $\eta \equiv 1/\nu$. ▯

An important case in which F, in addition to being monotone, satisfies the assumption of Lemma 4.6 is when $F(x) = Mx + q$ for all $x \in \Re^n_{++}$, with M an $n \times n$

positive semidefinite P-matrix and $q \in \Re^n$. It is well known that any positive definite matrix is a positive semidefinite P-matrix [2, p. 153], but the converse is not true, as is shown by the following 2×2 example suggested to the author by J.-S. Pang:

$$M = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}.$$

In general, the class of monotone uniform P-functions is significantly broader than the class of strongly monotone functions.

Finally, it follows from Theorem 4.2(c) and Lemmas 4.5 and 4.6 that if F, in addition to being monotone, is a uniform P-function on $\Re^n_{++}$ with Lipschitz continuous Jacobian on $\Re^n_{++}$, then $\{(x^t, y^t, \omega^t)\}$ generated by Algorithm 4.1 is bounded and $\{\omega^t\}$ converges to zero globally Q-linearly. While these assumptions on F may seem to be somewhat restrictive, it is worth noting that existing global Q-linear convergence results for the monotone CP all require either similar assumptions (see [26, Theorem 2.2] and [6, Proposition 4.4(b)]; in the latter, the assumption of F being Lipschitz continuous is missing but is actually needed) or a scaled Lipschitzian assumption (see [22, 25]).

Acknowledgment. Thanks are due to an anonymous reviewer whose questions and comments led to several improvements in the paper.

REFERENCES

[1] R. W. Cottle, F. Giannessi, and J.-L. Lions, eds., *Variational Inequalities and Complementarity Problems: Theory and Applications*, John Wiley & Sons, New York, NY, 1980.

[2] R. W. Cottle, J.-S. Pang, and R. E. Stone, *The Linear Complementarity Problem*, Academic Press, New York, NY, 1992.

[3] A. Fischer, *An NCP-function and its use for the solution of complementarity problems*, in Recent Advances in Nonsmooth Optimization, D.-Z. Du, L. Qi, and R. Womersley, eds., World Scientific Publishing, Singapore, 1995, pp. 88–105.

[4] M. Fukushima, *Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems*, Math. Programming, 53 (1992), pp. 99–110.

[5] C. C. Gonzaga, *The Largest Step Path Following Algorithm for Monotone Linear Complementarity Problems*, Math. Programming, to appear.

[6] P. Harker and J.-S. Pang, *Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications*, Math. Programming, 48 (1990), pp. 161–220.

[7] C. Kanzow, *Some equation-based methods for the nonlinear complementarity problem*, Optim. Methods Software, 3 (1994), pp. 327–340.

[8] M. Kojima, N. Megiddo, and S. Mizuno, *A general framework of continuation methods for complementarity problems*, Math. Oper. Res., 18 (1993), pp. 945–963.

[9] M. Kojima, N. Megiddo, and T. Noma, *Homotopy continuation methods for nonlinear complementarity problems*, Math. Oper. Res., 16 (1991), pp. 754–774.

[10] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise, *A unified approach to interior point algorithms for linear complementarity problems*, Lecture Notes in Computer Science 538, Springer-Verlag, Berlin, New York, 1991.

[11] M. Kojima, T. Noma, and A. Yoshise, *Global convergence in infeasible-interior-point algorithms*, Math. Programming, 65 (1994), pp. 43–72.

[12] G. M. Korpelevich, *The extragradient method for finding saddle points and other problems*, Matecon, 12 (1976), pp. 747–756.

[13] L. McLinden, *The complementarity problem for maximal monotone multifunctions*, in Variational Inequalities and Complementarity Problems: Theory and Applications, R. W. Cottle, F. Giannessi, and J.-L. Lions, eds., John Wiley & Sons, New York, NY, 1980, pp. 251–270.

[14] K. A. McShane, *Superlinearly convergent $O(\sqrt{n}L)$-iteration interior point algorithms for linear programming and the monotone linear complementarity problem*, SIAM J. Optim., 4 (1994), pp. 247–261.

[15] O. L. Mangasarian, *Equivalence of the complementarity problem to a system of nonlinear equations*, SIAM J. Appl. Math., 31 (1976), pp. 89–92.

[16] O. L. Mangasarian and M. V. Solodov, *Nonlinear complementarity as unconstrained and constrained minimization*, Math. Programming, 62 (1993), pp. 277–297.

[17] R. D. C. Monteiro and J.-S. Pang, *Properties of an interior-point mapping for mixed complementarity problems*, Math. Oper. Res., 21 (1996), pp. 629–654.

[18] R. D. C. Monteiro, J.-S. Pang, and T. Wang, *A positive algorithm for the nonlinear complementarity problem*, SIAM J. Optim., 5 (1995), pp. 129–148.

[19] J. J. Moré, *Global methods for nonlinear complementarity problems*, Math. Oper. Res., 21 (1996), pp. 589–614.

[20] J.-S. Pang and S. A. Gabriel, *NE/SQP: A robust algorithm for the nonlinear complementarity problem*, Math. Programming, 60 (1993), pp. 295–337.

[21] J.-S. Pang, *Complementarity problems*, in Handbook on Global Optimization, R. Horst and P. Pardalos, eds., Kluwer Academic Publishers, Norwell, MA, 1995, pp. 271–338.

[22] F. A. Potra and Y. Ye, *Interior point methods for nonlinear complementarity problems*, J. Optim. Theory Appl., 88 (1996), pp. 617–647.

[23] M. V. Solodov and P. Tseng, *Modified projection-type methods for monotone variational inequalities*, SIAM J. Control Optim., 34 (1996), pp. 1814–1830.

[24] P. K. Subramanian, *Gauss-Newton methods for the complementarity problem*, J. Optim. Theory Appl., 77 (1993), pp. 467–482.

[25] P. Tseng, *Global linear convergence of a path-following algorithm for some monotone variational inequality problems*, J. Optim. Theory Appl., 75 (1992), pp. 265–279.

[26] P. Tseng, *On linear convergence of iterative methods for the variational inequality problem*, J. Comput. Appl. Math., 60 (1995), pp. 237–252.

[27] T. Wang, R. D. C. Monteiro, and J.-S. Pang, *An interior point potential reduction method for constrained equations*, Math. Programming, 74 (1996), pp. 159–195.

[28] L. T. Watson, *Solving the nonlinear complementarity problem by a homotopy method*, SIAM J. Control Optim., 17 (1979), pp. 36–46.

[29] S. J. Wright and D. Ralph, *A superlinear infeasible-interior-point algorithm for monotone complementarity problems*, Math. Oper. Res., 21 (1996), pp. 815–838.

SIAM J. Optim.
Vol. 7, No. 2, pp. 403–420, May 1997

SMOOTH APPROXIMATIONS TO NONLINEAR COMPLEMENTARITY PROBLEMS[*]

BINTONG CHEN[†] AND PATRICK T. HARKER[‡]

Abstract. It is well known that a nonlinear complementarity problem (NCP) can be formulated as a system of nonsmooth equations. Chen and Mangasarian [*Comput. Optim. Appl.*, 5 (1996), pp. 97–138] proposed a class of parametric smooth functions by twice integrating a probability density function. As a result, the nonsmooth equations can be approximated by smooth equations. This paper refines the smooth functions proposed by Chen and Mangasarian and investigates their structural properties. The refinement allows us to establish the existence, uniqueness, and limiting properties of the trajectory defined by the solutions of these smooth equation approximations. In addition, global error bounds for the NCP with a uniform P-function are obtained.

Key words. nonlinear complementarity problem, smooth approximation, error bound, continuation method

AMS subject classifications. 90C33

PII. S1052623495280615

1. Introduction.

The nonlinear complementarity problem (NCP) is one of the fundamental problems of mathematical programming. In particular, the Karush–Kuhn–Tucker optimality conditions of any continuous optimization problem can be formulated as an NCP; see [4, 13] for a review of the literature in this area. Given a mapping $\mathbf{f} : R^n \to R^n$, the NCP with respect to $\mathbf{f}$, *NCP*[$\mathbf{f}$], finds an $\mathbf{x} \in R^n$ such that

$$\mathbf{x} \geq \mathbf{0}, \quad \mathbf{f}(\mathbf{x}) \geq \mathbf{0}, \quad \text{and} \quad \mathbf{x}^T\mathbf{f}(\mathbf{x}) = 0.$$

It is well known that an $\mathbf{x} \in R^n$ solves *NCP*[$\mathbf{f}$] if and only if it solves the following nonsmooth equations:

$$\mathbf{x} - (\mathbf{x} - \mathbf{f}(\mathbf{x}))_+ = \mathbf{0},$$

where the plus function $(\cdot)$ is defined by

$$(z)_+ = \max\{0, z\}.$$

In a recent paper, Chen and Mangasarian [3] proposed a class of parametric smooth functions that approximate the plus function by the double integration of a probability distribution function d that is defined by a smoothing parameter. This formulation leads to a class of smooth parametric nonlinear equation approximations to *NCP*[$\mathbf{f}$] and other complementarity problems. Using these approximations, a continuation method can be constructed to solve *NCP*[$\mathbf{f}$] that systematically solves the smooth equations and reduces the smoothing parameter to zero. Using this framework, many continuation methods that have been developed to date for *NCP*[$\mathbf{f}$] can be regarded as special cases of Chen and Mangasarian's smooth approximations; they

[*] Received by the editors January 30, 1995; accepted for publication (in revised form) January 3, 1996.

http://www.siam.org/journals/siopt/7-2/28061.html

[†] Department of Management and Systems, College of Business and Economics, Washington State University, Pullman, WA 99164-4736 (chenbi@wsu.edu).

[‡] Department of Systems Engineering, School of Engineering and Science, University of Pennsylvania, Philadelphia, PA 19104-6315 (harker@eniac.seas.upenn.edu). This author was supported by a grant from the Alfred P. Sloan Foundation.

vary in the choice of the particular form of the function d. Indeed, various interior point path-following algorithms for NCPs and other problems fit within this framework.

An important issue of using such a smooth approximation method to solve $NCP[\mathbf{f}]$ is the error that results from the approximation to the plus function as well as the computational efficiency of the proposed method. Chen and Mangasarian present an error bound for the strongly monotone NCP and report encouraging computational results based on their smooth approximations.

To ensure the success of continuation methods based on smooth approximations to $NCP[\mathbf{f}]$, one needs to investigate the trajectory consisting of solutions to the approximation subproblems as the smoothing parameter approaches zero. The trajectory of the interior point algorithm (using a particular smooth approximation) has been thoroughly studied in the literature. In particular, the paper by Kojima, Mizuno, and Noma [7] unified various interior point algorithms for NCPs and linear complementarity problems (LCPs); they provide a complete characterization of the trajectory that leads to a solution and show global and local convergence of the continuation method.

This paper refines the smooth functions introduced by Chen and Mangasarian [3] based on a set of structural properties that are necessary to establish various results on the trajectory of solutions. Using the techniques developed in Kojima, Mizuno, and Noma [7], this paper establishes the existence, uniqueness, and continuity of the trajectory that leads to a solution of $NCP[\mathbf{f}]$ under the assumption that the function $\mathbf{f}$ in $NCP[\mathbf{f}]$ satisfies both P_0- and R_0-properties. Finally, a global error bound for the NCP with a uniform P-function is obtained.

The following notation will be used throughout the paper. All vectors (vector functions) are column vectors (vector functions) and are denoted by boldface letters; $\mathbf{0}$ and $\mathbf{1}$ represent vectors of appropriate dimension with all components equal to 0 and 1, respectively. R^n, R^n_+, R^n_{++} denote, respectively, n-dimensional Euclidean space, the nonnegative orthant of R^n, and the strictly positive orthant of R^n. Given an $NCP[\mathbf{f}]$, $S_+[\mathbf{f}]$ represents the set of all feasible solutions, and $S_{++}[\mathbf{f}]$ represents the set of all strictly positive feasible solutions:

$$\begin{aligned} S_+[\mathbf{f}] &= \{\mathbf{x} \in R^n_+ : \mathbf{f}(\mathbf{x}) \geq \mathbf{0}\}, \\ S_{++}[\mathbf{f}] &= \{\mathbf{x} \in R^n_{++} : \mathbf{f}(\mathbf{x}) > \mathbf{0}\}. \end{aligned}$$

In addition, the following definitions related to function $\mathbf{f}$ will be used in the paper.

DEFINITION 1.1. *Let S be a nonempty subset of R^n. The mapping $\mathbf{f} : R^n \to R^n$ is said to be a*

1. *P_0-function over the set S if there is an index i such that*

$$x_i - y_i \neq 0 \textit{ and } [f_i(\mathbf{x}) - f_i(\mathbf{y})](x_i - y_i) \geq 0 \textit{ for all } \mathbf{x}, \mathbf{y} \in S \textit{ and } \mathbf{x} \neq \mathbf{y},$$

2. *uniform P-function over the set S if, for some $\gamma > 0$,*

$$\max_{1 \leq i \leq n} [f_i(\mathbf{x}) - f_i(\mathbf{y})](x_i - y_i) \geq \gamma \|\mathbf{x} - \mathbf{y}\|^2 \textit{ for all } \mathbf{x}, \mathbf{y} \in S,$$

3. *monotone function over the set S if*

$$[\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})]^T(\mathbf{x} - \mathbf{y}) \geq 0 \textit{ for all } \mathbf{x}, \mathbf{y} \in S,$$

4. *strongly monotone function over the set S if, for some $\gamma > 0$,*

$$[\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})]^T(\mathbf{x} - \mathbf{y}) \geq \gamma \|\mathbf{x} - \mathbf{y}\|^2 \textit{ for all } \mathbf{x}, \mathbf{y} \in S.$$

It is well known that strong monotonicity implies both the uniform P-property and monotonicity, which in turn imply the P_0-property.

DEFINITION 1.2. *Let S be a nonempty subset of R^n and $\mathbf{f}$ be a differentiable function. The Jacobian matrix $\nabla\mathbf{f}(\mathbf{x})$ of $\mathbf{f}$ is Lipschitz continuous over S if, for some $L > 0$,*

$$\|\nabla\mathbf{f}(\mathbf{x}) - \nabla\mathbf{f}(\mathbf{y})\| \le L\|\mathbf{x} - \mathbf{y}\| \textit{ for all } \mathbf{x}, \mathbf{y} \in S,$$

where $\|\mathbf{A}\|$ denotes the matrix norm $\max\{\|\mathbf{Aw}\| : \mathbf{w} \in \mathbf{R^n}, \|\mathbf{w}\| = \mathbf{1}\}$ for every $n \times n$ matrix $\mathbf{A}$.

2. Smooth approximation of $(z)_+$. This section refines the class of smooth functions proposed in [3] to approximate the fundamental plus function $(z)_+$ based on the structural properties that are needed to establish the theoretical properties on the trajectory of solutions in section 3.

Define the approximation function p to be a continuous mapping from $R \times R_+$ to R that satisfies the following assumption:

(A1'): $|p(z,u) - (z)_+| \le b(u)$ for all z and $u \ge 0$, where the function $b : R_+ \to R_+$ is continuous, strictly increasing, and satisfies $b(0) = 0$ and $\lim_{u\to\infty} b(u) = \infty$.

The function $p(z,u)$ is a reasonable approximation of $(z)_+$ since, by assumption (A1'),

$$\lim_{u\to+0} p(z,u) = (z)_+ \text{ for all } z \in R.$$

The variable u is called the parameter of the approximation function p, or the smoothing parameter. This approximation will now be refined using the following additional assumptions:

(A2'): $p(z,u)$ is convex with respect to z.

(A2): $p(z,u)$ is strictly convex with respect to z.

With these additional assumptions, the function p has the following properties.

PROPOSITION 2.1. *Let $p(z,u)$ be a differentiable function with respect to z for all $u > 0$.*

1. *Assumptions* (A1') *and* (A2') *imply that $0 \le p'(z,u) \le 1$ for all z and $\infty > u > 0$.*

2. *Assumptions* (A1') *and* (A2) *imply that $0 < p'(z,u) < 1$ for all z and $\infty > u > 0$.*

Proof. Since the function b is continuous on R_+ given any $\infty > u > 0$, there exists a $B > 0$ such that $b(u) < B$. To show part 1, suppose on the contrary that $p'(z,u) = \alpha > 1$ for some z. By assumption (A2'), $p(z,u)$ is convex in x. Thus,

$$p(y,u) \ge p(z,u) + \alpha(y - z) \text{ for all } y \ge z.$$

It follows that

$$\begin{aligned}\lim_{y\to\infty} |p(y,u) - (y)_+| &\ge \lim_{y\to\infty} p(y,u) - (y)_+ \\ &\ge \lim_{y\to\infty} \alpha y - (y)_+ + p(z,u) - \alpha z \\ &= \lim_{y\to\infty} \alpha y - y + p(z,u) - \alpha z \\ &= \infty > B.\end{aligned}$$

However, this contradicts assumption (A1') since $b(u) < B$. Therefore, $p'(z,u) \le 1$. That $p'(z,u) \ge 0$ can be shown in a similar manner.

To show part (2), suppose on the contrary that $p'(z,u) \geq 1$ for some z. By assumption (A2), there exists a $y > z$ such that $p'(y,u) > 1$. However, this contradicts part 1 of the proposition. Therefore, $p'(z,u) < 1$. Similarly, one can show that $0 < p'(z,u)$. □

The above class of approximation functions $p(z,u)$ includes those proposed in [3] as special cases. Indeed, the approximation function in [3] is given by

$$q(z,u) = \int_{-\infty}^{z} \int_{-\infty}^{t} d(x,u)dxdt, \tag{1}$$

where $d(x,u) = \frac{1}{u}d(\frac{x}{u})$ and $d(x)$ is a probability distribution function satisfying certain technical assumptions. It has been shown [3] that the function q is continuously differentiable with respect to z, continuous with respect to u, and satisfies both assumptions (A1') and (A2') with $b(u) = Bu$ for some $B > 0$. In addition, if the function d has an infinite support R, then q also satisfies assumption (A2).

In order to continue to refine the approximation function p, one must modify the existing assumptions and add several new conditions:

(A1): $0 \leq p(z,u) - (z)_+ \leq b(u)$ for all z and $u \geq 0$, where function b is defined in assumption (A1').

(A3): $\lim_{z\to\infty}(p(z,u) + p(-z,u) - z)z < \infty$ for all $\infty > u > 0$.

(A4): For any fixed $z \in R$, $p(z,u)$ is strictly increasing in u and $\lim_{u\to\infty} p(z,u) = \infty$.

The above refinements will enable the development of results on the trajectory of the continuation method based on the smooth approximation p; these results will be presented in the next section. In particular, assumption (A3) will play a pivotal role. Essentially, this assumption states that as $z \to \infty$, the approximation function p approaches the plus function faster than z approaches infinity.

Using the above assumptions, the following structural results on the function p can be obtained.

Proposition 2.2. *Assumptions* (A1) *and* (A2) *imply that* $p(z,u) > (z)_+ \geq 0$ *for all* $z \in R$ *and* $u \in R_{++}$.

Proof. From assumption (A1), $p(z,u) \geq (z)_+$. It suffices to show that $p(z,u) \neq (z)_+$ for $u > 0$. Suppose, on the contrary, that $p(z,u) = (z)_+$ for some z. From part 2 of Proposition 2.1, $0 < p'(z,u) < 1$. Therefore, if $z \leq 0$ then $p(y,u) < (y)_+$ for all $y < z$. Similarly, if $z > 0$ then $p(y,u) < (y)_+$ for all $y > z$. In either case, there exists a y such that $p(y,u) - (y)_+ < 0$. However, this contradicts assumption (A1). Therefore, $p(z,u) \neq (z)_+$. □

The next result will be used in the sequel to show boundedness of the solution trajectory.

Proposition 2.3. *Suppose that the function* p *satisfies assumptions* (A1)–(A3) *and that* $x = p(x-y,u)$ *for some* $\infty > u > \ (\geq)\ 0$. *Then*

$$x > \ (\geq)\ 0, \quad y > \ (\geq)\ 0, \quad xy \leq B$$

for some $0 < \ (\leq)\ B < \infty$.

Proof. If $u = 0$ then

$$0 = x - p(x-y,u) = x - (x-y)_+ = \min\{x,y\}.$$

Thus, the result is obviously true. Suppose now that $u > 0$. From Proposition 2.2, one has $p(x-y,u) > (x-y)_+$. By assumption, $x = p(x-y,u)$. Therefore,

$$x > (x-y)_+ \geq 0 \text{ and } y \geq x - (x-y)_+ > 0.$$

It remains to show that $xy \le B < \infty$. From Proposition 2.2, one has $p(z,u) - z > 0$ and $p(-z,u) > 0$ for all z. Therefore, assumption (A3) is equivalent to

$$\lim_{z\to\infty} (p(z,u) - z)z < \infty \text{ and } \lim_{z\to\infty} p(-z,u)z < \infty,$$

which in turn implies

$$\lim_{z\to\infty} p(z,u) - z = 0 \text{ and } \lim_{z\to\infty} p(-z,u) = 0.$$

To show that xy is bounded, consider the following three cases:

1. $|x-y|$ is bounded: since $x = p(x-y,u)$, both x and y must be bounded and, thus, so must xy.

2. $x - y \to \infty$: this implies $x \to \infty$ and $y = p(x-y,u) - (x-y) \to 0$. It follows that

$$\lim_{x\to\infty} xy = \lim_{x\to\infty} y(x-y) = \lim_{x\to\infty} [p(x-y,u) - (x-y)](x-y) = \lim_{z\to\infty} (p(z,u) - z)z < \infty.$$

3. $x - y \to -\infty$: then $y \to \infty$ and $x = p(x-y,u) \to 0$. It follows that

$$\lim_{y\to-\infty} xy = \lim_{y\to-\infty} x(y-x) = \lim_{y\to-\infty} p(x-y,u)(y-x) = \lim_{z\to\infty} p(-z,u)z < \infty. \quad \square$$

The next result will be used in section 3 to show the uniqueness of the solution trajectory.

PROPOSITION 2.4. *If the function p satisfies assumptions* (A1) *and* (A4), *then given any $x \ge$ (>) 0 and $y \ge$ (>) 0, there exists a unique $u \ge$ (>) 0 such that $x = p(x-y,u)$.*

Proof. By assumption (A1), one has

$$\lim_{u\to+0} p(x-y,u) = (x-y)_+ \le \ (<)\ x$$

since $x \ge$ (>) 0 and $y \ge$ (>) 0. By assumption (A4), one has

$$\lim_{u\to\infty} p(x-y,u) = \infty > x.$$

Since p is continuous in u, there exists a $0 \le$ (<) $u < \infty$ such that $x = p(x-y,u)$. In addition, u is unique, since p is strictly increasing in u by assumption (A4). $\square$

Below are two examples of approximation functions that satisfy assumptions (A1)–(A4).

Example 1. Neural network smoothing function [2].

$$p(z,u) = z + u\log(1 + e^{-\frac{z}{u}}),$$

where $b(u) = (\log 2)u$.

Example 2. Interior point smoothing function.

$$p(z,u) = \frac{z + \sqrt{z^2 + 4u}}{2},$$

where $b(u) = \sqrt{u}$. To see how this approximation is related to interior point algorithms, consider the equation $x = p(x-y,u)$; i.e.,

$$x = \frac{x - y + \sqrt{(x-y)^2 + 4u}}{2}.$$

It has been shown [1, 5] that x and y solve the above equation if and only if they solve the following system:

$$x > 0,\ y > 0,\ xy = u,$$

which is the fundamental approximation to complementarity conditions used in interior point algorithms.

The obvious question is whether there exists a class of functions $p(z, u)$ that are not equivalent to the functions $q(z, u)$ defined in (1) that still satisfy assumptions (A1)–(A4) and, hence, possess the desirable structural properties given in Propositions 2.1–2.4. As the next result shows, such a broad class of functions does not exist. Thus, the class of functions introduced by Chen and Mangasarian [3] are quite central in proving the convergence of algorithms for the *NCP*[**f**] based upon these smooth approximations.

PROPOSITION 2.5. *If the function $p(z, u)$ satisfies assumptions* (A1)–(A3) *and is differentiable with respect to z, then*

$$p(z, u) = \int_{-\infty}^{z} \int_{-\infty}^{t} d(x, u) dx dt$$

for some probability distribution function d.

Proof. By part 2 of Proposition 2.1, $0 < p'(z, u) < 1$ for all z and $u > 0$. It follows that

$$\lim_{z \to -\infty} p'(z, u) = 0, \qquad \lim_{z \to +\infty} p'(z, u) = 1;$$

otherwise, one would have $p(z, u) < (z)_+$ for some z, which contradicts Proposition 2.2. Therefore, $p'(z, u)$ is a cumulative distribution function defined by

$$p'(t, u) = \int_{-\infty}^{t} d(x, u) dx$$

for some probability distribution function d.

Thus,

$$p(z, u) = \int_{-\infty}^{z} \int_{-\infty}^{t} d(x, u) dx dt + \alpha$$

for some $\alpha \in R$. It suffices to show that $\alpha = 0$. Assumptions (A1) and (A3) imply (see, for example, the proof of Proposition 2.3) that

$$\lim_{z \to -\infty} p(z, u) = 0$$

or

$$\lim_{z \to -\infty} \int_{-\infty}^{z} \int_{-\infty}^{t} d(x, u) dx dt + \alpha = 0.$$

Since the first term must equal zero, one has $\alpha = 0$. $\square$

Thus, there exist many valid approximations to $(z)_+$. However, an approximation that will possess desirable algorithmic properties will most likely need to be a subset of functions defined as the double integration of the probability density function d.

Indeed, we can characterize a subset of density functions whose double integrations satisfy assumptions (A1)–(A4); to do this, we need the following results from [3].

Proposition 2.6. *Let $d(x)$ be a probability density function that satisfies the following conditions:*

(C1): *$d(x)$ is piecewise continuous with a finite number of pieces.*

(C4'): *$\int_{-\infty}^{+\infty} |x|d(x)dx < +\infty$.*

Then the function $q(x, u)$, defined by (1)*, has the following properties:*

1. *$q(x, u) = \int_{-\infty}^{z} (z - x)d(x, u)dx$;*
2. *$-D_2u \le q(z, u) - (z)_+ \le D_1u$, where*

$$D_1 = \int_{-\infty}^{0} |x|d(x)dx \quad \textit{and} \quad D_2 = \max\left\{\int_{-\infty}^{+\infty} xd(x)dx, 0\right\};$$

3. *$q(z, u)$ is strictly increasing and strictly convex with respect to z if the following condition also holds:*

(C2): *$d(x)$ has infinite support; i.e., supp$\{d(x)\} = R$.*

The next result characterizes a subset of density functions whose double integrations satisfy assumptions (A1)–(A4).

Proposition 2.7. *Let $d(x)$ be a probability density function that satisfies conditions* (C1), (C2), *and*

(C3): $\int_{-\infty}^{+\infty} xd(x)dx = 0$;

(C4): $\lim_{x\to+\infty} x^3d(x) < +\infty$ *and* $\lim_{x\to+\infty} x^3d(-x) < +\infty$.

Then $q(x, u)$ satisfies assumptions (A1)–(A4).

Proof. Clearly, condition (C4) implies (C4'). Then condition (C3) and part 2 of Proposition 2.6 imply that $q(z, u)$ satisfies assumption (A1) with $b(u) = D_1u$. That $q(z, u)$ satisfies assumption (A2) follows directly from part (3) of Proposition 2.6. To show that $q(z, u)$ satisfies assumption (A3), we start by proving that $\lim_{z\to+\infty} q(z, u) - z = 0$. Indeed,

$$\begin{aligned}
\lim_{z\to+\infty} q(z, u) - z &= \lim_{z\to+\infty} \int_{-\infty}^{z} (z - x)d(x, u)dx - z \\
&= \lim_{z\to+\infty} z\left(\int_{-\infty}^{z} d(x, u)dx - 1\right) - \lim_{z\to+\infty} \int_{-\infty}^{z} \frac{x}{u} d\left(\frac{x}{u}\right) dx \\
&= -\lim_{z\to+\infty} z^2d(z, u) - u \lim_{y\to+\infty} \int_{-\infty}^{y} xd(x)dx \\
&= -u \lim_{x\to+\infty} x^2d(x) \\
&= 0,
\end{aligned}$$

where the first equality follows part 1 of Proposition 2.6, the fourth equality follows condition (C3), and the fifth equality follows condition (C4). Therefore, the limit of $z(q(z, u) - z)$ can be evaluated by applying l'Hôpital's rule:

$$\begin{aligned}
\lim_{z\to+\infty} z(q(z, u) - z) &= \lim_{z\to+\infty} z\left(\int_{-\infty}^{z}\int_{-\infty}^{t} d(x, u)dxdt - z\right) \\
&= \lim_{z\to+\infty} z^2\left(1 - \int_{-\infty}^{z} d(x, u)dx\right) \\
&= \lim_{z\to+\infty} \frac{1}{2} z^3d(z, u)
\end{aligned}$$

$$\begin{aligned}&= \frac{u^2}{2} \lim_{x\to+\infty} x^3 d(x)\\&< +\infty.\end{aligned}$$

Similarly, one can also show that

$$\lim_{z\to+\infty} zq(-z,u) < +\infty.$$

Therefore, $q(z,u)$ satisfies assumption (A3). To show that $q(z,u)$ satisfies assumption (A4), we evaluate the derivative of $q(z,u)$ with respect to u:

$$\frac{dq(z,u)}{du} = -\int_{-\infty}^{\frac{z}{u}} xd(x)dx > -\int_{-\infty}^{+\infty} xd(x)dx = 0,$$

where the first equality is by direct evaluation, the inequality follows condition (C2), and the second equality follows condition (C3). It follows that $q(z,u)$ is strictly increasing with respect to u. In addition, notice that

$$\begin{aligned}\lim_{u\to+\infty} q(z,u) &= \lim_{u\to+\infty} \int_{-\infty}^{z} (z-x)d(x,u)dx\\&= \lim_{u\to+\infty}\left(z\int_{-\infty}^{\frac{z}{u}} d(x)dx - u\int_{-\infty}^{\frac{z}{u}} xd(x)dx\right)\\&= z\int_{-\infty}^{0} d(x)dx + u\left(-\int_{-\infty}^{\frac{z}{u}} xd(x)dx\right)\\&= +\infty,\end{aligned}$$

where the last equality follows from conditions (C2) and (C3). Therefore, $q(z,u)$ satisfies assumption (A4). □

3. Smooth approximation algorithms for the NCP. In this section, the smooth approximation functions developed in the previous section will be applied to NCPs. In particular, this section will investigate the existence, uniqueness, and continuity of the trajectory consisting of solutions of the smooth equation approximations for different parameters and establish the error bounds under the assumption that $\mathbf{f}$ is a uniform P-function.

As mentioned in the introduction, $\mathbf{x}$ solves $NCP[\mathbf{f}]$ if and only if it solves the following nonsmooth equations:

$$\mathbf{R}(\mathbf{x},\mathbf{0}) = \mathbf{x} - (\mathbf{x}-\mathbf{f}(\mathbf{x}))_+ = \mathbf{0}.$$

Using the smooth approximation functions developed in the previous section, one can approximate the nonsmooth equations as follows:

$$\mathbf{R}(\mathbf{x},\mathbf{u}) = \mathbf{x} - \mathbf{p}(\mathbf{x}-\mathbf{f}(\mathbf{x}),\mathbf{u}) = \mathbf{0},$$

where $\mathbf{p}(\mathbf{x}-\mathbf{f}(\mathbf{x}),\mathbf{u})$ is a column vector with components $p(x_i - f_i(\mathbf{x}), u_i)$, $i = 1,\dots,n$. Under assumption (A1'),

$$\lim_{\mathbf{u}\to\mathbf{0}} \mathbf{R}(\mathbf{x},\mathbf{u}) = \mathbf{R}(\mathbf{x},\mathbf{0})$$

for all $\mathbf{x} \in R^n$. Therefore, the smooth approximation becomes more accurate as the parameter $\mathbf{u}$ approaches $\mathbf{0}$. It will be assumed throughout this section that the approximation function p satisfies assumptions (A1)–(A4) and, thus, satisfies (1) for some probability distribution function d (although some results are true based on only a portion of these assumptions).

3.1. Solution of $\mathbf{R}(\mathbf{x},\mathbf{u}) = \mathbf{a}$. This subsection investigates the properties of the solution to equation $\mathbf{R}(\mathbf{x},\mathbf{u}) = \mathbf{a}$ as the parameters $\mathbf{u}$ and $\mathbf{a}$ vary. The existence, uniqueness, and continuity properties of the solution will be established. To begin, consider the solution to equation $\mathbf{R}(\mathbf{x},\mathbf{u}) = \mathbf{0}$.

Proposition 3.1. *The following statements are true:*

1. *Let* $\mathbf{x}$ *be a solution of* $\mathbf{R}(\mathbf{x},\mathbf{u}) = \mathbf{0}$ *for some* $\mathbf{u} \in R^n_{++}$ (R^n_+) *. Then* $\mathbf{x} \in S_{++}[\mathbf{f}]$ $(S_+[\mathbf{f}])$ *and* $\mathbf{x}^T\mathbf{f}(\mathbf{x}) \leq B$ *for some* $0 <$ $(\leq)$ $B < \infty$*.*

2. *For any* $\mathbf{x} \in S_{++}[\mathbf{f}]$ $(S_+[\mathbf{f}])$ *, there exists a unique* $\mathbf{u} \in R^n_{++}$ (R^n_+) *such that* $\mathbf{R}(\mathbf{x},\mathbf{u}) = \mathbf{0}$.

Proof. Since $\mathbf{x}$ is a solution of $\mathbf{R}(\mathbf{x},\mathbf{u}) = \mathbf{0}$, it follows that $x_i = p(x_i - f_i(\mathbf{x}), u_i)$ for all $i = 1, \ldots, n$. Part 1 then follows from Proposition 2.3. For part 2, since $\mathbf{x} \in S_{++}[\mathbf{f}]$ $(S_+[\mathbf{f}])$, it follows that $x_i > (\geq)\ 0$ and $f_i(\mathbf{x}) > \ (\geq)\ 0$ for all $i = 1, \ldots, n$. By Proposition 2.4, there exists a unique $u_i > \ (\geq)\ 0$ such that $x_i = p(x_i - f_i(\mathbf{x}), u_i)$ for each i. □

Now consider the uniqueness of the solution to equation $\mathbf{R}(\mathbf{x},\mathbf{u}) = \mathbf{a}$ under the following assumption:

(B1): $\mathbf{f}$ is a P_0-function on R^n.

Proposition 3.2. *For any* $\mathbf{u} \in R^n_{++}$ *and* $\mathbf{a} \in R^n$*, the solution of* $\mathbf{R}(\mathbf{x},\mathbf{u}) = \mathbf{a}$ *is unique if* $\mathbf{f}$ *satisfies assumption* (B1)*.*

Proof. Suppose, on the contrary, that $\mathbf{R}(\mathbf{x}^1,\mathbf{u}) = \mathbf{R}(\mathbf{x}^2,\mathbf{u})$ and $\mathbf{x}^1 \neq \mathbf{x}^2$ for some $\mathbf{u} \in R^n_{++}$. Since $\mathbf{f}$ is a P_0-function, there exists an index i such that

$$x^1_i \neq x^2_i \text{ and } (f_i(\mathbf{x}^1) - f_i(\mathbf{x}^2))(x^1_i - x^2_i) \geq 0. \tag{2}$$

Assume without loss of generality that $x^1_i > x^2_i$. Then the inequality in (2) implies that $f_i(\mathbf{x}^1) \geq f_i(\mathbf{x}^2)$. By assumption, $\mathbf{R}(\mathbf{x}^1,\mathbf{u}) = \mathbf{R}(\mathbf{x}^2,\mathbf{u})$. It follows that

$$\begin{aligned} x^1_i - x^2_i &= p(x^1_i - f_i(\mathbf{x}^1), u_i) - p(x^2_i - f_i(\mathbf{x}^2), u_i) \\ &\leq p(x^1_i - f_i(\mathbf{x}^2), u_i) - p(x^2_i - f_i(\mathbf{x}^2), u_i) \\ &\leq p'(x^1_i - f_i(\mathbf{x}^2), u_i)(x^1_i - x^2_i) \\ &< x^1_i - x^2_i, \end{aligned}$$

where the first inequality is true because $p(z,u)$ is an increasing function in z, the second inequality is true because $p(z,u)$ is strictly convex in z, and the third inequality is true because $0 < p'(z,u) < 1$ for all z and $u > 0$ and $x^1_i > x^2_i$ by assumption. However, this leads to a contradiction and hence, $\mathbf{x}$ is unique. □

For ease of exposition, define two mappings; mapping $\mathbf{G} : S_+[\mathbf{f}] \to R^n_+$ relates the solution of $\mathbf{R}(\mathbf{x},\mathbf{u}) = \mathbf{0}$ and the parameter $\mathbf{u}$ as follows:

$$\mathbf{G}(\mathbf{x}) = \{\mathbf{u} \in R^n_+ : \mathbf{R}(\mathbf{x},\mathbf{u}) = \mathbf{0}, \mathbf{x} \in S_+[\mathbf{f}]\}.$$

Mapping $\mathbf{G}$ is well defined by part 2 of Proposition 3.1 and is continuous because $\mathbf{R}$ is continuous. Mapping $\mathbf{R_u} : R^n \to R^n$ relates the solution of $\mathbf{R}(\mathbf{x},\mathbf{u}) = \mathbf{a}$ and the right-hand side $\mathbf{a}$ for a given $\mathbf{u} \in R^n_{++}$ as follows:

$$\mathbf{R_u}(\mathbf{x}) = \{\mathbf{a} \in R^n : \mathbf{R}(\mathbf{x},\mathbf{u}) = \mathbf{a}, \mathbf{x} \in R^n\}.$$

Clearly, for each $\mathbf{x} \in R^n$, $\mathbf{a} = \mathbf{R}(\mathbf{x},\mathbf{u})$ is uniquely determined. Based on Proposition 3.1 and Proposition 3.2, the following result is obtained.

Corollary 3.3. *Under assumption* (B1)*,*

1. *the mapping* $\mathbf{G}$ *is one-to-one between* $S_{++}[\mathbf{f}]$ *and* R^n_{++}*;*

2. *for any given* $\mathbf{u} \in R^n_{++}$*, the mapping* $\mathbf{R_u}$ *is one-to-one between* R^n *and* R^n*.*

To ensure that the equation $\mathbf{R}(\mathbf{x}, \mathbf{u}) = \mathbf{0}$ has a solution, two additional conditions on the function $\mathbf{f}$ must be assumed. Similar assumptions have been used in [7].

(B2): $S_{++}[\mathbf{f}] \neq \emptyset$; i.e., there exists an $\mathbf{x} > \mathbf{0}$ such that $\mathbf{f}(\mathbf{x}) > \mathbf{0}$.

(B3): The set $\mathbf{G}^{-1}(U) = \{\mathbf{x} \in R^n_+ : \mathbf{R}(\mathbf{x}, \mathbf{u}) = \mathbf{0}, \mathbf{u} \in U\}$ is bounded for every compact subset U of R^n_+.

To ensure that the equation $\mathbf{R}(\mathbf{x}, \mathbf{u}) = \mathbf{a}$ has a solution for any $\mathbf{u} \in R^n_+$ and $\mathbf{a} \in R^n$, a stronger assumption than (B3) is required:

(B4): The set $\mathbf{R}^{-1}_{\mathbf{u}}(L) = \{\mathbf{x} \in R^n : \mathbf{R}(\mathbf{x}, \mathbf{u}) = \mathbf{a}, \mathbf{a} \in L\}$ is bounded for every compact subset L of R^n and $\mathbf{u} \in R^n_+$.

Clearly, assumption (B4) implies assumption (B3) by setting $L = \{\mathbf{a}\} = \{\mathbf{0}\}$. The conditions under which assumptions (B1)–(B4) are satisfied will be discussed at the end of the subsection.

Applying the basic methodology of Kojima, Mizumo, and Noma [7], one can establish the existence of solutions to these systems of equations.

PROPOSITION 3.4. *The following statements are true:*

1. *Under assumptions* (B1)–(B3)*, the system of equations* $\mathbf{R}(\mathbf{x}, \mathbf{u}) = \mathbf{0}$ *has a solution for every* $\mathbf{u} \in R^n_+$*.*

2. *Under assumptions* (B1)–(B4)*, the systems of equations* $\mathbf{R}(\mathbf{x}, \mathbf{u}) = \mathbf{a}$ *have a solution for every* $\mathbf{u} \in R^n_+$ *and* $\mathbf{a} \in R^n$*.*

Proof. From assumption (B2), there exists an $\hat{\mathbf{x}} > 0$ such that $\mathbf{f}(\hat{\mathbf{x}}) > \mathbf{0}$. By Proposition 3.1, there exists a $\hat{\mathbf{u}} > 0$ such that $\mathbf{R}(\hat{\mathbf{x}}, \hat{\mathbf{u}}) = \mathbf{0}$. Now consider the family of equations with parameter $t \in [0, 1]$:

$$\mathbf{R}(\mathbf{x}, (1-t)\hat{\mathbf{u}} + t\mathbf{u}) = \mathbf{0}. \tag{3}$$

Let $\bar{t} \leq 1$ be the supremum of t's such that equation (3) has a solution for every $t \in [0, \bar{t}]$. Then there exists a sequence $\{(\mathbf{x}^k, t^k)\}$ of solutions of equation (3) such that $\lim_{k\to\infty} t^k = \bar{t}$. Since the parameter $(1-t)\hat{\mathbf{u}} + t\mathbf{u}$ lies in the compact convex subset $U = \{(1-t)\hat{\mathbf{u}} + t\mathbf{u} : t \in [0, 1]\}$ of R^n_+ for all $t \in [0, 1]$, assumption (B3) ensures that the sequence $\{x^k\}$ is bounded. Hence, one may assume that it converges to some $\bar{\mathbf{x}}$. Since the function $\mathbf{R}$ is continuous in both x and u, it must be the case that

$$\mathbf{R}(\bar{\mathbf{x}}, (1-\bar{t})\hat{\mathbf{u}} + \bar{t}\mathbf{u}) = \mathbf{0} \text{ or } \mathbf{G}(\bar{\mathbf{x}}) = (1-\bar{t})\hat{\mathbf{u}} + \bar{t}\mathbf{u}.$$

Hence, if $\bar{t} = 1$, the desired result follows. Assume, on the contrary, that $\bar{t} < 1$. By Corollary 3.3, the mapping $\mathbf{G}$ between the solution of equation (3) and the parameter $\mathbf{u}$ is a local homeomorphism at $\bar{\mathbf{x}}$ (see, for example, the domain invariance theorem in Schwartz [15]). Hence, $\mathbf{G}(\mathbf{x}) = (1-t)\hat{\mathbf{u}} + t\mathbf{u}$ or equation (3) has a solution for every t sufficiently close to $\bar{t}$. This contradicts the definition of $\bar{t}$.

The proof of part 2 consists of two steps. First, it is shown that for any given $\mathbf{u} \in R^n_{++}$, equation $\mathbf{R}(\mathbf{x}, \mathbf{u}) = \mathbf{a}$ has a solution for every $\mathbf{a} \in R^n$. It is then shown that for any given $\mathbf{a} \in R^n$, equation $\mathbf{R}(\mathbf{x}, \mathbf{u}) = \mathbf{a}$ has a solution for every $\mathbf{u} \in R^n_+$.

Let $\mathbf{u}$ be any given parameter in R^n_{++} and $\hat{\mathbf{x}}$ be the solution of equation $\mathbf{R}(\mathbf{x}, \mathbf{u}) = \mathbf{0}$, which exists following part 1 of this result. Consider the family of equations with parameter $t \in [0, 1]$:

$$\mathbf{R}(\mathbf{x}, \mathbf{u}) = t\mathbf{a}. \tag{4}$$

Let $\bar{t} \leq 1$ be the supremum of t's such that equation (4) has a solution for every $t \in [0, \bar{t}]$. Then there exists a sequence $\{(\mathbf{x}^k, t^k)\}$ of solutions of equation (4) such that $\lim_{k\to\infty} t^k = \bar{t}$. Following essentially the same proof as in part 1, together with assumption (B4) and Proposition 3.2, one has $\lim \mathbf{x}^k = \bar{\mathbf{x}}$ and $\bar{t} = 1$. Therefore, equation $\mathbf{R}(\mathbf{x}, \mathbf{u}) = \mathbf{a}$ has a solution for every $\mathbf{a} \in R^n$ and $\mathbf{u} \in R^n_{++}$.

Now let $\mathbf{a} \in R^n$ be any given right-hand side. Using the above result, there exist a $\hat{\mathbf{u}} \in R^n_{++}$ and $\hat{\mathbf{x}} \in R^n$ such that $\mathbf{R}(\hat{\mathbf{x}}, \hat{\mathbf{u}}) = \mathbf{a}$. Then follow the same proof as for part 1 of this result, together with assumption (B4); it can be shown that equation $\mathbf{R}(\mathbf{x}, \mathbf{u}) = \mathbf{a}$ has a solution for every $\mathbf{u} \in R^n_+$. Since $\mathbf{a} \in R^n$ is chosen arbitrarily, part 2 is proven. □

COROLLARY 3.5. *The following statements are true:*

1. *under assumptions* (B1)–(B3), $\mathbf{G}$ *maps* $S_{++}[\mathbf{f}]$ *onto* R^n_{++} *homeomorphically*;

2. *under assumptions* (B1)–(B4), $\mathbf{R}_\mathbf{u}$ *maps* R^n *onto* R^n *homeomorphically for any* $\mathbf{u} \in R^n_{++}$.

Proof. Since the proofs for both statements are essentially the same, the proof of only part 1 will be stated herein. By Proposition 3.1, $\mathbf{G}(S_{++}[\mathbf{f}]) \subset R^n_{++}$. By Proposition 3.4, $R^n_{++} \subset \mathbf{G}(S_{++}[\mathbf{f}])$. Hence, $\mathbf{G}$ maps $S_{++}[\mathbf{f}]$ onto R^n_{++}. By Corollary 3.3, the continuous map $\mathbf{G}$ is one-to-one between two open subsets $S_{++}[\mathbf{f}]$ and R^n_{++}. The homeomorphism follows from the domain invariance theorem (see Schwartz [15]). □

Let $\mathbf{u} \in R^n_{++}$ and $t > 0$. Denote $\mathbf{x}(t)$ to be the solution of a family of equations $\mathbf{R}(\mathbf{x}, t\mathbf{u}) = t\mathbf{a}$. The next result states that the solution $\mathbf{x}(t)$ forms a trajectory with respect to parameter t and that it leads to a solution of *NCP*[$\mathbf{f}$] as t approaches 0.

THEOREM 3.6. *Let* $\mathbf{u} \in R^n_{++}$ *and* $U = \{t\mathbf{u} : t > 0\}$. *The following statements are true under assumptions* (B1)–(B4) *(assumptions* (B1)–(B3) *if* $\mathbf{a} = \mathbf{0}$*):*

1. *For every* $t > 0$, *equation* $\mathbf{R}(\mathbf{x}, t\mathbf{u}) = t\mathbf{a}$ *has a unique solution* $\mathbf{x}(t)$ *which is continuous in* t; *hence, the set* $\{\mathbf{x}(t) : t > 0\}$ *forms a trajectory.*

2. *For every* $t^0 > 0$, *the subtrajectory* $\{\mathbf{x}(t) : 0 < t < t^0\}$ *is bounded*; *hence, there is at least one limit point of* $\mathbf{x}(t)$ *as* $t \to 0$.

3. *Every limit point of* $\mathbf{x}(t)$ *as* $t \to 0$ *is a complementarity solution of NCP*[$\mathbf{f}$].

Proof. Since $t\mathbf{u} \in R^n_{++}$ for every $t > 0$, part 1 follows from Corollary 3.5. Notice that the parameter set

$$U = \{t\mathbf{u} : 0 < t < t^0\} \subset \{t\mathbf{u} : 0 \le t \le t^0\}$$

is a compact subset of R^n_+. Similarly, the right-hand set

$$L = \{t\mathbf{a} : 0 < t < t^0\} \subset \{t\mathbf{a} : 0 \le t \le t^0\}$$

is a compact subset of R^n. By assumption (B4), the set of solutions $\{\mathbf{x} \in R^n : \mathbf{R}(\mathbf{x}, t\mathbf{u}) = t\mathbf{a}, \mathbf{u} \in U, \mathbf{a} \in L\}$ is bounded. Thus, part (2) is established. By the continuity of $\mathbf{R}$, if $\mathbf{x}$ is a limiting point of $\mathbf{x}(t)$ as $t \to 0$, one has $\mathbf{R}(\mathbf{x}, \mathbf{0}) = \mathbf{0}$ or $\mathbf{x} - (\mathbf{x} - \mathbf{f}(\mathbf{x}))_+ = \mathbf{0}$; hence, $\mathbf{x}$ is a complementarity solution of the *NCP*[$\mathbf{f}$]. □

To complete the section, a set of conditions are provided under which assumptions (B1)–(B4) are satisfied.

PROPOSITION 3.7. *Assumptions* (B1)–(B3) *are satisfied if* $\mathbf{f}$ *is a monotone function and the set* $S_{++}[\mathbf{f}]$ *is nonempty.*

Proof. Assumptions (B1) and (B2) are satisfied since monotonicity implies the P_0-property. Given any $\mathbf{u} \in R^n_{++}$, let $\mathbf{x}$ be the solution of $\mathbf{R}(\mathbf{x}, \mathbf{u}) = \mathbf{0}$ and $\hat{\mathbf{x}} \in S_{++}[\mathbf{f}]$ be any strictly feasible solution of *NCP*[$\mathbf{f}$]. Since $\mathbf{f}$ is a monotone function,

$$(\mathbf{x} - \hat{\mathbf{x}})^T(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\hat{\mathbf{x}})) \ge 0.$$

It follows that

$$\mathbf{x}^T\mathbf{f}(\hat{\mathbf{x}}) + \hat{\mathbf{x}}^T\mathbf{f}(\mathbf{x}) \leq \mathbf{x}^T\mathbf{f}(\mathbf{x}) + \hat{\mathbf{x}}^T\mathbf{f}(\hat{\mathbf{x}}).$$

Since $\mathbf{x} \geq \mathbf{0}$, $\mathbf{f} \geq \mathbf{0}$, $\mathbf{x}^T\mathbf{f}(\mathbf{x})$ is bounded by Proposition 3.1, and $\hat{\mathbf{x}}$ and $\mathbf{f}(\hat{\mathbf{x}})$ are strictly positive by assumption, $\mathbf{x}$ must be bounded. Since $\mathbf{u}$ is chosen arbitrarily, assumption (B3) is satisfied. ∎

To ensure that assumption (B4) is satisfied, one must introduce the class of R_0-function.

DEFINITION 3.8. *Let S be a nonempty subset of R^n. The mapping $\mathbf{f} : R^n \to R^n$ is said to be an R_0-function over set S if for any sequence $\{\mathbf{x}^k\} \in S$ satisfying $\{\|\mathbf{x}^k\|\} \to \infty$ and*

$$\liminf_{k\to\infty} \frac{\min_i x_i^k}{\|\mathbf{x}^k\|} \geq 0, \quad \liminf_{k\to\infty} \frac{\min_i f_i(\mathbf{x}^k)}{\|\mathbf{x}^k\|} \geq 0,$$

there exists an index j such that $\{x_j^k\} \to \infty$ and $\{f_j(\mathbf{x}^k)\} \to \infty$.

Notice that the above definition of an R_0-function is slightly different from that used in [11]. R_0-functions may be viewed as a generalization of the concept of an R_0-matrix when $\mathbf{f}$ is affine.

DEFINITION 3.9. *A matrix $\mathbf{M} \in R^{n\times n}$ is said to be an R_0-matrix if the following system has no nonzero solution:*

$$\begin{aligned} &\mathbf{x} \geq \mathbf{0}, \\ &\mathbf{M}_{i\cdot}\mathbf{x} = 0 \textit{ if } x_i > 0, \\ &\mathbf{M}_{i\cdot}\mathbf{x} \geq 0 \textit{ if } x_i = 0. \end{aligned}$$

PROPOSITION 3.10. *Let $\mathbf{f}(\mathbf{x}) = \mathbf{M}\mathbf{x} + \mathbf{q}$ be an affine function. Then $\mathbf{f}$ is an R_0-function if and only if $\mathbf{M}$ is an R_0-matrix.*

Proof. Necessity: Let $\{\mathbf{x}^k\}$ be the sequence as given in Definition 3.8. Let $\{\mathbf{y}^k\}$ be any convergent subsequence of $\{\mathbf{x}^k\}$ such that $\lim_{k\to\infty} \mathbf{y}^k/\|\mathbf{y}^k\| = \mathbf{w}$. The assumptions on the sequence $\{\mathbf{x}^k\}$ imply that $\mathbf{w} \neq \mathbf{0}$, $\mathbf{w} \geq 0$, and $\mathbf{M}\mathbf{w} \geq 0$. If $\mathbf{M}$ is an R_0-matrix, then $\mathbf{w}^T\mathbf{M}\mathbf{w} > 0$. Thus, there exists an index j such that $w_j > 0$ and $(\mathbf{M}\mathbf{w})_j > 0$, which implies that $\{y_j^k\} \to \infty$ and $\{f_j(\mathbf{y}^k)\} \to \infty$. Since this is true for any convergent subsequence, $\mathbf{f}$ is an R_0-function by definition.

Sufficiency: Let $\mathbf{w}$ be a vector such that $\mathbf{w} \neq \mathbf{0}$, $\mathbf{w} \geq \mathbf{0}$, and $\mathbf{M}\mathbf{w} \geq \mathbf{0}$. Define a sequence $\{\mathbf{x}^k\} = \{t^k\mathbf{w}\}$, where $\{t^k\} \to \infty$ is a sequence of scalars. Clearly, the sequence $\{\mathbf{x}^k\}$ satisfies all the conditions in Definition 3.8. If $\mathbf{f}$ is an R_0-function, then, by definition, there exists a j such that $\{x_j^k\} \to \infty$ and $\{f_j(\mathbf{x}^k)\} \to \infty$. This implies $w_j > 0$ and $(\mathbf{M}\mathbf{w})_j > 0$. Therefore, $\mathbf{w}^T\mathbf{M}\mathbf{w} > 0$ and $\mathbf{M}$ is an R_0-matrix. ∎

An R_0-function can also be viewed as a generalization of a uniform P-function when $\mathbf{f}$ is nonlinear, as demonstrated by the following results.

PROPOSITION 3.11. *If $\mathbf{f}$ is a uniform P-function, then it is an R_0-function.*

Proof. The proof uses a similar proof technique by Kanzow for a related result (Theorem 3.9 of [6]). Let $\{\mathbf{x}^k\}$ be an unbounded sequence as given in Definition 3.8. Then the index set $I = \{i : |x_i^k| \to \infty\}$ is nonempty. Define another bounded sequence $\{\mathbf{y}^k\} \in R^n$ by

$$y_i^k = \begin{cases} 0 & \text{if } i \in I, \\ x_i^k & \text{if } i \notin I. \end{cases}$$

Since $\mathbf{f}$ is a uniform P-function, one has

$$\begin{aligned}\gamma \sum_{i\in I}(x_i^k)^2 &= \gamma\|\mathbf{x}^k - \mathbf{y}^k\|^2 \\ &\leq \max_{1\leq i\leq n}(x_i^k - y_i^k)(f_i(\mathbf{x}^k) - f_i(\mathbf{y}^k)) \\ &= \max_{i\in I} x_i^k(f_i(\mathbf{x}^k) - f_i(\mathbf{y}^k)).\end{aligned}$$

Since the sequence $\{\mathbf{y}^k\}$ is bounded, the sequence $\{f_i(\mathbf{y}^k)\}$ is also bounded by the assumption of continuity on f. Let j be the index that achieves the maximum in the right-hand side. If $x_j^k \to -\infty$, one of the following relations is true:

$$\lim_{k\to\infty}\inf \frac{x_j^k}{\|\mathbf{x}^k\|} < 0, \quad \lim_{k\to\infty}\inf \frac{f_j(\mathbf{x}^k)}{\|\mathbf{x}^k\|} < 0.$$

However, this contradicts the assumption of sequence $\{\mathbf{x}^k\}$. Therefore, the inequality implies $x_j^k \to \infty$ and $f_j(\mathbf{x}^k) \to \infty$. By definition, $\mathbf{f}$ is an R_0-function. □

Assumption (B4) is closely related to the growth behavior of $\|\mathbf{R}(\mathbf{x}, \mathbf{u})\|$, which in turn depends on growth of the natural residual $\mathbf{r}(\mathbf{x}) = \min\{\mathbf{f}(\mathbf{x}), \mathbf{x}\}$ of *NCP*[$\mathbf{f}$]. Indeed, many other merit functions are also related to $\mathbf{r}(\mathbf{x})$ (cf. [10]). Therefore, the following limit property of $\mathbf{r}(\mathbf{x})$ is established first.

PROPOSITION 3.12. *If* $\mathbf{f}$ *is an* R_0*-function over* R^n*, then for any unbounded sequence* $\{\mathbf{x}^k\}$*,*

$$\lim_{k\to\infty}\|\mathbf{r}(\mathbf{x}^k)\| = \infty.$$

Proof. Let $\{\mathbf{x}^k\}$ be any unbounded sequence. The result is clearly true if there exists a j such that $x_j^k \to -\infty$ or $f_j(\mathbf{x}^k) \to -\infty$. Therefore, one may assume that the sequence $\{\mathbf{x}^k\}$ satisfies all the conditions in Definition 3.8. Since $\mathbf{f}$ is an R_0-function, there exists a j such that $x_j^k \to \infty$ and $f_j(\mathbf{x}^k) \to \infty$. Thus, $\|\mathbf{r}(\mathbf{x})\|$ is unbounded. □

PROPOSITION 3.13. *If* $\mathbf{f}$ *is an* R_0*-function over* R^n*, then assumption* (B4) *is satisfied.*

Proof. For any given $\mathbf{u} \in R_+^n$, suppose there exists an unbounded sequence $\{\mathbf{x}^k\} \in \mathbf{R}_{\mathbf{u}}^{-1}(L)$ for some compact subset $L \in R^n$. Since $\mathbf{f}$ is an R_0-function, $\|\mathbf{r}(\mathbf{x}^k)\|$ is unbounded by Proposition 3.12. From assumption (A1),

$$\mathbf{r}(\mathbf{x}) - \mathbf{b}(\mathbf{u}) \leq \mathbf{R}(\mathbf{x}, \mathbf{u}) \leq \mathbf{r}(\mathbf{x}).$$

Thus, $\|\mathbf{R}(\mathbf{x}^k, \mathbf{u})\|$ is unbounded. However, this contradicts the assumptions that either $\{\mathbf{x}^k\} \in \mathbf{R}_{\mathbf{u}}^{-1}(L)$ or $\mathbf{R}(\mathbf{x}^k, \mathbf{u}) = \mathbf{a}^k$, where $\mathbf{a}^k$ belongs to a compact set L. □

Based on Proposition 3.13, assumptions (B1)–(B4) are satisfied if $\mathbf{f}$ is both a P_0- and an R_0-function and $S_{++}[\mathbf{f}] \neq \emptyset$. In particular, these conditions will hold if $\mathbf{f}$ is a uniform P-function.

3.2. A continuation method for NCP. Let $\mathbf{x}(\mathbf{u})$ be the solution of $\mathbf{R}(\mathbf{x}, \mathbf{u}) = \mathbf{0}$. From the discussion of the previous section, $\mathbf{x}(\mathbf{u})$ converges to a solution of *NCP*[$\mathbf{f}$] as $\mathbf{u}$ approaches zero. Based on this, a continuation method is constructed for *NCP*[$\mathbf{f}$] in this section. At each iteration, the continuation method first solves equation $\mathbf{R}(\mathbf{x}, \mathbf{u}) = \mathbf{0}$ for a fixed $\mathbf{u}$ by the damped Newton's method to certain accuracy (to be specified below) and then reduces parameter $\mathbf{u}$ systematically (to be specified below). A solution of *NCP*$[\mathbf{f}]$ is obtained when parameter $\mathbf{u}$ is reduced to zero. Let $H(\mathbf{x},\mathbf{u}) = \frac{1}{2}\mathbf{R}(\mathbf{x},\mathbf{u})^T\mathbf{R}(\mathbf{x},\mathbf{u})$ be the merit function of equation $\mathbf{R}(\mathbf{x},\mathbf{u}) = \mathbf{0}$. The continuation method is defined in detail below.

ALGORITHM 1.

Let $\sigma \in (0, 1/2)$, $\delta, \eta \in (0, 1)$, and β, ϵ be fixed positive constants. Choose $\mathbf{u}^0 \in R^n_{++}$ and $\mathbf{x}^0 \in R^n$ and set $k = 0$.

Step 1 *If $\|\mathbf{r}(\mathbf{x}^k)\| \le \epsilon$, stop; $\mathbf{x}^k$ is an approximate solution of NCP*$[\mathbf{f}]$.

Step 2 *Solve for the direction $\Delta\mathbf{x}^k$ as*

$$\Delta\mathbf{x}^k = -\nabla\mathbf{R}(\mathbf{x}^k, \mathbf{u}^k)^{-1}\mathbf{R}(\mathbf{x}^k, \mathbf{u}^k).$$

Step 3 *Compute the new point $\mathbf{x}^{k+1}$ by performing an Armijo line search:*

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \lambda_k\Delta\mathbf{x}^k$$

where $\lambda_k = \max\{1, \delta, \delta^2, \dots\}$ subject to

$$H(\mathbf{x}^k + \lambda_k\Delta\mathbf{x}^k, \mathbf{u}^k) \le (1 - \sigma\lambda_k)H(\mathbf{x}^k, \mathbf{u}^k)$$

Step 4 *If $\|\mathbf{R}(\mathbf{x}^{k+1}, \mathbf{u}^k)\| \le \beta\|\mathbf{u}^k\|$, then $\mathbf{u}^{k+1} = \eta\mathbf{u}^k$, else $\mathbf{u}^k = \mathbf{u}^{k+1}$. Set $k = k + 1$ and go to Step* 1.

The use of $\|\mathbf{r}(\mathbf{x})\|$ as a termination criterion is justified in section 3.3. The next result assures that the algorithm is well defined under assumption (B1).

PROPOSITION 3.14. *Under assumption* (B1)*,*

1. *$\nabla\mathbf{R}(\mathbf{x},\mathbf{u})$ is nonsingular for all $\mathbf{x} \in R^n$ and all $\mathbf{u} \in R^n_{++}$.*
2. *$\mathbf{x}$ is a stationary point of $H(\mathbf{x},\mathbf{u})$ if and only if it is a solution of $\mathbf{R}(\mathbf{x},\mathbf{u}) = \mathbf{0}$.*
3. *$H(\mathbf{x},\mathbf{u})$ has no more than one stationary point.*

Proof. By definition,

$$\nabla\mathbf{R}(\mathbf{x},\mathbf{u}) = \mathrm{diag}\{p'(x_i - f_i(\mathbf{x}), u_i)\}[\mathrm{diag}\{p'^{-1}(x_i - f_i(\mathbf{x}), u_i) - 1\} + \nabla\mathbf{f}(\mathbf{x})].$$

From part 2 of Proposition 2.1, $0 < p'(z, u_i) < 1$ for all z. By assumption (B1), $\mathbf{f}(\mathbf{x})$ is a P_0-function and, thus, $\nabla\mathbf{f}(\mathbf{x})$ is a P_0-matrix. It follows that $\nabla\mathbf{R}(\mathbf{x},\mathbf{u})$ is a P-matrix and, therefore, is nonsingular for all $\mathbf{x}$. The proof of the if statement in part 2 is trivial. Suppose $\mathbf{x}$ is a stationary point of $H(\mathbf{x},\mathbf{u})$; then

$$\nabla H(\mathbf{x},\mathbf{u}) = \nabla\mathbf{R}(\mathbf{x},\mathbf{u})^T\mathbf{R}(\mathbf{x},\mathbf{u}) = \mathbf{0}.$$

Since $\nabla\mathbf{R}(\mathbf{x},\mathbf{u})$ is nonsingular, one has $\mathbf{R}(\mathbf{x},\mathbf{u}) = \mathbf{0}$ and $\mathbf{x}$ solves the equation. Part 3 follows immediately from part 2 and the fact that the solution of equation $\mathbf{R}(\mathbf{x},\mathbf{u}) = \mathbf{0}$ is unique. □

The next result shows that the algorithm is globally convergent under assumptions (B1)–(B4).

THEOREM 3.15. *Let $\{\mathbf{x}^k\}$ be any infinite sequence generated by Algorithm* 1 *with $\epsilon = 0$. Then, under assumptions* (B1)–(B4)*:*

1. *The sequence $\{\mathbf{x}^k\}$ has at least one accumulation point.*
2. *Any accumulation point of $\{\mathbf{x}^k\}$ is a solution of NCP*$[\mathbf{f}]$.

Proof. By Proposition 3.14, the sequence $\{\mathbf{x}^k\}$ generated by Algorithm 1 is well defined. By assumption (B4), the level set $L_{\mathbf{u}}(\mathbf{x}^0) = \{\mathbf{x} \in R^n : \|\mathbf{R}(\mathbf{x},\mathbf{u})\| \le \|\mathbf{R}(\mathbf{x}^0,\mathbf{u})\|\}$ is bounded for any $\mathbf{u} \in R^n_{++}$ and initial point $\mathbf{x}^0 \in R^n$. Since Algorithm 1 is a descent method, the entire sequence $\{\mathbf{x}^k\}$ remains bounded and, therefore, has at least one accumulation point. For part 2, in view of Theorem 3.6, it suffices to show that $\mathbf{u}^k$ converges to $\mathbf{0}$. Suppose on the contrary that $\mathbf{u}^k = \mathbf{u}^{k+1}$ for all $k \geq K$. Then Algorithm 1 reduces to the damped Newton's method for a fixed parameter $\mathbf{u}^K$. However, for a fixed $\mathbf{u}^K$, it is well known that the condition in Step 4 will be satisfied in finite steps since $\nabla \mathbf{R}$ is nonsingular by part 1 of Proposition 3.14. Thus, $\mathbf{u}^k$ will be reduced and this leads to a contradiction. □

3.3. Error bounds of NCP with uniform P-function.

In the previous section, an algorithm to find an approximate solution of *NCP*[$\mathbf{f}$] was stated for a given value of $\mathbf{u}$. In this section, two error bounds on the distance between the approximate solution and the exact solution of the NCP with a uniform P-function are defined. These bounds can be used to ascertain the quality of the solution obtained from the approximation algorithm for a given value of $\mathbf{u}$.

First, consider the related error bounds (measured by $\mathbf{r}(\mathbf{x})$) that have appeared in the literature. Mathias and Pang [8] obtained both absolute and relative error bounds for the LCP with P-matrix. Pang [12] obtained both absolute and relative error bounds for the strongly monotone variational inequality with a linear constraint set. Ren [14] obtained absolute error bounds for the strongly monotone NCP for using the 1, 2, and ∞ norms. The error bounds stated below extend the error bounds in [8].

Assume throughout this section that $\mathbf{f}(\mathbf{x})$ is a uniform P-function and is Lipschitz continuous for all $\mathbf{x} \in R^n$.

LEMMA 3.16. *For all* $\mathbf{x}, \mathbf{y} \in R^n$,

$$\|\mathbf{x} - \mathbf{y}\| \leq \frac{L+1}{\gamma} \|\mathbf{r}(\mathbf{x}) - \mathbf{r}(\mathbf{y})\|.$$

Proof. Assume $\mathbf{x} \neq \mathbf{y}$, since the result is clearly true if $\mathbf{x} = \mathbf{y}$. Define two mappings $\mathbf{v}, \mathbf{w} : R^n \to R^n$ by

$$\mathbf{v}(\mathbf{z}) = \mathbf{z} - \mathbf{r}(\mathbf{z}),\ \ \mathbf{w}(\mathbf{z}) = \mathbf{f}(\mathbf{z}) - \mathbf{r}(\mathbf{z}).$$

Then

$$\mathbf{v}(\mathbf{z}) \geq \mathbf{0},\ \ \mathbf{w}(\mathbf{z}) \geq \mathbf{0},\ \text{ and } \mathbf{v}(\mathbf{z})^T\mathbf{w}(\mathbf{z}) = 0 \text{ for all } \mathbf{z} \in R^n.$$

Thus, for each $i = 1, \ldots, n$, one has

$$\begin{aligned} 0 &\geq (v_i(\mathbf{x}) - v_i(\mathbf{y}))(w_i(\mathbf{x}) - w_i(\mathbf{y})) \\ &= (x_i - y_i - r_i(\mathbf{x}) + r_i(\mathbf{y}))(f_i(\mathbf{x}) - f_i(\mathbf{y}) - r_i(\mathbf{x}) + r_i(\mathbf{y})) \\ &\geq (x_i - y_i)(f_i(\mathbf{x}) - f_i(\mathbf{y})) - (r_i(\mathbf{x}) - r_i(\mathbf{y}))(x_i - y_i + f_i(\mathbf{x}) - f_i(\mathbf{y})) \\ &\geq (x_i - y_i)(f_i(\mathbf{x}) - f_i(\mathbf{y})) - (L+1)\|\mathbf{r}(\mathbf{x}) - \mathbf{r}(\mathbf{y})\|\|\mathbf{x} - \mathbf{y}\|. \end{aligned}$$

Therefore,

$$(L+1)\|\mathbf{r}(\mathbf{x}) - \mathbf{r}(\mathbf{y})\|\|\mathbf{x} - \mathbf{y}\| \geq \max_i (x_i - y_i)(f_i(\mathbf{x}) - f_i(\mathbf{y})) \geq \gamma \|\mathbf{x} - \mathbf{y}\|^2.$$

The desired result follows by the assumption that $\mathbf{x} \neq \mathbf{y}$. □

LEMMA 3.17. *Let* $a, b, c, d \in R$. *Then*

$$|\min\{a, b\} - \min\{c, d\}| \leq \max\{|a - c|, |b - d|\}.$$

Proof. If $a \geq b \geq d \geq c$, then

$$|\min\{a,b\} - \min\{c,d\}| = |b-c| \leq |a-c| \leq \max\{|a-c|, |b-d|\}.$$

All other cases can be shown in a similar manner. □

THEOREM 3.18. *For any* $\mathbf{x}, \mathbf{y} \in R^n$,

$$\frac{1}{\max\{L,1\}} \|\mathbf{r}(\mathbf{x}) - \mathbf{r}(\mathbf{y})\| \leq \|\mathbf{x}-\mathbf{y}\| \leq \frac{L+1}{\gamma} \|\mathbf{r}(\mathbf{x}) - \mathbf{r}(\mathbf{y})\|.$$

Proof. In view of Lemma 3.16, it suffices to show that

$$\|\mathbf{r}(\mathbf{x}) - \mathbf{r}(\mathbf{y})\| \leq \max\{L,1\}\|\mathbf{x}-\mathbf{y}\|.$$

Indeed,

$$\begin{aligned}\|\mathbf{r}(\mathbf{x}) - \mathbf{r}(\mathbf{y})\| &\leq \max\{\|\mathbf{x}-\mathbf{y}\|, \|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\|\} \\ &\leq \max\{\|\mathbf{x}-\mathbf{y}\|, L\|\mathbf{x}-\mathbf{y}\|\} \\ &= \max\{L,1\}\|\mathbf{x}-\mathbf{y}\|,\end{aligned}$$

where the first inequality follows from the previous lemma. □

Using the above results, one can now obtain an absolute error bound for the NCP with uniform P-function as a corollary of Theorem 3.18.

COROLLARY 3.19. *Let* $\mathbf{z}$ *be the unique solution of NCP*$[\mathbf{f}]$. *Then*

$$\frac{1}{\max\{L,1\}} \|\mathbf{r}(\mathbf{x})\| \leq \|\mathbf{x}-\mathbf{z}\| \leq \frac{L+1}{\gamma} \|\mathbf{r}(\mathbf{x})\|.$$

The next lemma will be used to obtain a relative error bound.

LEMMA 3.20. *Let* $\mathbf{z}$ *be the unique solution of NCP*$[\mathbf{f}]$. *Then*

$$\frac{\gamma}{L}\|(-\mathbf{f}(\mathbf{0}))_+\| \leq \|\mathbf{z}\| \leq \frac{1}{\gamma}\|(-\mathbf{f}(\mathbf{0}))_+\|.$$

Proof. If $\mathbf{f}(\mathbf{0}) \geq \mathbf{0}$, then $\mathbf{z} = \mathbf{0}$ and the result clearly holds. Assume that $\mathbf{f}(\mathbf{0}) \ngeq \mathbf{0}$. Then $\mathbf{z} \neq \mathbf{0}$. In this case, one has

$$\begin{aligned}\gamma\|\mathbf{0}-\mathbf{z}\|^2 &\leq \max_i (0 - z_i)(f_i(\mathbf{0}) - f_i(\mathbf{z})) \\ &= \max_i z_i(-f_i(\mathbf{0})) \\ &\leq \|\mathbf{z}\|\|(-\mathbf{f}(\mathbf{0}))_+\|.\end{aligned}$$

This inequality gives the upper bound of $\|\mathbf{z}\|$. To establish the lower bound of $\|\mathbf{z}\|$, note that since $\mathbf{f}$ is a uniform P-function and is Lipschitz continuous, it must be the case that

$$\max_i (x_i - y_i)(f_i(\mathbf{x}) - f_i(\mathbf{y})) \geq \gamma\|\mathbf{x}-\mathbf{y}\|^2 \geq \frac{\gamma}{L}\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\|^2 \text{ for all } \mathbf{x}, \mathbf{y} \in R^n.$$

Since $\mathbf{z}$ is the solution of *NCP*$[\mathbf{f}]$, one has $\mathbf{f}(\mathbf{z}) - \mathbf{f}(\mathbf{0}) \geq -\mathbf{f}(\mathbf{0})$, which implies $\|(-\mathbf{f}(\mathbf{0}))_+\| \leq \|\mathbf{f}(\mathbf{z}) - \mathbf{f}(\mathbf{0})\|$. In addition, $z_i f_i(\mathbf{z}) = 0$ for all $i = 1, \dots, n$. It follows that

$$\begin{aligned}\frac{\gamma}{L}\|(-\mathbf{f}(\mathbf{0}))_+\|^2 &\leq \frac{\gamma}{L}\|\mathbf{f}(\mathbf{z}) - \mathbf{f}(\mathbf{0})\|^2 \\ &\leq \max_i (z_i - 0)(f_i(\mathbf{z}) - f_i(\mathbf{0})) \\ &= \max_i z_i(-f_i(\mathbf{0})) \\ &\leq \|\mathbf{z}\|\|(-\mathbf{f}(\mathbf{0}))_+\|.\end{aligned}$$

As a result, the lower bound on $\|\mathbf{z}\|$ is obtained. □

Combining Lemma 3.20 and Corollary 3.19, one can now obtain a relative error bound for any point $\mathbf{x}$ in R^n.

THEOREM 3.21. *Let* $\mathbf{z}$ *be the unique solution of NCP*$[\mathbf{f}]$ *and* $\|\mathbf{z}\| \neq 0$. *Then*

$$\frac{\gamma}{\max\{L, 1\}} \frac{\|\mathbf{r}(\mathbf{x})\|}{\|(-\mathbf{f}(\mathbf{0}))_+\|} \leq \frac{\|\mathbf{x} - \mathbf{z}\|}{\|\mathbf{z}\|} \leq \frac{L(L+1)}{\gamma^2} \frac{\|\mathbf{r}(\mathbf{x})\|}{\|(-\mathbf{f}(\mathbf{0}))_+\|}.$$

Under assumption (A1), one can easily express the above error bounds in terms of $\mathbf{R}(\mathbf{x}, \mathbf{u})$ by observing that

$$\mathbf{R}(\mathbf{x}, \mathbf{u}) \leq \mathbf{r}(\mathbf{x}) \leq \mathbf{R}(\mathbf{x}, \mathbf{u}) + \mathbf{b}(\mathbf{u}),$$

and, therefore,

$$\min\{\|\mathbf{R}(\mathbf{x}, \mathbf{u})\|, \|\mathbf{R}(\mathbf{x}, \mathbf{u}) + \mathbf{b}(\mathbf{u})\|\} \leq \|\mathbf{r}(\mathbf{x})\| \leq \max\{\|\mathbf{R}(\mathbf{x}, \mathbf{u})\|, \|\mathbf{R}(\mathbf{x}, \mathbf{u}) + \mathbf{b}(\mathbf{u})\|\}.$$

COROLLARY 3.22. *Let* $\mathbf{z}$ *be the unique solution of NCP*$[\mathbf{f}]$ *and* $\|\mathbf{z}\| \neq 0$. *Then under assumption* (A1),

$$\begin{aligned}\frac{1}{\max\{L, 1\}}\min\{\|\mathbf{R}(\mathbf{x}, \mathbf{u})\|, \|\mathbf{R}(\mathbf{x}, \mathbf{u}) + \mathbf{b}(\mathbf{u})\|\} &\leq \|\mathbf{x} - \mathbf{z}\| \\ &\leq \frac{L+1}{\gamma}\max\{\|\mathbf{R}(\mathbf{x}, \mathbf{u})\|, \|\mathbf{R}(\mathbf{x}, \mathbf{u}) + \mathbf{b}(\mathbf{u})\|\}, \\ \frac{\gamma}{\max\{L, 1\}} \frac{\min\{\|\mathbf{R}(\mathbf{x}, \mathbf{u})\|, \|\mathbf{R}(\mathbf{x}, \mathbf{u}) + \mathbf{b}(\mathbf{u})\|\}}{\|(-\mathbf{f}(\mathbf{0}))_+\|} &\leq \frac{\|\mathbf{x} - \mathbf{z}\|}{\|\mathbf{z}\|} \\ &\leq \frac{L(L+1)}{\gamma^2} \frac{\max\{\|\mathbf{R}(\mathbf{x}, \mathbf{u})\|, \|\mathbf{R}(\mathbf{x}, \mathbf{u}) + \mathbf{b}(\mathbf{u})\|\}}{\|(-\mathbf{f}(\mathbf{0}))_+\|}.\end{aligned}$$

In particular, for the solution $\mathbf{x}(\mathbf{u})$ *of* $\mathbf{R}(\mathbf{x}, \mathbf{u}) = \mathbf{0}$, *one has*

$$\|\mathbf{x}(\mathbf{u}) - \mathbf{z}\| \leq \frac{L+1}{\gamma}\|\mathbf{b}(\mathbf{u})\| \quad \textit{and} \quad \frac{\|\mathbf{x}(\mathbf{u}) - \mathbf{z}\|}{\|\mathbf{z}\|} \leq \frac{L(L+1)}{\gamma^2} \frac{\|\mathbf{b}(\mathbf{u})\|}{\|(-\mathbf{f}(\mathbf{0}))_+\|}.$$

This corollary is an extension to Theorem 3.2 of [3], where $\mathbf{f}$ was assumed to be strongly monotone and no relative error bound was given.

4. Future research.

Given the computational success of these smoothing methods as reported in [3], the next phase of this research involves the development and testing of alternative smoothing functions, including those that are not derivable from the integration of probability densities. One such function is shown in the following example.

Example 3. Auto-scaling interior point smooth function.

$$p(z,u) = \frac{z + \sqrt{z^2 + 4u^2}}{2} + u,$$

where $b(u) = 2u$. With this approximation, it can be shown that x and y solve the equation $x = p(x - y, u)$ if and only if they solve the following system:

$$x > 0,\ y > 0,\ xy = u(x + y).$$

The approximation is similar to the interior point smooth function except that it scales u in the right-hand side of the above system by $(x + y)$. However, it does not satisfy assumption (A3). Extensions to the theory to handle such functions, if they prove computationally efficient, will also be addressed in future research.

Acknowledgments. The authors are thankful to the referees and Jong-Shi Pang for their insightful comments. In particular, Proposition 2.7 is motivated by an interesting question raised by a referee.

REFERENCES

[1] B. Chen and P. T. Harker, *A noninterior point continuation method for linear complementarity problems*, SIAM J. Matrix Anal. Appl., 20 (1993), pp. 1168–1190.

[2] C. Chen and O. L Mangasarian, *Smoothing Methods for Convex Inequalities and Linear Complementarity Problems*, Math. Programming, 71 (1995), pp. 51–69.

[3] C. Chen and O. L. Mangasarian, *A Class of Smoothing Function for Nonlinear and Mixed Complementarity Problems*, Comput. Optim. Appl., 5 (1996), pp. 97–138.

[4] P. T. Harker and J. S. Pang, *Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications*, Math. Programming, 48 (1990), pp. 161–220.

[5] C. Kanzow, *Some noninterior continuation methods for linear complementarity problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 851–868.

[6] C. Kanzow, *A New Approach to Continuation Methods for Complementarity Problems with Uniform P-Functions*, Technical report, Institute of Applied Mathematics, University of Hamburg, Germany, 1994.

[7] M. Kojima, S. Mizuno, and T. Noma, *Homotopy continuation method for nonlinear complementarity problems*, Math. Oper. Res., 16 (1991), pp. 754–774.

[8] R. Mathias and J. S. Pang, *Error bounds for the linear complementarity problem with P-matrix*, Linear Algebra Appl., 132 (1990), pp. 123–136.

[9] J. M. Ortega and W. G. Rheinbolt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[10] P. Tseng, *Growth behavior of a class of merit functions for the nonlinear complementarity problem*, J. Optim. Theory Appl., 89 (1996), pp. 446–460.

[11] P. Tseng, *An infeasible path-following method for monotone complementarity problem*, SIAM J. Optim., 7 (1997), pp. 386–402.

[12] J. S. Pang, *A posterior error bounds for the linearly-constrained variational inequality problem*, Math. Oper. Res., 12 (1987), pp. 474–484.

[13] J. S. Pang, *Complementarity Problems*, in Handbook on Global Optimization, R. Horst and P. Pardalos, eds., Kluwer Academic Publishers, Norwell, MA, 1994.

[14] J. Ren, *Computable Error Bounds in Mathematical Programming*, Technical report 1173, Computer Science Department, University of Wisconsin, Madison, WI, 1993.

[15] J. T. Schwartz, *Nonlinear Functional Analysis*, Gordon and Breach Science Publishers, New York, 1969.

SIAM J. Optim.
Vol. 7, No. 2, pp. 421–444, May 1997

CONVERGENCE RATES IN FORWARD–BACKWARD SPLITTING[*]

GEORGE H-G. CHEN[†] AND R. T. ROCKAFELLAR[‡]

Abstract. Forward–backward splitting methods provide a range of approaches to solving large-scale optimization problems and variational inequalities in which structures conducive to decomposition can be utilized. Apart from special cases where the forward step is absent and a version of the proximal point algorithm comes out, efforts at evaluating the convergence potential of such methods have so far relied on Lipschitz properties and strong monotonicity, or inverse strong monotonicity, of the mapping involved in the forward step, the perspective mainly being that of projection algorithms. Here, convergence is analyzed by a technique that allows properties of the mapping in the backward step to be brought in as well. For the first time in such a general setting, global and local contraction rates are derived; moreover, they are derived in a form which makes it possible to determine the optimal step size relative to certain constants associated with the given problem. Insights are thereby gained into the effects of shifting strong monotonicity between the forward and backward mappings when a splitting is selected.

Key words. forward–backward splitting, numerical optimization, variational inequalities, projection algorithms, matrix splitting, operator splitting, convex programming

AMS subject classifications. 49R40, 49M27, 90C25, 90C06

PII. S1052623495290179

1. Introduction.

This paper concerns a class of numerical methods for finding solutions to variational inequalities and other "generalized equations," especially in circumstances where a need for decomposition into simpler subproblems is apparent. Optimization problems fit the framework of these methods through the ways that variational inequalities can express first-order optimality conditions in primal, dual, or primal–dual form. Variational inequalities serve also in models of equilibrium and a diversity of other applications.

In general, the *variational inequality* problem for a closed convex set $C \subset \mathbb{R}^n$ and a continuous mapping $F : C \to \mathbb{R}^n$ looks for a vector $\bar{x}$ such that

$$0 \in T(\bar{x}) \quad \text{for} \quad T(x) = F(x) + N_C(x), \tag{1.1}$$

where $N_C(x)$ is the set-valued normal cone mapping associated with C:

$$N_C(x) = \begin{cases} \{ w \in \mathbb{R}^n \mid \langle w, x' - x \rangle \le 0 \text{ for all } x' \in C \} & \text{when } x \in C, \\ \emptyset & \text{when } x \notin C, \end{cases} \tag{1.2}$$

with $\langle \cdot, \cdot \rangle$ denoting the canonical scalar product of vectors. The variational inequality problem is a *complementarity* problem when $C = \mathbb{R}^n_+$. Especially important is the case where F is *monotone* on C, in the sense that

$$\langle F(x') - F(x), x' - x \rangle \ge 0 \quad \text{for all} \quad x, x' \in C, \tag{1.3}$$

which in the optimization setting characterizes problems of convex type. Then the set-valued mapping T is itself monotone:

$$\langle w' - w, x' - x \rangle \ge 0 \quad \text{whenever} \quad w \in T(x),\ w' \in T(x'). \tag{1.4}$$

[*] Received by the editors August 10, 1995; accepted for publication (in revised form) December 1, 1995. This work was supported in part by NSF grants DMS–9200303 and DMS–9500957.
http://www.siam.org/journals/siopt/7-2/29017.html

[†] Aero-Geo-Astro, Ltd., 15631 King Place, Lynnwood, WA 98037 (a-georch@exchange.microsoft.com).

[‡] Dept. of Mathematics, Box 354350, University of Washington, Seattle, WA 98195-4350 (rtr@math.washington.edu).

In fact, it is *maximal* monotone—its graph set $\{(x,w) \,|\, w \in T(x)\}$ can't be enlarged without destroying monotonicity.

Forward–backward splitting methods are versatile in offering ways of exploiting the special structure of variational inequality problems. Following Lions and Mercier [1], such methods can be posed broadly in terms of solving

$$0 \in T(\bar{x}) \quad \text{when} \quad T(x) = T_1(x) + T_2(x) \tag{1.5}$$

for any mapping T that associates with each $x \in \mathbb{R}^n$ a (possibly empty) set $T(x) \subset \mathbb{R}^n$, a situation we symbolize by $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$, and any representation of T as a sum of two other such mappings T_1 and T_2. The representation $T = T_1 + T_2$, which might be set up in a multitude of different ways, is called a *splitting* of T. From an initial point x_0, a point x_k is generated in each iteration k for $k = 1, 2, \ldots$ by solving the subproblem

$$0 \in (T_{1k} + T_2)(x_k) \quad \text{with} \quad T_{1k}(x) = T_1(x_{k-1}) + \tfrac{1}{\lambda_k} H_k[x - x_{k-1}] \tag{1.6}$$

for a *step size* value $\lambda_k > 0$ and an *implementation matrix* $H_k \in \mathbb{R}^{n \times n}$. Under the license of denoting the linear mapping $x \mapsto H_k x$ by the same symbol H_k, the iterations can be written in the form

$$x_k \in S_k(x_{k-1}) \quad \text{for} \quad S_k = \big(H_k + \lambda_k T_2\big)^{-1}\big(H_k - \lambda_k T_1\big). \tag{1.7}$$

The *forward–backward* name comes from the fact that (as long as H_k is nonsingular) the iteration mapping S_k has the equivalent expression

$$S_k = \big(I + \lambda_k H_k^{-1} T_2\big)^{-1}\big(I - \lambda_k H_k^{-1} T_1\big).$$

In the language of numerical analysis, $I - \lambda_k H_k^{-1} T_1$ gives a *forward* step with step size λ_k and direction vector $d_k = -H_k^{-1} u_k$, $u_k \in T_1(x_k)$ (or $u_k = T_1(x_k)$ when T_1 is single valued), whereas $(I + \lambda_k H_k^{-1} T_2)^{-1}$ gives a *backward* step. Implementations where H_k is symmetric and positive definite are central, but weaker requirements are of interest in some situations.

For the purpose of solving a variational inequality (1.1), forward–backward splitting methods can be applied to

$$T = F + N_C, \qquad T_1 = F_1, \qquad T_2 = F_2 + N_C, \qquad \text{with } F = F_1 + F_2 \tag{1.8}$$

for a choice of continuous mappings $F_1 : C \to \mathbb{R}^n$ and $F_2 : C \to \mathbb{R}^n$. The iterations mean then that x_k is determined by solving

$$\begin{aligned} 0 \in T_k(x_k) \quad \text{for} \quad & T_k(x) = (F_{1k} + F_2)(x) + N_C(x) \quad \text{with} \\ & F_{1k}(x) = F_1(x_{k-1}) + \tfrac{1}{\lambda_k} H_k[x - x_{k-1}]. \end{aligned} \tag{1.9}$$

This covers many numerical procedures, the most familiar among them being ones that correspond to the splitting choices where either $F_1 = F$, $F_2 = 0$, or, at the other extreme, $F_1 = 0$, $F_2 = F$.

For the splitting where $F_1 = F$ and $F_2 = 0$ in (1.8), so that $T_1 = F$ and $T_2 = N_C$, the forward–backward iterations with symmetric, positive definite H_k give a *projection algorithm* (of possibly "variable metric" type): x_k is the point of C nearest to

$$x'_k = x_{k-1} - \lambda_k H_k^{-1} F(x_{k-1}), \tag{1.10}$$

with respect to the norm induced by H_k. Indeed, (1.9) can be written in terms of (1.10) as the relation $-H_k[x_k - x'_k] \in N_C(x_k)$, which is necessary and sufficient for having

$$x_k = \operatorname*{argmin}_{x \in C} \Big\langle [x - x'_k], H_k[x - x'_k] \Big\rangle. \tag{1.11}$$

Of course, if $C = \mathbb{R}^n$ then the projection trivializes and there's no backward step, just a forward step: one has $x_k = x_{k-1} - \lambda_k H_k^{-1} F(x_{k-1})$.

Among projection algorithms (1.9)–(1.10), the *gradient* case $F = \nabla f$ is the best known. If $H_k = I$, a variant of Cauchy's method is obtained, whereas if H_k is taken to be an approximation to $\nabla F(\bar{x}) = \nabla^2 f(\bar{x})$, a form of Newton's method comes out. Gradient projection algorithms were first studied in the Cauchy form by Goldstein [2] and in the Newton form by Levitin and Polyak [3], and they have since generated a large literature in optimization. For general variational inequalities, projection algorithms go back to Brézis and Sibony [4]; see also Sibony [5], Gajewski and Kluge [6], and for early developments attuned to mathematical programming, especially Dafermos [7].

For the other extreme splitting in (1.8), where $F_1 = F$ and $F_2 = 0$ so that $T_1 = 0$ and $T_2 = F + N_C$, the forward–backward procedure specializes to backward steps only and thus turns into (a "variable metric" form of) the *proximal point algorithm* for the mapping $T = F + N_C$. The proximal point algorithm was developed as a numerical method by Rockafellar [8], [9] in the case of $H_k \equiv I$ or, equivalently, $H_k \equiv H$ symmetric and positive definite, since that differs only in the designation of the norm (the context being one of a Hilbert space anyway). This algorithm is known to include, through various special choices, many other schemes such as generalized Douglas–Rachford splitting, cf. Eckstein and Bertsekas [10], and Spingarn splitting [11], which apply to maximal monotone mappings T not just of the variational inequality type in (1.1). An illuminating overview of splitting methods of all kinds has been provided by Eckstein [12].

Forward–backward splitting is closely related to an algorithmic approach introduced by Cohen as the "auxiliary problem principle" for problems of optimization in [13], [14], and variational inequalities in [15]. Cohen's formulation allows for the replacement of the linear implementation mapping $x \mapsto H_k x$ by a kind of nonlinear mapping, an idea treated also by Pang and Chan [16], among others. Patriksson [17] has explored this possibility broadly, showing how a vast array of known procedures can thereby be put into the framework of forward–backward methods.

Our focus in this paper is on the general iterations (1.7) for splittings $T = T_1 + T_2$ with T_1 single valued in which T, T_1, and T_2 are monotone and both $T_1 \not\equiv 0$ and $T_2 \not\equiv 0$, so that nontrivial forward steps as well as nontrivial backward steps can be expected. In the variational inequality context this corresponds to splittings of type (1.8) in which F, F_1, and F_2 are monotone and $F_1 \not\equiv 0$. We aim in particular at an understanding of convergence in cases where $F_2 \not\equiv 0$ too, so that more than a projection algorithm is involved. Such forms of forward–backward splitting methods are suggested by the decomposition needs of large-scale optimization problems with dynamic or stochastic structure [18], [19], [20] or PDE structure [21], but they haven't previously received much attention.

Except in connection with a weak ergodic type of convergence, cf. Passty [22], most of the research on general forward–backward splitting methods has relied on assumptions of strong monotonicity. Recall that a mapping $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is *strongly* *monotone* if there is a constant $\mu > 0$ such that

$$(1.12) \qquad \langle w' - w,\, x' - x\rangle \geq \mu\|x' - x\|^2 \quad \text{whenever} \quad w \in T(x),\ w' \in T(x')$$

or, equivalently, the mapping $T - \mu I$ is monotone. By the same token the inverse mapping T^{-1}, defined by taking $x \in T^{-1}(w)$ to mean that $w \in T(x)$, is strongly monotone if there is a constant $\nu > 0$ such that

$$(1.13) \qquad \langle w' - w,\, x' - x\rangle \geq \nu\|w' - w\|^2 \quad \text{whenever} \quad w \in T(x),\ w' \in T(x').$$

The strong monotonicity of T^{-1} is sometimes called the *Dunn* property or the *cocoercivity* of T. If T is single valued and Lipschitz continuous with constant κ and strongly monotone with constant μ, then T^{-1} is strongly monotone with constant $\nu = \mu/\kappa^2$.

For implementations with $H_k \equiv I$ and $\lambda_k \equiv \lambda$, Gabay [23] showed that if T_1 is single valued and maximal monotone with constant μ_1 as well as Lipschitz continuous with constant κ_1, the sequence of iterates x_k generated from any starting point x_0 converges to the unique solution $\bar{x}$ to (1.1), as long as $0 < \lambda < 2\mu_1/\kappa_1^2$. Alternatively he obtained convergence by assuming that a solution exists and T_1^{-1} is strongly monotone with constant ν_1 (which entails T_1 being Lipschitz continuous with constant $1/\nu_1$) and by taking $0 < \lambda < 2\nu_1$. Tseng [24] extended the latter result to nonconstant step sizes λ_k and used it in that paper and in [25] to verify convergence for some schemes of problem decomposition. Further work in this vein, allowing for nonlinear implementation mappings and even for the approximation of T_1 and T_2 by mappings T_1^k and T_2^k in iteration k, was carried out to a certain degree by Mouallif, Nguyen, and Strodiot [26] and Makler–Scheimberg, Nguyen, and Strodiot [27].

In the special case of projection algorithms, Dafermos in [7] obtained Q-linear convergence as a consequence of deriving a global contraction rate for the iterations (1.10)–(1.11). She did this for a fixed matrix $H_k \equiv H$, possibly different from I, employing the H-norm

$$(1.14) \qquad \|x\|_H = \sqrt{\langle x, Hx\rangle}$$

and its dual instead of the canonical norm $\|x\|$. She determined the fixed step size $\lambda_k \equiv \lambda$ for which the contraction rate would be optimal relative to constants of Lipschitz continuity and strong monotonicity for F when estimated in a certain way. These results were sharpened for affine variational inequalities by Dupuis and Darveau [28]. Bertsekas and Gafni [29] demonstrated R-linear convergence, i.e.,

$$(1.15) \qquad \limsup_{k\to\infty} \|x_k - \bar{x}\|^{1/k} < 1,$$

for the case where C is polyhedral but F is not itself strongly monotone, rather just of the form $A^\top F_0 A$ for a strongly monotone mapping F_0 and a matrix A. Zanni [30] showed that the rate estimates of Dafermos and of Bertsekas and Gafni could not be expected to support rapid convergence; as an alternative he developed for the affine case in [31] a change of variables which offers a substantial improvement. Renaud in his thesis [32] got a contraction rate based on strong monotonicity constants for both F and F^{-1}. Marcotte and Wu [33], in proceeding from Tseng [25] and Luo and Tseng [34], proved linear convergence when C is polyhedral and F is affine with F^{-1} strongly monotone. Tseng in [35] developed broad conditions for Q-linear convergence of iterative methods which he applied to projection methods for affine variational inequalities without, however, dealing explicitly with rate estimates or step sizes. For a survey of solution methods for finite-dimensional variational inequalities more generally, see Harker and Pang [36].

Little was known until recently about linear rates of convergence in the general setting of forward–backward methods. Renaud [32] succeeded in demonstrating R-linear convergence (1.15), although not actual contraction, in circumstances where T_1^{-1} is strongly monotone while T exhibits strong monotonicity relative to a unique solution $\bar{x}$. In Chen's thesis [37], contraction rates were developed under a variety of hypotheses entailing strong monotonicity of T, and step size optimization relative to those rate estimates was carried out.

Our efforts here take off from [37] in directions pioneered by Dafermos [7], going further than she and through territory encompassing much more than just projection algorithms. We reach conclusions significantly stronger than those of Chen [37] in some respects.

For simplicity at the start, we concentrate in section 2 on a constant step size $\lambda_k \equiv \lambda$ and a constant matrix $H_k \equiv H$, which we allow to differ from I but assume to be symmetric and positive definite. We work at establishing linear convergence in the strong sense of global contractivity of the mapping

$$S_\lambda = \big(H + \lambda T_2\big)^{-1}\big(H - \lambda T_1\big) = \big(I + \lambda H^{-1} T_2\big)^{-1}\big(I - \lambda H^{-1} T_1\big) \tag{1.16}$$

with respect to the norm $\|\cdot\|_H$. Thus, we seek $\theta_\lambda \in [0,1)$ such that $\|S_\lambda(x') - S_\lambda(x)\|_H \le \theta_\lambda \|x' - x\|_H$ for all x and x', hence, in particular,

$$\|S_\lambda(x) - \bar{x}\|_H \le \theta_\lambda \|x - \bar{x}\|_H \quad \text{for all} \quad x. \tag{1.17}$$

We try to do this in such a manner that θ_λ can be expressed in terms of estimated properties of the given problem, thereby opening the way to optimizing θ_λ with respect to the choice of λ and obtaining some guidance on how λ might be selected in practice.

Obviously $\alpha_{\min}\|x\| \le \|x\|_H \le \alpha_{\max}\|x\|$ for the lowest and highest eigenvalues $\alpha_{\min}$ and $\alpha_{\max}$ of H, so that linear convergence with respect to $\|\cdot\|_H$ is equivalent to linear convergence with respect to $\|\cdot\|$. But the *rate* of linear convergence, as quantified by the size of the contraction factor, which is the crucial measure for numerical purposes, could be quite different in the two cases. By working with $\|\cdot\|_H$ we are able to capture a better rate through finer tuning. This corresponds essentially to a change of variables in which we look at behavior in $u = H^{-1/2}x$ instead of x, but our pattern is to proceed with the analysis directly in terms of x. More consistently than Dafermos and others in this subject, we avoid reference to the canonical norm $\|\cdot\|$ so as to keep our results close to the natural geometry of the method and away from extraneous dependence on the condition number of H through appeal to the eigenvalues $\alpha_{\min}$ and $\alpha_{\max}$. The philosophy is that if the condition number is to have any role at all, it should only be relative to a *one-time* change of variables, not a change to another norm and back again in every iteration, which is the unfortunate effect of bringing $\alpha_{\min}$ and $\alpha_{\max}$ into estimates of a contraction rate.

We utilize Lipschitz properties of T_1, but, in contrast to all previous research, we base the constant on a *residual* part of T_1, obtained by subtracting off the strong monotonicity that has been identified. We refer the Lipschitz constant to $\|\cdot\|_H$ and the corresponding dual norm $\|\cdot\|_{H^{-1}}$. Likewise, we adapt our estimates of strong monotonicity to $\|\cdot\|_H$ instead of $\|\cdot\|$.

Especially to be noted is that we don't insist on strong monotonicity of either T_1 or T_1^{-1}. This is motivated by prospective applications to the large-scale problems cited in [18], [19], and [20]. Roughly, such problems follow the lines of minimizing $f(x) + g\big(D(x)\big)$ for proper lower semicontinuous (lsc) convex functions f and g and a mapping D like a discrete differential operator, integration operator, or expectation operator. The subgradient condition for $\bar{x}$ to be optimal involves a dual element $\bar{y}$ such that $-D^\top \bar{y} \in \partial f(\bar{x})$ and $D\bar{x} \in \partial g^*(\bar{y})$, where g^* is the convex function conjugate to g. This condition can be written as

$$(1.18) \qquad (0,0) \in (T_1 + T_2)(\bar{x}, \bar{y}) \quad \text{for} \quad \begin{cases} T_1(x,y) &= (D^\top y, -Dx), \\ T_2(x,y) &= \big(\partial f(x), \partial g^*(y)\big), \end{cases}$$

and it thus corresponds to a problem in $z = (x,y)$ that consists of solving $0 \in T(\bar{z})$ in the presence of a splitting $T = T_1 + T_2$ with T_1 and T_2 maximal monotone. Separability properties of f and g, reflected in a parallel choice of H, typically make it easy to iterate with $(x_k, y_k) = S_\lambda(x_{k-1}, y_{k-1})$, but T_1 is an *antisymmetric* linear mapping, so that neither T_1 nor T_1^{-1} can be strongly monotone. No results prior to ours could say anything substantial about convergence in this instance of a forward–backward splitting method. Note that (1.18) also gives incentive for not stopping at variational inequality models (1.8) in the treatment of such methods.

In section 3 we study the implications of our basic results for the ways that a splitting $T = T_1 + T_2$ might be set up most advantageously. Applications are made to procedures for solving variational inequalities—in particular, projection algorithms. We show a better contraction rate than that of Dafermos [7] or the one of Dupuis and Darveau [28] for affine variational inequalities; the result resembles a recent one of Zanni [31] but goes further. The step size associated with our contraction rate has the remarkable property of automatically optimizing performance with respect to the possible shifts of strong monotonicity between T_1 and T_2. The surprising result that as long as our step size prescription is followed any forward–backward method in the variational inequality case (1.8)–(1.9) can equally well be executed as a projection algorithm is thus achieved.

The global analysis of section 3 is supplemented in section 4 by a local analysis of convergence. Variable step sizes λ_k and implementation matrices H_k are taken up in section 5 and methods with asymmetric implementation matrices in section 6. For the literature on asymmetric implementations in solving variational inequalities, see Pang and Chan [16], Dafermos [38], Tseng [25], and Patriksson [17].

Because we are concerned with broad theoretical issues, we omit from the present study a number of refinements that could be pursued. The question of what happens when the subproblems in (1.6) or (1.9) are solved only approximately is not dealt with here, nor is the question of improvements based on augmenting the procedure with line search relative to some merit function. On the other hand, because we put our energy into the task of solving $0 \in T(\bar{x})$ for mappings T not necessarily of the variational inequality form (1.1), we get results that apply equally well to problems where, for example as in (1.18), the normal cone mapping N_C in (1.1) may be replaced by the subgradient mapping associated with a possibly nonsmooth convex function.

2. Global convergence analysis.

A mapping T that assigns to each $x \in \mathbb{R}^n$ a set $T(x) \subset \mathbb{R}^n$ (perhaps a singleton) is indicated by $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$. The effective domain of such a mapping is the set $\operatorname{dom} T = \big\{x \,\big|\, T(x) \neq \emptyset\big\}$. When T is maximal monotone, $\operatorname{dom} T$ is almost convex, in the sense that $\operatorname{cl}(\operatorname{dom} T)$ is a convex set whose relative interior lies within $\operatorname{dom} T$; cf. Minty [39]. The graph of T is considered to be the set of pairs (x, w) such that $w \in T(x)$, and the graph of T^{-1} consists therefore of the reversals (w, x) of all such pairs. The set of solutions $\bar{x}$ to $0 \in T(\bar{x})$ is $T^{-1}(0)$.

We investigate the feasibility of determining a solution $\bar{x}$ through iterations $x_k \in S_\lambda(x_{k-1})$ of the mapping in (1.16), as dictated by a choice of a splitting $T = T_1 + T_2$, a step size $\lambda > 0$, and an implementation matrix H. We don't suppose necessarily that T takes the variational inequality form in (1.1), but we do, for now, make the following assumptions.

Basic assumptions (A). *The mapping $T_2 : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is maximal monotone, and the set $\operatorname{dom} T_2$, denoted for simplicity by D, contains more than just one point (to avoid trivialities). The mapping $T_1 : D \to \mathbb{R}^n$ is single valued, monotone, and Lipschitz continuous, so in particular the mapping $T = T_1 + T_2$ has effective domain D like T_2. The matrix $H \in \mathbb{R}^{n \times n}$ is symmetric and positive definite (hence nonsingular with H^{-1} symmetric and positive definite), while μ_1 and μ_2 denote constants such that*

$$\left\{ \begin{array}{l} \textit{the mappings } \widetilde{T}_1 = T_1 - \mu_1 H \textit{ and } \widetilde{T}_2 = T_2 - \mu_2 H \textit{ are} \\ \textit{monotone on } D \textit{ with } \mu_1 \geq 0,\ \mu_2 \geq 0,\ \mu_1 + \mu_2 > 0. \end{array} \right. \tag{2.1}$$

Furthermore, $\tilde{\kappa}_1$ is a Lipschitz constant for $\widetilde{T}_1$ on D from $\|\cdot\|_H$ to $\|\cdot\|_{H^{-1}}$:

$$\|\widetilde{T}_1(x') - \widetilde{T}_1(x)\|_{H^{-1}} \ \leq\ \tilde{\kappa}_1 \|x' - x\|_H \quad \textit{for all} \quad x', x \in D. \tag{2.2}$$

Here in parallel to (1.14) we use the notation $\|w\|_{H^{-1}} = \sqrt{\langle w, H^{-1} w\rangle}$. The norm $\|w\|_{H^{-1}}$ is dual to the norm $\|\cdot\|_H$; one has

$$\big\langle x, w \big\rangle \ \leq\ \|x\|_H \|w\|_{H^{-1}} \quad \text{for all} \quad x,\ w \in \mathbb{R}^n. \tag{2.3}$$

Because monotone mappings must be interpreted technically as going from a vector space to its dual, it's natural in (2.2) to match the H-metric on the domain of $\widetilde{T}_1$ with the H^{-1}-metric on the range of $\widetilde{T}_1$.

The monotonicity assumptions in (2.1) correspond (in the face of T_1 being single valued on $D = \operatorname{dom} T_2$) to requiring that

$$\big\langle T_1(x') - T_1(x),\, x' - x \big\rangle \ \geq\ \mu_1 \big\langle x' - x, H[x' - x] \big\rangle \quad \text{for all} \quad x,\ x' \in D,$$

$$\big\langle w' - w,\, x' - x \big\rangle \ \geq\ \mu_2 \big\langle x' - x, H[x' - x] \big\rangle \quad \text{whenever} \quad w \in T_2(x),\ w' \in T_2(x').$$

Because $\mu_1 + \mu_2 > 0$, these inequalities combine to imply that T is strongly monotone with constant $(\mu_1 + \mu_2)\alpha_{\min}$, where $\alpha_{\min}$ stands again for the lowest eigenvalue of H. But this constant of strong monotonicity won't come into play. We'll stay entirely with μ_1 and μ_2 as measures of monotonicity adapted to $\|\cdot\|_H$ rather than to $\|\cdot\|$.

Assumptions (A) in the variational inequality case (1.1) (for a closed convex set C with more than one point and continuous mappings F_1 and F_2 from C to $\mathbb{R}^n$) have $D = C$ and mean that $F_1 - \mu_1 H$ and $F_2 - \mu_2 H$ are monotone on C or, equivalently for $i = 1, 2$, that

$$\big\langle F_i(x') - F_i(x),\, x' - x \big\rangle \geq \mu_i \big\langle x' - x, H[x' - x] \big\rangle \quad \text{when} \quad x,\ x' \in C,$$

while $\widetilde{F}_1 = F_1 - \mu_1 H$ is Lipschitz continuous on C with constant $\tilde{\kappa}_1$ from $\|\cdot\|_H$ to $\|\cdot\|_{H^{-1}}$. For the maximal monotonicity of $T_2 = F_2 + N_C$, see Rockafellar [40, Theorem 3].

The introduction in (A) of a Lipschitz constant not for T_1 but the residual mapping $\widetilde{T}_1 = T_1 - \mu_1 H$ may seem odd, but it's crucial to our strategy of trying to separate the convergence analysis of forward–backward splitting methods from certain "unessential" features of the splitting. This will be clarified in section 3. For practical purposes there's no disadvantage, at least, by virtue of the following fallback estimate.

Proposition 2.1 (Lipschitz estimate). *Suppose κ_1 is a Lipschitz constant for T_1 itself on D from the norm $\|\cdot\|_H$ to the norm $\|\cdot\|_{H^{-1}}$:*

$$\|T_1(x') - T_1(x)\|_{H^{-1}} \;\le\; \kappa_1 \|x' - x\|_H \quad \textit{for all} \quad x', x \in D.$$

Then $\kappa_1 \ge \mu_1$, and the value $\sqrt{\kappa_1^2 - \mu_1^2}$ is a Lipschitz constant for $\widetilde{T}_1 = T_1 - \mu_1 H$ on D with respect to the same norms. Thus, one can always take $\tilde{\kappa}_1 = \sqrt{\kappa_1^2 - \mu_1^2}$ in the absence of anything better.

Proof. By squaring both sides of the Lipschitz inequality given by κ_1, we can write

$$\begin{aligned}
\kappa_1^2 \big\|x' - x\big\|_H^2 \;\ge\; \big\|T_1(x') - T_1(x)\big\|_{H^{-1}}^2 \;&=\; \big\|(\widetilde{T}_1 + \mu_1 H)(x') - (\widetilde{T}_1 + \mu_1 H)(x)\big\|_{H^{-1}}^2 \\
&=\; \big\|\,[\widetilde{T}_1(x') - \widetilde{T}_1(x)] + \mu_1 H[x' - x]\,\big\|_{H^{-1}}^2 \\
&=\; \big\|\widetilde{T}_1(x') - \widetilde{T}_1(x)\big\|_{H^{-1}}^2 + 2\mu_1 \big\langle H[x' - x],\, H^{-1}[\widetilde{T}_1(x') - \widetilde{T}_1(x)]\big\rangle \\
&\qquad + \mu_1^2 \big\langle H[x' - x],\, H^{-1}H[x' - x]\big\rangle \\
&=\; \big\|\widetilde{T}_1(x') - \widetilde{T}_1(x)\big\|_{H^{-1}}^2 + 2\mu_1 \big\langle x' - x,\, \widetilde{T}_1(x') - \widetilde{T}_1(x)\big\rangle + \mu_1^2 \big\|x' - x\big\|_H^2.
\end{aligned}$$

Here $\big\langle x' - x,\, \widetilde{T}_1(x') - \widetilde{T}_1(x)\big\rangle \ge 0$ because $\widetilde{T}_1$ is monotone by assumption. Hence

$$\big\|\widetilde{T}_1(x') - \widetilde{T}_1(x)\big\|_{H^{-1}}^2 \;\le\; (\kappa_1^2 - \mu_1^2)\big\|x' - x\big\|_H^2.$$

Because this holds for all x and x' in D and D has more than one point, it's apparent that $\kappa_1 \ge \mu_1$ and that $\sqrt{\kappa_1^2 - \mu_1^2}$ serves as a Lipschitz constant $\tilde{\kappa}_1$ for $\widetilde{T}_1$ on D. □

We develop next a technical fact which will repeatedly be brought into play.

Proposition 2.2 (inverse Lipschitz continuity from strong monotonicity). *If a mapping $T_0 : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is maximal monotone and $T_0 - \mu_0 H$ is monotone for some $\mu_0 > 0$ where H is symmetric and positive definite, then T_0^{-1} is single valued and Lipschitz continuous, with μ_0^{-1} serving as a Lipschitz constant from the $\|\cdot\|_{H^{-1}}$ metric to the $\|\cdot\|_H$ metric.*

Proof. Whenever $w \in T_0(x)$ and $w' \in T_0(x')$ we have by assumption that

$$\begin{aligned}
0 \;&\le\; \big\langle [w' - \mu_0 H x'] - [w - \mu_0 H x],\, x' - x\big\rangle \\
&=\; \big\langle w' - w,\, x' - x\big\rangle - \mu_0 \big\langle H[x' - x], [x' - x]\big\rangle \\
&\le\; \|x' - x\|_H \|w' - w\|_{H^{-1}} - \mu_0 \|x' - x\|_H^2.
\end{aligned}$$

Thus, $\|x' - x\|_H \le \mu_0^{-1}\|w' - w\|_{H^{-1}}$ whenever $x' \in T_0(w')$ and $x \in T_0^{-1}(w)$, so that $T_0^{-1}(w)$ can't contain more than one point, and T_0^{-1} is Lipschitz continuous on its effective domain with constant μ_0^{-1} and in particular is locally bounded everywhere. But T_0^{-1} inherits maximal monotonicity from T_0, so the latter necessitates T_0^{-1} being nonempty valued everywhere, cf. Rockafellar [41]. □

Theorem 2.3 (algorithmic background). *Under* (A) *the mapping* $T = T_1 + T_2$ *is maximal monotone and also strongly monotone. There is a unique solution* $\bar{x}$ *to* $0 \in T(\bar{x})$*, and for any* $\lambda > 0$ *the iteration mapping* S_λ *is single valued and Lipschitz continuous from the set* $D = \operatorname{dom} T$ *into itself, with unique fixed point* $\bar{x}$.

Proof. Although the single-valued mapping T_1 need not be defined outside of D, it at least has (through Lipschitz continuity) a unique continuous extension T_1' to the closed convex set $C = \operatorname{cl} D$, this extension being monotone and having the same Lipschitz constant as T_1. We can enlarge T_1' to a maximal monotone mapping $T_1'' : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ by defining $T_1''(x) = T_1'(x) + N_C(x)$ when $x \in C$ but $T_1''(x) = \emptyset$ when $x \notin C$, cf. Rockafellar [40, Theorem 3]. Since $\operatorname{dom} T_1'' = C = \operatorname{cl}(\operatorname{dom} T_2)$, the relative interiors of $\operatorname{dom} T_1''$ and $\operatorname{dom} T_2$ have nonempty intersection (they actually coincide). Then, because T_2 like T_1'' is maximal monotone, it follows that $T_1'' + T_2$ is maximal monotone, cf. Rockafellar [40, Theorem 2]. To deduce that T is maximal monotone, it suffices therefore to demonstrate that $T_1''(x) + T_2(x) = T_1(x) + T_2(x)$ for all $x \in C$ or, in other words, that $T_2(x) + N_C(x) \subset T_2(x)$ for all $x \in C$. Unless actually $x \in D$, this holds trivially with both sides empty.

For any $x \in D$ and $w \in T_2(x)$, we have $\langle x' - x,\, w' - w \rangle \geq 0$ whenever $w' \in T_2(x')$; also, for any $u \in N_C(\hat{x})$, we have $\langle x' - x,\, u \rangle \leq 0$ for all $x' \in C$. Consequently, we have $\langle x' - x,\, w' - (w + u) \rangle \geq 0$ whenever $w' \in T_2(x')$. The maximal monotonicity of T_2 then implies $w + u \in T_2(x)$; if it doesn't, the pair $(x, w + u)$ could be added to the graph of T_2 to get a properly larger mapping that is still monotone. Therefore, $T_2(x) + N_C(x) \subset T_2(x)$ for all $x \in D$, as required. Thus, T is maximal monotone.

From the representation $T = (T_1 - \mu_1 H) + (T_2 - \mu_2 H) + (\mu_1 + \mu_2)H$ with $\mu_1 + \mu_2 > 0$, where the first two terms are monotone by assumption, we have $T - (\mu_1 + \mu_2)H$ monotone. Because H is itself strongly monotone, as a consequence of being positive definite, T likewise is strongly monotone. Hence T^{-1} is single valued and Lipschitz continuous by Proposition 2.2. In particular the set $T^{-1}(0)$, which consists of the solutions $\bar{x}$ to $0 \in T(\bar{x})$, has to be a singleton.

Turning now to the properties of S_λ, we observe first that the mapping λT_2, like T_2 itself, is maximal monotone and has the same effective domain as T_2, the relative interior of which meets that of the mapping $x \mapsto Hx$ (namely, $\mathbb{R}^n$). Furthermore, the latter mapping, by virtue of linearity and positive definiteness, is maximal monotone, even strongly monotone. It follows through [40, Theorem 2] that $H + \lambda T_2$ is maximal monotone. Moreover, the mapping $[H + \lambda T_2] - (1 + \lambda\mu_2)H$ is monotone, so by Proposition 2.2 the mapping $(H + \lambda T_2)^{-1}$ must be single valued everywhere and Lipschitz continuous. In fact, it has constant $(1 + \lambda\mu_2)^{-1}$ from $\|\cdot\|_{H^{-1}}$ to $\|\cdot\|_H$. At the same time the mapping $H - \lambda T_1$ is single valued and Lipschitz continuous on D under (A), and, therefore, S_λ, the composite of these two mappings, is of such type as well.

The condition $x = S_\lambda(x)$ corresponds to having $[H - \lambda T_1](x) \in [H + \lambda T_2](x)$ and hence to having $-T_1(x) \in T_2(x)$, which is the same as $0 \in T(x)$. Therefore, the unique fixed point of S_λ on D is the unique $\bar{x}$ with $0 \in T(\bar{x})$. □

Theorem 2.4 (global contraction rate). *Under* (A) *and for any* $\lambda > 0$*, the value*

$$\theta_\lambda = \begin{cases} \dfrac{\sqrt{(1-\lambda\mu_1)^2 + \lambda^2\tilde{\kappa}_1^2}}{1+\lambda\mu_2} & \text{when } \lambda^{-1} \geq \mu_1, \\[2ex] \dfrac{\lambda(\tilde{\kappa}_1 + \mu_1) - 1}{1 + \lambda\mu_2} & \text{when } \lambda^{-1} \leq \mu_1, \end{cases} \tag{2.4}$$

which depends continuously on λ*, is a Lipschitz constant for* $S_\lambda : D \to D$ *as a mapping* *from the $\|\cdot\|_H$ metric to the $\|\cdot\|_H$ metric. In particular,*

$$\|S_\lambda(x)-\bar{x}\|_H \;\le\; \theta_\lambda\|x-\bar{x}\|_H \quad \textit{for all} \quad x\in D,$$

so that S_λ is globally contractive to $\bar{x}$ on D when $\theta_\lambda<1$, which is true for all $\lambda>0$ sufficiently small, specifically if and only if λ is chosen small enough that

$$\lambda^{-1} \;>\; \frac{\mu_1-\mu_2}{2}+\frac{\tilde{\kappa}_1}{2}\max\left\{1,\,\frac{\tilde{\kappa}_1}{\mu_1+\mu_2}\right\}. \tag{2.5}$$

The best such estimated contraction rate, θ_λ, as λ ranges over these choices, is

$$\bar{\theta}=\theta_{\bar{\lambda}}=\frac{1}{\sqrt{1+\left(\dfrac{\mu_1+\mu_2}{\tilde{\kappa}_1}\right)^2}}, \qquad \text{for } \bar{\lambda}=\frac{1}{\left(\dfrac{\tilde{\kappa}_1^2}{\mu_1+\mu_2}\right)+\mu_1}. \tag{2.6}$$

Proof. As already argued in the proof of Proposition 2.2, our assumptions on T_2 in (A) ensure that $(H+\lambda T_2)^{-1}$ is single valued and Lipschitz continuous with constant $(1+\lambda\mu_2)^{-1}$ from $\|\cdot\|_{H^{-1}}$ to $\|\cdot\|_H$. Since $S_\lambda=(H+\lambda T_2)^{-1}(H-\lambda T_1)$, our task in establishing the Lipschitz constant θ_λ for S_λ comes down to showing that the second factor in the formula for θ_λ serves as a Lipschitz constant for $H-\lambda T_1$ on D from $\|\cdot\|_H$ to $\|\cdot\|_{H^{-1}}$. Fix any points x and x' in D. In terms of having $T_1=\widetilde{T}_1+\mu_1 H$, we expand

$$\begin{aligned}
&\big\|(H-\lambda T_1)(\,x')-(H-\lambda T_1)(x)\big\|_{H^{-1}}^2\\
&\quad=\ \big\|\big[(1-\lambda\mu_1)H-\lambda\widetilde{T}_1\big](x')-\big[(1-\lambda\mu_1)H-\lambda T_1\big](x)\big\|_{H^{-1}}^2\\
&\quad=\ \big\|(1-\lambda\mu_1)H[x'-x]-\lambda\big[\widetilde{T}_1(x')-\widetilde{T}_1(x)\big]\,\big\|_{H^{-1}}^2\\
&\quad=\ (1-\lambda\mu_1)^2\big\langle H[x'-x],H^{-1}H[x'-x]\big\rangle\\
&\qquad\qquad -2\lambda(1-\lambda\mu_1)\big\langle H[x'-x],H^{-1}\big[\widetilde{T}_1(x')-\widetilde{T}_1(x)\big]\big\rangle\\
&\qquad\qquad\qquad +\lambda^2\big\langle\big[\widetilde{T}_1(x')-\widetilde{T}_1(x)\big],H^{-1}\big[\widetilde{T}_1(x')-\widetilde{T}_1(x)\big]\big\rangle\\
&\quad=\ (1-\lambda\mu_1)^2\big\|x'-x\big\|_H^2+\lambda^2\big\|\widetilde{T}_1(x')-\widetilde{T}_1(x)\big\|_{H^{-1}}^2\\
&\qquad\qquad -2\lambda(1-\lambda\mu_1)\big\langle x'-x,\,\widetilde{T}_1(x')-\widetilde{T}_1(x)\big\rangle.
\end{aligned} \tag{2.7}$$

At this stage our analysis divides into the cases where $1-\lambda\mu_1\ge 0$ or $1-\lambda\mu_1\le 0$, which correspond to $\lambda^{-1}\ge\mu_1$ or $\lambda^{-1}\le\mu_1$. (When equality holds in these relations the two paths of argument will lead to the same thing.)

In the case where $1-\lambda\mu_1\ge 0$, we can invoke the fact that $\big\langle x'-x,\,\widetilde{T}_1(x')-\widetilde{T}_1(x)\big\rangle\ge 0$ because $\widetilde{T}_1$ is monotone on D. We then get from (2.7) and the specification of $\tilde{\kappa}_1$ that

$$\big\|(H-\lambda T_1)(x')-(H-\lambda T_1)(x)\big\|_{H^{-1}}^2\ \le\ (1-\lambda\mu_1)^2\big\|x'-x\big\|_H^2+\lambda^2\tilde{\kappa}_1^2\big\|x'-x\big\|_H^2,$$

hence $\|(H-\lambda T_1)(x')-(H-\lambda T_1)(x)\|_{H^{-1}}\ \le\ [(1-\lambda\mu_1)^2+\lambda^2\tilde{\kappa}_1^2]^{1/2}\|x'-x\|_H$ in accordance with the first version of θ_λ. In the case where $1-\lambda\mu_1\le 0$ instead, we use the inequality

$$\big\langle x'-x,\,\widetilde{T}_1(x')-\widetilde{T}_1(x)\big\rangle\ \le\ \|x'-x\|_H\|\widetilde{T}_1(x')-\widetilde{T}_1(x)\|_{H^{-1}}\ \le\ \tilde{\kappa}_1\|x'-x\|_H^2$$

from (2.3) to argue through (2.7) that

$$\begin{aligned}
&\big\|(H-\lambda T_1)(x')-(H-\lambda T_1)(x)\big\|^2_{H^{-1}}\\
&\quad\le\ (1-\lambda\mu_1)^2\big\|x'-x\big\|^2_H+\lambda^2\tilde\kappa_1^2\big\|x'-x\big\|^2_H+2\lambda(\lambda\mu_1-1)\tilde\kappa_1\big\|x'-x\big\|^2_H\\
&\quad\le\ \big[(1-\lambda\mu_1)^2+\lambda^2\tilde\kappa_1^2+2\lambda(\lambda\mu_1-1)\tilde\kappa_1\big]\big\|x'-x\big\|^2_H\\
&\quad=\ \big[\lambda(\tilde\kappa_1+\mu_1)-1\big]^2\big\|x'-x\big\|^2_H.
\end{aligned}$$

We obtain $\|(H-\lambda T_1)(x')-(H-\lambda T_1)(x)\|_{H^{-1}}\le[\lambda(\tilde\kappa_1+\mu_1)-1]\|x'-x\|_H$ in accordance with the second version of θ_λ.

In order to understand the nature of the factor θ_λ better, we begin by observing that for λ large enough that $\lambda^{-1}\le\mu_1$, the function $\phi(\lambda)=\theta_\lambda=[\lambda(\tilde\kappa_1+\mu+1)-1]/(1+\lambda\mu_2)$ has $\phi'(\lambda)=[\tilde\kappa_1+\mu_1+\mu_2]/(1+\lambda\mu_2)^2>0$ and hence is an increasing function. In seeking low values of θ_λ we therefore aren't interested in λ with $\lambda^{-1}<\mu_1$ and can concentrate on the case of $\lambda^{-1}\ge\mu_1$, where the other formula holds for θ_λ. Note, though, that

$$\theta_\lambda<1\iff\lambda^{-1}>(\mu_1-\mu_2+\tilde\kappa_1)/2\quad\text{ when }\lambda^{-1}<\mu_1. \tag{2.8}$$

The analysis of θ_λ when $\lambda^{-1}\ge\mu_1$ is simplified by passing temporarily from λ to the parameter

$$\tau=(\lambda^{-1}+\mu_2)^{-1},\quad\text{ which gives }\quad\tau^{-1}=\lambda^{-1}+\mu_2,\quad\lambda^{-1}=\tau^{-1}-\mu_2.$$

The condition $\lambda^{-1}\ge\mu_1$ means $\tau^{-1}\ge\mu$, where we introduce the notation $\mu=\mu_1+\mu_2$ for simplicity. We get

$$\theta_\lambda^2\ =\ \frac{(\lambda^{-1}-\mu_1)^2+\tilde\kappa_1^2}{(\lambda^{-1}+\mu_2)^2}\ =\ \tau^2\big[(\tau^{-1}-\mu)^2+\tilde\kappa_1^2\big]\ =\ 1-2\mu\tau+(\tilde\kappa_1^2+\mu^2)\tau^2. \tag{2.9}$$

From this expression it's obvious that $\theta_\lambda<1$ if and only if $(\tilde\kappa_1^2+\mu^2)\tau^2<2\mu\tau$ or, in other words, $\tau^{-1}>(\tilde\kappa_1^2+\mu^2)/2\mu$. This condition translates back to $\lambda^{-1}>\big[(\tilde\kappa_1^2+\mu^2)/2\mu\big]-\mu_2=\big[\tilde\kappa_1^2/2\mu\big]+(\mu_1-\mu_2)/2$, because $\mu^2-2\mu\mu_2=\mu(\mu_1+\mu_2-2\mu_2)=\mu(\mu_1-\mu_2)$. Thus,

$$\theta_\lambda<1\iff\lambda^{-1}>\frac{\mu_1-\mu_2}{2}+\frac{\tilde\kappa_1^2}{2(\mu_1+\mu_2)}\quad\text{ when }\lambda^{-1}\ge\mu_1. \tag{2.10}$$

The union of (2.8) with (2.10) furnishes the condition claimed in (2.5) for having $\theta_\lambda<1$.

The expression in (2.9) is a strictly convex function of τ which achieves its minimum uniquely when $-2\mu+2(\tilde\kappa_1^2+\mu^2)\tau=0$ or, in other words, the value $\bar\tau=\mu/(\tilde\kappa_1^2+\mu^2)$. This does have the property that $\bar\tau^{-1}\ge\mu$, so the associated step size $\bar\lambda$ satisfies $\bar\lambda^{-1}\ge\mu_1$. The corresponding minimum value for the expression in (2.9) is $\tilde\kappa_1^2/(\tilde\kappa_1^2+\mu^2)$. Therefore, the lowest achievable value for θ_λ is $\theta_{\bar\lambda}=\tilde\kappa_1/\sqrt{\tilde\kappa_1^2+\mu^2}$ for

$$\bar\lambda=1/(\bar\tau^{-1}-\mu_2)=\mu/[(\tilde\kappa_1^2+\mu^2)-\mu\mu_2],$$

which works out to the value claimed for $\bar\lambda$ in the theorem. □

Corollary 2.5 (special rate estimates). *When the estimate $\tilde{\kappa}_1 = \sqrt{\kappa_1^2 - \mu_1^2}$ is used in accordance with Proposition* 2.1, *the corresponding best contraction rate that can be guaranteed is*

$$\theta_{\bar\lambda} = \frac{1}{\sqrt{1 + \dfrac{(\mu_1+\mu_2)^2}{\kappa_1^2 - \mu_1^2}}} \qquad \textit{for } \bar\lambda = \frac{\mu_1+\mu_2}{\kappa_1^2 + \mu_1\mu_2}. \tag{2.11}$$

In the case of $\mu_2 = 0$ this reduces to

$$\theta_{\bar\lambda} = \sqrt{1 - \left(\frac{\mu_1}{\kappa_1}\right)^2} \qquad \textit{for } \bar\lambda = \frac{\mu_1}{\kappa_1^2}. \tag{2.12}$$

Proof. The case in (2.11) is obvious from Theorem 2.4, and the one in (2.12) then follows by elementary algebra in replacing μ_2 by 0. □

The convergence result in Corollary 2.5 was developed in Chen's thesis [37], but Theorem 2.4 itself, with its emphasis on $\tilde{\kappa}_1$ instead of κ_1, appears here for the first time.

An alternative result of Renaud [32, Proposition VI.25] under the assumption that T and T_1^{-1} are strongly monotone gives R-linear convergence. The convergence factor (not necessarily a contraction factor as above) is

$$\frac{1}{\sqrt{1 + \dfrac{\mu\nu_1}{\alpha_{\min}/\alpha_{\max}}}} \qquad \text{for } \lambda = \nu_1\alpha_{\min}, \tag{2.13}$$

where μ and ν_1 are strong monotonicity constants for T and T_1^{-1} in the sense of (1.12) and (1.13) (i.e., calibrated by I instead of H), and $\alpha_{\min}$ and $\alpha_{\max}$ are the smallest and biggest eigenvalues of H. (Here we specialize to $\mathbb{R}^n$; Renaud operated in the context of a possibly infinite-dimensional Hilbert space.) Renaud didn't actually require μ to be a strong monotonicity constant in the full sense of (1.12) but just a value satisfying

$$\langle w - \bar{w},\, x - \bar{x}\rangle \geq \mu\|x - \bar{x}\|^2 \quad \text{if} \quad w \in T(x), \quad \text{where} \quad \bar{w} = 0 \in T(\bar{x}). \tag{2.14}$$

Likewise this would suffice in Theorem 2.4 if we aimed at Q-linear convergence to $\bar{x}$ instead of insisting that S_λ be a contraction mapping; see section 4.

The dependence of Renaud's factor in (2.13) on $\alpha_{\min}/\alpha_{\max}$, which is the condition number of H, should be noted. This is disadvantageous unless $H = I$, so the condition number is 1; see section 3. When $H = I$ and T_1 is strongly monotone, it's possible under our assumptions to take $\nu_1 = \mu_1/\kappa_1^2$. Then Renaud's factor in (2.13) becomes

$$\frac{1}{\sqrt{1 + \dfrac{\mu_1(\mu_1+\mu_2)}{\kappa_1^2}}} \qquad \text{for } \lambda = \frac{\mu_1}{\kappa_1^2},$$

which isn't as sharp as our factor in Corollary 2.5. On the other hand, if $\nu_1 > 0$ is known directly one can take $\kappa_1 = 1/\nu_1$ and get $1/\sqrt{1 + (\mu_1+\mu_2)\nu_1}$ in (2.13) in comparison to $1/\sqrt{1 + [(\mu_1+\mu_2)\nu_1]^2}$ in Corollary 2.5, where $\mu_1\nu_1 \leq 1$ but perhaps $(\mu_1+\mu_2)\nu_1 > 1$.

3. Utilization of strong monotonicity. A major purpose of our analysis has been to gain insight into how a splitting can be set up advantageously. In expressing T as a sum $T_1 + T_2$, there may be terms that could be assigned either to T_1 or to T_2 without creating an obstacle to the implementation of the forward–backward method. What approach is best in enhancing convergence?

Let's focus on shifts of positive monotonicity. On the basis of (A) we can write $T = \widetilde{T}_1 + \widetilde{T}_2 + \mu H$ for $\widetilde{T}_1 = T_1 - \mu_1 H$, $\widetilde{T}_2 = T_2 - \mu_2 H$, and $\mu = \mu_1 + \mu_2$. Here $\widetilde{T}_1$ and $\widetilde{T}_2$ are maximal monotone (for if they are not, that would mean the graph of one of them, say $\widetilde{T}_1$, could be enlarged without destroying monotonicity, in which case the same would be true for $\widetilde{T}_1 + \mu_1 H = T_1$, contrary to the maximality of T_1).

Suppose we were to divide up μ in a different way, i.e., $\mu = \mu'_1 + \mu'_2$ with $\mu'_1 \geq 0$ and $\mu'_2 \geq 0$, and set $T'_1 = \widetilde{T}_1 + \mu'_1 H$ and $T'_2 = \widetilde{T}_2 + \mu'_2 H$. This would give a different splitting, $T = T'_1 + T'_2$, in which T'_1 and T'_2 are again maximal monotone. Could there be any advantage in this for the algorithm's performance when implemented with the matrix H?

The answer is *no* as long as the optimal step size prescription of Theorem 2.4 is employed. This is clear from the fact that the optimal contraction rate $\bar{\theta}$ in (2.6) depends only on $\tilde{\kappa}_1$ and the sum $\mu_1 + \mu_2$ and therefore would be the same under the different splitting, since $\mu'_1 + \mu'_2 = \mu_1 + \mu_2$ and even $T'_1 - \mu'_1 H = \widetilde{T}_1 = T_1 - \mu_1 H$ (so κ is unaffected). Indeed, the contraction rate has been optimized in Theorem 2.4 with respect to the whole range of splittings that we are looking at. In using the step size $\bar{\lambda}$ prescribed for the splitting $T = T_1 + T_2$, one is able *automatically* to capture whatever algorithmic advantages may lie in this direction. Although the step sizes for the splittings $T = T_1 + T_2$ and $T = T'_1 + T'_2$ are given differently as

$$\bar{\lambda} = \frac{1}{\left(\dfrac{\tilde{\kappa}_1^2}{\mu_1 + \mu_2}\right) + \mu_1}, \qquad \bar{\lambda}' = \frac{1}{\left(\dfrac{\tilde{\kappa}_1^2}{\mu'_1 + \mu'_2}\right) + \mu'_1}$$

and may not themselves be the same, they necessarily result in the same optimal rate $\bar{\theta}$.

But a subtle distinction must be noted between Theorem 2.4 and Corollary 2.5. If the tactic in developing a Lipschitz constant for $\widetilde{T}_1$ were to use an estimate based on Proposition 2.1, the answer to the question posed would instead be *yes*!

The reason is that in passing from $T = T_1 + T_2$ to $T = T'_1 + T'_2$, such an estimate $\tilde{\kappa}_1 = \sqrt{\kappa_1^2 - \mu_1^2}$, where κ_1 is a Lipschitz constant for T_1, would be replaced by a *different* value $\tilde{\kappa}'_1 = \sqrt{{\kappa'_1}^2 - {\mu'_1}^2}$, where κ'_1 is a Lipschitz constant for T'_1 (relative to the specified norms). Then not only would the corresponding step sizes $\bar{\lambda}$ and $\bar{\lambda}'$, as dictated by (2.11), be different, they would result in different contraction rates: $\theta_{\bar{\lambda}} \neq \theta_{\bar{\lambda}'}$ in (2.11). The issue would arise of determining which splitting $T = T'_1 + T'_2$ minimizes $\sqrt{{\kappa'_1}^2 - {\mu'_1}^2}$ and thus furnishes the best contraction rate. Actually, we know from Proposition 2.1 that the minimum is achieved when $T'_1 = \widetilde{T}_1 = T_1 - \mu_1 H$, $T'_2 = \widetilde{T}_2 + \mu H = T_2 + \mu_1 H$. Thus, if we were to rely on the result in Corollary 2.5 rather than the one in Theorem 2.4, as for instance in [37], the optimal splitting would be obtained by extracting all possible strong monotonicity from T_1 and reassigning it to T_2, a qualitatively very different conclusion.

This highlights the contrast between the technique adopted here and previous research, which has utilized a Lipschitz constant for T_1 itself (moreover one in terms of the canonical norm only), not to speak of concentrating on strong monotonicity of

T_1. Through Theorem 2.4 we can optimally exploit strong monotonicity of T_1 or T_2 or both, without having to switch any terms in the splitting in the end.

The idea is illustrated by its application to solving variational inequalities.

THEOREM 3.1 (application to projection algorithms). *Consider the variational inequality problem* (1.1) *in the case of a nonempty, closed convex set $C \subset \mathbb{R}^n$ and a continuous, single-valued mapping $F : C \to \mathbb{R}^n$. Let H be a symmetric positive definite matrix, and let $\mu > 0$ be a constant such that F satisfies the strong monotonicity condition*

$$\big\langle F(x') - F(x),\, x' - x \big\rangle \;\geq\; \mu \, \|x' - x\|_H^2 \quad \textit{for all} \quad x,\ x' \in C. \tag{3.1}$$

Let $\tilde{\kappa} \geq 0$ be a Lipschitz constant for $\widetilde{F} = F - \mu H$ on C from the norm $\|\cdot\|_H$ to the norm $\|\cdot\|_{H^{-1}}$. Then in applying Theorem 2.4 *to the splitting $T = T_1 + T_2$ for $T_1 = F$, $T_2 = N_C$, and with $\mu_1 = \mu$, $\mu_2 = 0$, the optimal contraction rate is*

$$\bar{\theta} = \theta_{\bar{\lambda}} = \frac{1}{\sqrt{1 + (\mu/\tilde{\kappa})^2}} \qquad \textit{for } \bar{\lambda} = \frac{\mu}{\tilde{\kappa}^2 + \mu^2}. \tag{3.2}$$

No alternative splitting $T = T_1' + T_2'$ in the mode of $T_1' = F - \tau H$ and $T_2' = \tau H + N_C$ for some $\tau \in (0, \mu]$ can provide a better contraction rate through Theorem 2.4.

Proof. This is evident from the preceding remarks. The assumptions furnish a specialization of the conditions in (A) to the special case in question. □

The fact that under the circumstances described, execution of the forward–backward splitting method as a projection method is just as good as any alternative execution obtainable by shifting the strong monotonicity from the "forward" part to the "backward" part of the iteration mapping, is perhaps surprising. But again, it must be remembered that this result depends on utilizing a Lipschitz constant $\tilde{\kappa}$ for $\widetilde{F} = F - \mu H$ rather than a constant κ attached directly to F itself.

COROLLARY 3.2. *When the estimate $\tilde{\kappa} = \sqrt{\kappa^2 - \mu^2}$ is used in Corollary* 3.2 *in accordance with Proposition* 2.1, *κ being a Lipschitz constant for F on C from $\|\cdot\|_H$ to $\|\cdot\|_{H^{-1}}$, the corresponding best contraction rate that can be guaranteed for the projection algorithm is*

$$\theta_{\bar{\lambda}} = \sqrt{1 - \left(\frac{\mu}{\kappa}\right)^2} \qquad \text{for } \bar{\lambda} = \frac{\mu}{\kappa^2}. \tag{3.3}$$

Proof. This applies the second part of Corollary 2.5. □

For the case of $H = I$, results related to Corollary 3.2 were obtained recently by Renaud. He noted in [32, p. 143] the contraction rate in (3.3) and went on to demonstrate Q-linear convergence, although not the full contraction property, under the assumption that F^{-1} is strongly monotone with constant $\nu > 0$. The factor is then

$$\frac{1}{\sqrt{1 + \mu\nu}} \qquad \text{for } \lambda = \nu,$$

cf. [32, Proposition VI.2]. This alternative assumption is satisfied when F is Lipschitz continuous with constant κ (from $\|\cdot\|$ to $\|\cdot\|$), namely with $\nu = \mu/\kappa^2$, and Renaud's factor reduces then to ours.

For the general case where $H \neq I$, the contraction rate in Corollary 3.2 may be compared for the one derived for projection algorithms by Dafermos [7]. In effect she got

$$\sqrt{1-\frac{\mu^2}{\beta_1^2\beta_2^2\mathrm{cond}(H)}}, \tag{3.4}$$

where $\mathrm{cond}(H)$ is the condition number of H (its highest eigenvalue divided by its lowest eigenvalue), β_1 is a conversion factor from $\|\cdot\|_H$ to $\|\cdot\|$, and β_2 is a Lipschitz constant for F from $\|\cdot\|$ to $\|\cdot\|_{H^{-1}}$, so that $\beta_1\beta_2$ is an (upper) estimate for the Lipschitz constant κ in Corollary 3.2. Unless $H = I$, Dafermos' denominator in (3.4) has to be greater than ours in (3.3), and her contraction factor accordingly has to be nearer to 1, thus not as good. The dependence of (3.4) on the condition number for H illustrates very well the unwarranted consequences of bringing in the canonical norm $\|\cdot\|$ instead of sticking consistently with the method's intrinsic geometry. The canonical norm is irrelevant in this appraisal of algorithmic performance.

THEOREM 3.3 (application to affine variational inequalities). *Consider the variational inequality problem* (1.1) *in the case of a nonempty, closed convex set $C \subset \mathbb{R}^n$ and an affine mapping $F(x) = Mx+q$. Let $M_s = \frac{1}{2}(M+M^\top)$ and $M_a = \frac{1}{2}(M-M^\top)$ be the symmetric and antisymmetric parts of the matrix M, and suppose M_s is positive definite. Take $H = M_s$ and define (with the canonical matrix norm)*

$$\mathrm{skew}(M) \;=\; \|M_s^{-1/2}M_aM_s^{-1/2}\|. \tag{3.5}$$

Then Theorem 3.1 *applies with $\mu = 1$ and $\tilde\kappa_1 = \mathrm{skew}(M)$, which is the minimal Lipschitz constant for this case. The projection algorithm thus attains the global contraction rate*

$$\bar\theta = \theta_{\bar\lambda} = \frac{1}{\sqrt{1+\dfrac{1}{\mathrm{skew}(M)^2}}} \qquad \textit{for } \bar\lambda = \frac{1}{1+\mathrm{skew}(M)^2}. \tag{3.6}$$

Proof. Here $\widetilde{T}_1(x) = (F-\mu H)(x) = M_ax+q$, an affine monotone mapping devoid of strong monotonicity. We must verify that the specified value of $\tilde\kappa_1$ serves as the minimal Lipschitz constant for this mapping from the norm $\|\cdot\|_H$ to the norm $\|\cdot\|_{H^{-1}}$. The square of the required constant is the supremum of the quotient

$$\frac{\big\|\widetilde{T}_1(x')-\widetilde{T}_1(x)\big\|_{H^{-1}}^2}{\|x'-x\|_H^2} = \frac{\big\langle M_a[x'-x], M_s^{-1}M_a[x'-x]\big\rangle}{\big\langle [x'-x], M_s[x'-x]\big\rangle} = \frac{\big\|M_s^{-1/2}M_a[x'-x]\big\|^2}{\big\|M_s^{1/2}[x'-x]\big\|^2}$$

over all x and x' with $x' \neq 0$. Through the change of variables $u = M_s^{1/2}[x'-x]$, giving $[x'-x] = M_s^{-1/2}u$, we see that the constant is the supremum of the expression $\big\|M_s^{-1/2}M_aM_s^{-1/2}u\big\|/\|u\|$ over all $u \neq 0$, and this is $\|M_s^{-1/2}M_aM_s^{-1/2}\|$. □

The value $\mathrm{skew}(M) \in (0,\infty)$ in (3.5) intrinsically measures the *skewness* of the matrix M. Obviously

$$\mathrm{skew}(M) \leq \|M_a\|/\|M_s\| \tag{3.7}$$

in particular, but the right side of this inequality is dependent on the "conditioning" of M with respect to the canonical norm, whereas $\mathrm{skew}(M)$ itself isn't. The smaller $\mathrm{skew}(M)$ is, the nearer M is to being symmetric and the better the rate of convergence that is assured for the solution method addressed by Theorem 3.3. Of course, this realization of forward–backward splitting is practical only when it's easy to project onto C with respect to the norm induced by M_s as H, but that does cover many applications in which C has a product structure matched by a box-diagonal pattern of M_s, as in [20].

Dupuis and Darveau [28], in building on the result of Dafermos [7], likewise obtained for the affine variational inequality case of projection algorithms a contraction factor incorporating the value $\|M_s^{-1/2}M_aM_s^{-1/2}\|$. But the factor they got resembles the one in (3.4) in being the square root of an expression that depends in part on the condition number of H. In contrast to our contraction factor in (3.3), it doesn't tend to 0 as M approaches symmetry and the implementation matrix $H = M_s$ coalesces with M. Again, the cost of deviating from the underlying geometry is evident.

The result in Theorem 3.3 can best be compared with a recent result of Zanni [31] for the same method. He obtains the rate

$$\sqrt{1 - \frac{1}{\left\|M_s^{-1/2}MM_s^{-1/2}\right\|^2}} \quad \text{for } \lambda = \frac{1}{\left\|M_s^{-1/2}MM_s^{-1/2}\right\|^2}, \tag{3.8}$$

which he elaborates by the estimate

$$\left\|M_s^{-1/2}MM_s^{-1/2}\right\| \le 1 + \operatorname{cond}(M_s)\,\frac{\|M_a\|}{\|M_s\|}, \tag{3.9}$$

taking the ratio $\|M_a\|/\|M_s\|$ as a measure of skewness. The appearance of M instead of M_a in (3.8) can be seen as reflecting a reliance on a Lipschitz constant for M instead of for M_a; this parallels the difference between Corollary 3.2 and Theorem 3.1. The estimate in (3.9) suffers from dependence on translation to the canonical norm, but to avoid this it could be replaced by

$$\begin{aligned}\left\|M_s^{-1/2}MM_s^{-1/2}\right\| &= \left\|M_s^{-1/2}(M_s + M_a)M_s^{-1/2}\right\| \\ &\le \left\|M_s^{-1/2}M_sM_s^{-1/2}\right\| + \left\|M_s^{-1/2}M_aM_s^{-1/2}\right\| = 1 + \operatorname{skew}(M).\end{aligned}$$

Even so, it wouldn't yield the lower contraction factor in Theorem 3.3.

Yet another measure of skewness was introduced by Marcotte and Guélat [42] for the special context of solving problems of traffic equilibrium. This differs from ours in being localized to the solution point $\bar{x}$ and dependent on the vector q as well as on the submatrices M_s and M_a. These authors nonetheless demonstrate through numerical testing of several algorithms an empirical relationship between skewness and difficulty of solvability such as appears in Theorem 3.3.

For projected *gradient* algorithms, where $F = \nabla f$ for a $\mathcal{C}^2$ function f with bounded Hessians $\nabla^2 f(x)$, better contraction estimates can be given than are obtainable by specializing the ones here; see Polyak [43].

4. Local convergence analysis. Our efforts so far have gone into identifying a rate of linear convergence that's effective immediately from any starting point x_0 for a forward–backward splitting method. There is interest too, of course, in knowing what might be possible with convergence as the solution $\bar{x}$ is neared. For this purpose we don't have to start building up a broader theory but can make use of the results we already have. Although Theorem 2.4 presents a contraction rate relative to the entire set $D = \operatorname{dom} T$, its formulation already allows us to deduce local contraction rates in a neighborhood of $\bar{x}$.

Theorem 4.1 (local contraction rates). *Let U be an open ball around $\bar{x}$ with respect to the norm $\|\cdot\|_H$, and let $\hat{\mu}_1$, $\hat{\mu}_2$, and $\hat{\kappa}_1$ be constants as in* (A) *but relative to $D \cap U$ in place of D. Then, as long as $\lambda > 0$ is small enough that*

$$\lambda^{-1} > \frac{\hat{\mu}_1 - \hat{\mu}_2}{2} + \frac{\hat{\kappa}_1}{2} \max\left\{1, \frac{\hat{\kappa}_1}{\hat{\mu}_1 + \hat{\mu}_2}\right\}, \tag{4.1}$$

the mapping S_λ carries $D \cap U$ into $D \cap U$, and the conclusions of Theorem 2.4 *hold for this localization of S_λ but with $\hat{\mu}_1$, $\hat{\mu}_2$, and $\hat{\kappa}_1$ in place of μ_1, μ_2, and $\tilde{\kappa}_1$.*

Proof. Taking $C = \operatorname{cl} U$, define $\widehat{T}_2 = T_2 + N_C$. This mapping, like T_2, is maximal monotone; cf. [39, Theorem 2]. Proposition 2.1 and Theorem 2.4 are applicable to $\widehat{T} = T_1 + \widehat{T}_2$ with respect to the constants $\hat{\mu}_1$, $\hat{\mu}_2$, and $\hat{\kappa}_1$ on $\widehat{D} = \operatorname{dom} \widehat{T}_2 = D \cap C$. In particular, $\widehat{T}^{-1}(0)$ must be a singleton, but because $\bar{x}$ belongs to the interior of C, we have $N_C(\bar{x}) = \{0\}$ and $\widehat{T}(\bar{x}) = T(\bar{x})$. Hence $\widehat{T}^{-1}(0) = \{\bar{x}\}$, and the contraction properties given by Theorem 2.4 for the mapping $\widehat{S}_\lambda = (H + \lambda \widehat{T}_2)^{-1}(H - \lambda T_1)$ must refer to this same $\bar{x}$. Distances from $\bar{x}$ can then only be decreased under $\widehat{S}_\lambda$, so $\widehat{S}_\lambda$ must carry $D \cap U$ into itself.

Consider now any $x \in D \cap U$ and let $w = \widehat{S}_\lambda(x)$. As just seen, we have $w \in D \cap U$, which implies that w belongs to the interior of C, so $N_C(w) = \{0\}$. From the definition of $\widehat{S}_\lambda$ we see that

$$(H - \lambda T_1)(x) \in (H + \lambda \widehat{T}_2)(w) = (H + \lambda T_2)(w) + N_C(w) = (H + \lambda T_2)(w),$$

hence, in fact, $w = (H + \lambda T_2)(H - \lambda T_1)(x) = S_\lambda(x)$. This shows that $\widehat{S}_\lambda$ agrees with S_λ on $D \cap U$. The conclusions about the behavior of $\widehat{S}_\lambda$ on $D \cap U$ therefore translate to ones about S_λ. □

The proof of Theorem 2.4 reveals a way of refining that result and with it Corollary 2.5 and Theorem 4.1. Although the monotonicity of $T_2 - \mu_2 H$ is fully utilized in obtaining a Lipschitz constant for the factor $(H + \lambda T_2)^{-1}$ of S_λ, the assumptions in (A) about μ_1 and $\tilde{\kappa}_1$ could be weakened if instead of asking for S_λ to be contractive on D we merely asked for a bound in $[0, 1)$ on the ratios $\|S_\lambda(x) - S_\lambda(\bar{x})\|_H / \|x - \bar{x}\|_H$. The key is just to observe that if the argument for estimating $\|(H - \lambda T_1)(x') - (H - \lambda T_1)(x)\|_{H^{-1}}$ is applied only to $\|(H - \lambda T_1)(x) - (H - \lambda T_1)(\bar{x})\|_{H^{-1}}$, all that one needs from μ_1 and $\tilde{\kappa}_1$ is that the mapping $\widetilde{T}_1 = T_1 - \mu_1 H$ satisfies

$$\left\langle x - \bar{x}, \widetilde{T}_1(x) - \widetilde{T}_1(\bar{x}) \right\rangle \geq 0, \qquad \left\|\widetilde{T}_1(x) - \widetilde{T}_1(\bar{x})\right\|_{H^{-1}} \leq \tilde{\kappa}_1 \left\|x - \bar{x}\right\|_H.$$

Likewise, under these inequalities the estimate in Proposition 2.1 remains valid with respect to a constant κ_1 merely satisfying

$$\left\|T_1(x) - T_1(\bar{x})\right\|_{H^{-1}} \leq \kappa_1 \left\|x - \bar{x}\right\|_H.$$

This refinement appears to offer little advantage in general over the global picture in Theorem 2.4, inasmuch as special properties of T_1 and T_2 around the solution point $\bar{x}$, in contrast to other points, can hardly be available in advance of calculating $\bar{x}$, which threatens a kind of circularity. Indeed, if the assumption on μ_1 is weakened, the very existence and uniqueness of $\bar{x}$ could be thrown into doubt, because Theorem 2.3 might no longer be applicable. Yet in the localized context of Theorem 4.1, the refinement does at least furnish insights into what might be expected of the rate of convergence in the tail of a forward–backward sequence as x_k nears $\bar{x}$. The following is what we get.

Theorem 4.2 (ultimate linear convergence rate). *Assuming* (A)*, define the constants $\bar\mu_1 \geq 0$, $\bar\mu_2 \geq 0$, and $\bar\kappa_1 \geq 0$ at the unique solution point $\bar x$ by*

$$\begin{aligned}\bar\mu_2 &= \limsup_{\substack{w\in T_2(x),\, w'\in T_2(x')\\ x,x'\to\bar x,\, x'\neq x}} \frac{\langle x'-x,\, w'-w\rangle}{\|x'-x\|_H^2},\\ \bar\mu_1 &= \limsup_{\substack{x\to\bar x\\ x\in D,\, x\neq\bar x}} \frac{\langle x-\bar x,\, T_1(x)-T_1(\bar x)\rangle}{\|x-\bar x\|_H^2},\\ \bar\kappa_1 &= \limsup_{\substack{x\to\bar x\\ x\in D,\, x\neq\bar x}} \frac{\|\overline{T}_1(x)-\overline{T}_1(\bar x)\|_{H^{-1}}}{\|x-\bar x\|_H} \quad \textit{for} \quad \overline{T}_1 = T_1 - \bar\mu_1 H,\end{aligned}$$

necessarily obtaining $\bar\mu_1 \geq \mu_1$, $\bar\mu_2 \geq \mu_2$, and $\bar\kappa_1 \leq \tilde\kappa_1$; in fact, $\bar\kappa_1 \leq \sqrt{\tilde\kappa_1^2 - (\bar\mu_1-\mu_1)^2}$ (hence $\bar\mu_1 \leq \mu_1 + \tilde\kappa_1$). Also define

$$\gamma = \liminf_{\substack{(x,u)\to(\bar x, T_1(\bar x))\\ -u\in T_2(x)}} \frac{\|u-T_1(\bar x)\|_{H^{-1}}}{\|x-\bar x\|_H},$$

necessarily obtaining $\gamma \geq \bar\mu_2$. Then for any step size $\lambda > 0$, the sequence of points x_k generated by $x_k = S_\lambda(x_{k-1})$ from any starting point $x_0 \in D$ will satisfy

$$\limsup_{k\to\infty} \frac{\|x_k-\bar x\|_H}{\|x_{k-1}-\bar x\|_H} \leq \begin{cases} \dfrac{\sqrt{(1-\lambda\bar\mu_1)^2+\lambda^2\bar\kappa_1^2}}{\sqrt{1+2\lambda\bar\mu_2+\lambda^2\gamma^2}} & \textit{when } \lambda^{-1} \geq \bar\mu_1,\\[2ex] \dfrac{\lambda(\bar\kappa_1+\bar\mu_1)-1}{\sqrt{1+2\lambda\bar\mu_2+\lambda^2\gamma^2}} & \textit{when } \lambda^{-1} \leq \bar\mu_1. \end{cases} \tag{4.2}$$

In particular, this holds for the step size $\bar\lambda$ identified in (2.6) *as optimal relative to the globally estimated constants μ_1, μ_2, and $\tilde\kappa_1$.*

Proof. It's clear from (A) that $\bar\mu_1 \geq \mu_1$ and $\bar\mu_2 \geq \mu_2$, since the monotonicity of $T_i - \mu_i H$ on D corresponds to having $\langle x'-x, T_i(x')-T_i(x)\rangle \geq \mu_i\|x'-x\|_H^2$ for $x, x' \in D$. The verification that $\bar\kappa_1 \leq \tilde\kappa_1$ takes more effort. It relies indirectly on the observation above that Proposition 2.1 stays valid when the context is that of points x compared to $\bar x$ rather than general pairs x and x'. If $\bar\mu_1 = \mu_1$, we have $\overline{T}_1 = \widetilde{T}_1$ and the inequality $\bar\kappa_1 \leq \tilde\kappa_1$ is elementary from the definitions, so we can concentrate on the case where $\bar\mu_1 > \mu_1$.

Consider any $\delta \in (0, \bar\mu_1 - \mu_1)$. From the definition of $\bar\mu_1$ there's a neighborhood Z of $\bar x$ consisting of points x for which $\langle x-\bar x,\, T_1(x)-T_1(\bar x)\rangle \geq (\bar\mu_1-\delta)\|x-\bar x\|_H^2$. This inequality means that for the mapping $\overline{T}_1^{\delta} = T_1 - (\bar\mu_1-\delta)H = \overline{T}_1 - \delta H$,

$$\langle x-\bar x,\, \overline{T}_1^{\delta}(x)-\overline{T}_1^{\delta}(\bar x)\rangle \geq 0 \quad \text{for } x \in D\cap Z. \tag{4.3}$$

But $\overline{T}_1^{\delta} = \widetilde{T}_1 - \tau H$ for $\tau = \bar\mu_1 - \mu_1 - \delta > 0$. It follows from applying the extended version of Proposition 2.1 to this relation in light of (4.3) that

$$\|\overline{T}_1^{\delta}(x)-\overline{T}_1^{\delta}(\bar x)\|_{H^{-1}} \leq \sqrt{\tilde\kappa_1^2-\tau^2}\,\|x-\bar x\|_H \quad \text{for } x \in D\cap Z.$$

On the other hand, since $\overline{T}_1^{\delta} = \overline{T}_1 - \delta H$ and $\|H[x-\bar x]\|_{H^{-1}} = \|H(x-\bar x)\|_{H^{-1}} = \|x'-x\|_H$, we know that $\|\overline{T}_1(x)-\overline{T}_1(\bar x)\|_{H^{-1}} \leq \|\overline{T}_1^{\delta}(x)-\overline{T}_1^{\delta}(\bar x)\|_{H^{-1}} + \delta\|x-\bar x\|_H$.

This tells us that

$$\big\|\overline{T}_1(x)-\overline{T}_1(\bar{x})\big\|_{H^{-1}} \le \left(\delta+\sqrt{\tilde{\kappa}_1^2-(\bar{\mu}_1-\mu_1-\delta)^2}\right)\big\|x-\bar{x}\big\|_H \quad \text{for } x\in D\cap Z.$$

Taking the limit in the definition of $\bar{\kappa}_1$ and using the fact that a neighborhood Z like this exists for any $\delta>0$, we obtain $\bar{\kappa}_1 \le \sqrt{\tilde{\kappa}_1^2-(\bar{\mu}_1-\mu_1)^2}$; hence, $\bar{\kappa}_1\le\tilde{\kappa}_1$ because $\bar{\mu}_1\ge\mu_1$.

We look next at the claims about γ and θ^*_λ, the latter being the symbol by which we'll denote the right side of (4.2). For any $\epsilon>0$, let $\hat{\mu}_i=\max\{\mu_i,\bar{\mu}_i-\epsilon\}$ for $i=1,2$ and $\hat{\kappa}_1=\min\{\tilde{\kappa}_1,\bar{\kappa}_1+\epsilon\}$. On the basis of the definitions we know there's a ball U around $\bar{x}$ with respect to the norm $\|\cdot\|_H$ such that

$$\begin{cases} \big\langle x'-x,\, w'-w\big\rangle \ge \hat{\mu}_2\|x'-x\|_H & \text{if } x,x'\in D\cap U,\ w\in T_2(w),\ w'\in T_2(x'),\\ \big\langle x-\bar{x},\, T_1(x)-T_1(\bar{x})\big\rangle \ge \hat{\mu}_1\|x-\bar{x}\|_H & \text{if } x\in D\cap U,\\ \|\overline{T}_1(x)-\overline{T}_1(\bar{x})\|_{H^{-1}} \le \hat{\kappa}_1\|x-\bar{x}\|_H & \text{if } x\in D\cap U. \end{cases}$$

We are then in the framework of the extended version of Theorem 4.1 and are able to see that $\limsup_k \|x_k-\bar{x}\|_H\big/\|x_{k-1}-\bar{x}\|_H \le \widehat{\theta}_\lambda$, the latter being the same as θ_λ except that $\hat{\mu}_1$, $\hat{\mu}_2$, and $\hat{\kappa}_1$ replace μ_1, μ_2, and $\tilde{\kappa}_1$. An improvement can be made, however, in taking advantage of the constant γ.

Let $\hat{\gamma}=\max\{0,\gamma-\epsilon\}$. From the definition of γ there's a neighborhood V of $T_1(\bar{x})$ with respect to the norm $\|\cdot\|_{H-1}$ such that, when the ball U is small enough, we have

$$(4.4)\qquad \big\|u-T_1(\bar{x})\|_{H^{-1}} \ge \hat{\gamma}\|x-\bar{x}\|_H^2 \quad \text{when} \quad x\in D\cap U,\ u\in V,\ -u\in T_2(x).$$

However, the point $\bar{u}=T_1(\bar{x})$ also satisfies $-\bar{u}\in T_2(\bar{x})$ because $0\in T(\bar{x})=T_1(\bar{x})+T_2(\bar{x})$. This implies from earlier that

$$\hat{\mu}_2\|x-\bar{x}\|_H^2 \ \le\ \big\langle -u+\bar{u},\, x-\bar{x}\big\rangle \ \le\ \|x-\bar{x}\|_H\|u-\bar{u}\|_{H^{-1}},$$

so $\hat{\mu}_2\|x-\bar{x}\|_H \le \|u-\bar{u}\|_{H^{-1}}$. Therefore, $\hat{\mu}_2\le\hat{\gamma}$, which establishes $\bar{\mu}_2\le\gamma$ through the arbitrariness of ϵ in the definition of $\hat{\mu}_2$ and $\hat{\gamma}$.

Let $\bar{w}=(H-\lambda T_1)(\bar{x})=H\bar{x}-\lambda T_1(\bar{x})$. Since

$$\bar{x}=S_\lambda(\bar{x})=(H+\lambda T_2)^{-1}(H-\lambda T_1)(\bar{x}),$$

we have $\bar{x}=(H+\lambda T_2)^{-1}(\bar{w})$. Consider along with this any elements w and x with $x=(H+\lambda T_2)^{-1}(w)$. For these, the set $T_2^{-1}(x)$ contains $\lambda^{-1}[w-Hx]$, whereas $T_2^{-1}(\bar{x})$ contains $\lambda^{-1}[\bar{w}-H\bar{x}]$, the latter being just $-T_1(\bar{x})$. When w is close to $\bar{w}$, not only does x lie in the ball U around $\bar{x}$, due to the continuity of $(H+\lambda T_2)^{-1}$, but also the vector $u=-\lambda^{-1}[w-Hx]$ lies in the neighborhood V of $\bar{u}=-T_1(\bar{x})$. Then by (4.4),

$$\begin{aligned}\hat{\gamma}^2\|x-\bar{x}\|_H^2 \ &\le\ \big\|u-T_1(\bar{x})\|_{H^{-1}}^2 \ =\ \big\|-\lambda^{-1}[w-Hx]+\lambda^{-1}[\bar{w}-H\bar{x}]\big\|_{H^{-1}}^2\\ &=\ \lambda^{-2}\|w-\bar{w}\|_{H^{-1}}^2-2\lambda^{-2}\big\langle w-\bar{w},\, x-\bar{x}\big\rangle+\|x-\bar{x}\|_H^2\\ &\le\ \lambda^{-2}\|w-\bar{w}\|_{H^{-1}}^2-2\lambda^{-2}\hat{\mu}_2\|x-\bar{x}\|_H^2+\lambda^{-2}\|x-\bar{x}\|_H^2,\end{aligned}$$

where the last inequality invokes the property arranged for $\hat{\mu}_2$. Rearranging, we obtain $\|x-\bar{x}\|_H^2\le[1+\lambda\hat{\mu}_2+\lambda^2\hat{\gamma}^2]\|w-\bar{w}\|_{H^{-1}}^2$. This shows that the factor $(1+\lambda\hat{\mu}_2)^{-1}$ in $\widehat{\theta}_\lambda$ can be replaced by $(1+\lambda\hat{\mu}_2+\lambda^2\hat{\gamma}^2)^{-1/2}$, which if anything is lower.

It remains only to observe that, having demonstrated that this modified factor $\widehat{\theta}_\lambda$ operates in terms of $\hat{\mu}_1$, $\hat{\mu}_2$, $\hat{\kappa}_1$, and $\hat{\gamma}$ as defined for arbitrary $\epsilon>0$, we must in the limit as $\epsilon\searrow 0$ get the factor θ^*_λ corresponding to $\bar{\mu}_1$, $\bar{\mu}_2$, $\bar{\kappa}_1$, and γ. □

5. Variable step sizes and matrices. In the introduction, forward–backward splitting methods were described with variable step sizes λ_k and matrices H_k. We now look at what can be said about such methods on the basis of our contraction results for fixed λ and H. The easier case of variable λ_k with a fixed H has broader significance, so we deal with it first.

Theorem 5.1 (convergence with variable step sizes). *Under assumptions* (A), *consider any step size interval* $[\lambda_-, \lambda_+] \subset (0, \infty)$ *with* λ_+ *small enough that*

$$\lambda_+^{-1} > \frac{\mu_1 - \mu_2}{2} + \frac{\tilde{\kappa}_1}{2} \max\left\{1, \frac{\tilde{\kappa}_1}{\mu_1 + \mu_2}\right\}. \tag{5.1}$$

Let $\theta(\lambda_-, \lambda_+) = \max\{\theta_{\lambda_-}, \theta_{\lambda_+}\}$ *for* θ_λ *defined as in* (2.4). *Then* $\theta(\lambda_-, \lambda_+) < 1$, *and for any sequence of step sizes* $\lambda_k \in [\lambda_-, \lambda_+]$, *all the iteration mappings*

$$S_k = (H + \lambda_k T_2)^{-1}(H - \lambda_k T_1) = (I + \lambda_k H^{-1} T_2)^{-1}(I - \lambda_k H^{-1} T_1) \tag{5.2}$$

are contractions from $D = \operatorname{dom} T$ *into itself with fixed point* $\bar{x}$ *and contraction factor* $\theta(\lambda_-, \lambda_+)$. *In particular, the iterates* $x_k = S_k(x_{k-1})$ *from any starting point* $x_0 \in D$ *converge linearly to* $\bar{x}$ *at a rate no worse than* $\theta(\lambda_-, \lambda_+)$. *Indeed,*

$$\limsup_{k\to\infty} \frac{\|x_k - \bar{x}\|_H}{\|x_{k-1} - \bar{x}\|_H} \le \min\big\{\theta(\lambda_-, \lambda_+), \theta^*(\lambda_-, \lambda_+)\big\}, \tag{5.3}$$

where $\theta^*(\lambda_-, \lambda_+) = \min\{\theta^*_{\lambda_-}, \theta^*_{\lambda_+}\}$ *with* θ^*_λ *denoting the right side of* (4.2).

Proof. The justification of this lies in the proof of Theorem 2.4. It was demonstrated there that θ_λ is an increasing function of λ on the interval of λ values satisfying $\lambda^{-1} < \mu_1$, which includes all λ sufficiently large. On the other hand, it was observed that on the complementary interval, where $\lambda^{-1} \ge \mu_1$, the expression θ_λ^2 is convex as a function of τ under the change of variables induced by taking $\tau^{-1} = \lambda^{-1} + \mu_2$. This implies that θ_λ^2 is unimodal on that interval with respect to λ, and the same then holds for θ_λ. Indeed, we saw for the value $\bar{\lambda}$ defined in (2.6) that θ_λ is a continuous, decreasing function of λ on $\big(0, \bar{\lambda}\big]$ but a continuous, increasing function of λ on $\big[\bar{\lambda}, \infty\big)$.

It follows that the max of θ_λ over any interval $[\lambda_-, \lambda_+] \subset (0, \infty)$ is $\theta(\lambda_-, \lambda_+)$. As long as this value doesn't exceed 1, as guaranteed by (5.1) through Theorem 2.4, we get contraction at the claimed rate $\theta(\lambda_-, \lambda_+)$. An appeal to the ultimate convergence property in Theorem 4.2 then justifies the assertion in (5.3). □

For the case of variable implementation matrices, we won't attempt to prove a result along the lines of a Newton or quasi-Newton method. That would be incompatible with most applications of forward–backward splitting to problem decomposition, where the need to preserve a degree of separability, in order to facilitate computation of the backward steps, is paramount. Also, such applications tend to demand a global statement rather than a local one. For literature on Newton-like results for variational inequalities, see Pang and Chan [16] and Patriksson [17].

Theorem 5.2 (convergence with variable matrices). *Under* (A), *suppose the iterates* $x_k = S_k(x_{k-1})$ *are generated from any* $x_0 \in D$ *by the mappings*

$$S_k = (H_k + \lambda_k T_2)^{-1}(H_k - \lambda_k T_1) = (I + \lambda_k H_k^{-1} T_2)^{-1}(I - \lambda_k H_k^{-1} T_1) \tag{5.4}$$

through a sequence of step sizes $\lambda_k > 0$ *and symmetric, positive definite matrices* H_k *converging to* H. *Let* $\lambda_- = \liminf_k \lambda_k$ *and* $\lambda_+ = \limsup_k \lambda_k$, *and suppose that* $\lambda_- > 0$ *while* λ_+ *satisfies* (5.1). *Then* (5.3) *holds for these values* λ_- *and* λ_+.

Proof. The convergence of H_k to H implies the existence of values $0 < \alpha_k \nearrow 1$ and $0 < \beta_k \nearrow 1$ such that $H - \alpha_k H_k$ and $H_k - \beta_k H$ are positive definite. Through this, the monotonicity of $T_1 - \mu_1 H$ and $T_2 - \mu_2 H$ in condition (2.1) of (A) yields the monotonicity of $T_1 - \mu_{1k} H_k$ and $T_2 - \mu_{2k} H_k$ for the values $\mu_{1k} = \mu_1 \alpha_k \nearrow \mu_1$ and $\mu_{2k} = \mu_2 \alpha_k \nearrow \mu_2$.

We develop now a Lipschitz constant for $\widetilde{T}_{1k} = T_1 - \mu_{1k} H_k = \widetilde{T}_1 + \mu_1(H - \alpha_k H_k)$ from the norm $\|\cdot\|_{H_k}$ to the norm $\|\cdot\|_{H_k^{-1}}$. First, let η_k be such a constant for $H - \alpha_k H_k$, then $\eta_k \to 0$. Next, observe that $\|\cdot\|_{H_k} \geq \sqrt{\beta_k}\,\|\cdot\|_H$, which means that for the corresponding dual norms given by the inverse matrices, $\sqrt{\beta_k}\,\|\cdot\|_{H_k^{-1}} \leq \|\cdot\|_{H^{-1}}$. By these estimates, the Lipschitz inequality in condition (2.2) of (A) gives us

$$\sqrt{\beta_k}\,\|\widetilde{T}_1(x') - \widetilde{T}_1(x)\|_{H_k^{-1}} \leq \tilde{\kappa}_1(1/\sqrt{\beta_k})\|x' - x\|_{H_k} \quad \text{for all } x', x \in D.$$

Hence $\tilde{\kappa}_1/\beta_k$ is a Lipschitz constant for $\widetilde{T}_1$ on D from $\|\cdot\|_{H_k}$ to $\|\cdot\|_{H_k^{-1}}$. Since $\widetilde{T}_{1k} = \widetilde{T}_1 + \mu_1(H - \alpha_k H_k)$, we conclude that the constant $\tilde{\kappa}_{1k} = (\tilde{\kappa}_1/\beta_k) + \mu_1 \eta_k \searrow \tilde{\kappa}_1$ has this property for $\widetilde{T}_{1k}$.

It follows that the splitting $T = T_{1k} + T_{2k}$ with implementation matrix H_k satisfies (A$_k$), the version of (A) in which μ_1, μ_2, and $\tilde{\kappa}_1$ are replaced by μ_{1k}, μ_{2k}, and $\tilde{\kappa}_{1k}$. Now let ϕ stand for the value on the right side of (2.5) and ϕ_k for the corresponding value under this same replacement of constants. Obviously $\phi_k \to \phi$.

Consider any $\epsilon > 0$ small enough that the value $\lambda_-^\epsilon = \lambda_- - \epsilon$ is positive, while the value $\lambda_+^\epsilon = \lambda_+ + \epsilon$ satisfies (5.1); i.e., $(\lambda_+^\epsilon)^{-1} > \phi$. For all k sufficiently large, we have $\lambda_k \in [\lambda_-^\epsilon, \lambda_+^\epsilon]$ and also that $(\lambda_+^\epsilon)^{-1} > \phi_k$. Then by Theorem 2.4 as applied under (A$_k$), the mapping S_k is a contraction from D into itself at the rate θ_{k,λ_k}, where $\theta_{k,\lambda}$ denotes the factor obtained from formula (2.4) with μ_{1k}, μ_{2k}, and $\tilde{\kappa}_{1k}$ substituting for μ_1, μ_2, and $\tilde{\kappa}_1$. Furthermore, we have

$$\theta_{k,\lambda} \leq \theta_k(\lambda_-^\epsilon, \lambda_+^\epsilon) = \max\{\theta_{k,\lambda_-^\epsilon}, \theta_{k,\lambda_+^\epsilon}\}$$

for the reasons in the proof of Theorem 5.1 (when applied to $\theta_{k,\lambda}$ as a function of λ). Therefore, the lim sup in (5.3) is bounded above by the limit of $\theta_k(\lambda_-^\epsilon, \lambda_+^\epsilon)$ as $k \to \infty$, which is $\theta(\lambda_-^\epsilon, \lambda_+^\epsilon)$. This being valid for all $\epsilon > 0$ sufficiently small, we can take the limit as $\epsilon \searrow 0$ and obtain the inequality in (5.3), as targeted. □

6. Asymmetric implementations.

Only symmetric implementation matrices H_k are covered directly by our results up to this stage, but what about the possibility of more general matrices that are not symmetric, although still positive definite? Such modes of implementation crop up, for example, in applications to variational inequality when H_k is taken to be an approximation to the Jacobian matrix $\nabla F(x_k)$ or some part of it. Aside from the gradient case where $F = \nabla f$ and $\nabla F(x_k) = \nabla^2 f(x_k)$, H_k may then lack symmetry.

Asymmetric implementation matrices can be incorporated into our theory by a simple device. This device has already been used by others, e.g., Tseng in [25], but we go beyond previous instances because of the attention we pay to step sizes. To explain the idea, we keep to the case of constant H for simplicity. Also, to avoid conflicts with our earlier statements, we follow the notational strategy of replacing H by $H + K$ with K antisymmetric ($K^\top = -K$) and H still symmetric, rather than taking H itself to lack symmetry. This conforms to the fact that any positive definite matrix can be written as the sum of an antisymmetric matrix and a symmetric, positive definite matrix.

In this mode, the iteration mappings for the forward–backward method with respect to a splitting $T = T_1 + T_2$ take the form

$$\big([H+K] + \lambda T_2\big)^{-1}\big([H+K] - \lambda T_1\big). \tag{6.1}$$

Their practicality hinges on the ease of calculating images under the inverse mapping $\big([H+K]+\lambda T_2\big)^{-1}$. This has to be assumed for any analysis to be worthwhile, and it's true that in applications such have been pinpointed by Pang and Chan [16] and Tseng [25].

For our purposes we'll make such practicality of backward step execution part of the framework by assuming that for any $\tau \in (-\infty, \infty)$, the inverse $\big([H+\tau K]+\lambda T_2\big)^{-1}$ can be handled just as readily as $\big([H+K]+\lambda T_2\big)^{-1}$. We put our focus therefore on *two-parameter* iteration mappings, namely

$$S_{\lambda,\tau} = \big([H+\tau K] + \lambda T_2\big)^{-1}\big([H+\tau K] - \lambda T_1\big). \tag{6.2}$$

These mappings, like the earlier ones where K didn't appear, all have the unique solution $\bar{x}$ as their unique fixed point. We explore the relation between contraction properties of $S_{\lambda,\tau}$ and the values of both λ and τ.

THEOREM 6.1 (reduction of asymmetric to symmetric implementations). *Assume* (A) *as before, except for what it says about* $\tilde{\kappa}_1$; *in place of that, consider a Lipschitz constant* $\tilde{\kappa}_1(\sigma)$ *for the mapping* $\widetilde{T}_1 - \sigma K$ *on* D, *with* σ *being any value in* $(-\infty,\infty)$. *Let*

$$\lambda(\sigma) = \frac{1}{\left(\dfrac{\tilde{\kappa}_1(\sigma)^2}{\mu_1+\mu_2}\right) + \mu_1}, \qquad \tau(\sigma) = \sigma\lambda(\sigma). \tag{6.3}$$

Then the asymmetrically implemented iteration mapping

$$S_{\lambda(\sigma),\tau(\sigma)} \;=\; \big([H+\tau(\sigma)K] + \lambda(\sigma)T_2\big)^{-1}\big([H+\tau(\sigma)K] - \lambda(\sigma)T_1\big), \tag{6.4}$$

with respect to the splitting $T = T_1 + T_2$, *is identical to the symmetrically implemented iteration mapping*

$$S'_{\lambda(\sigma)} \;=\; \big(H + \lambda(\sigma)T'_2\big)^{-1}\big(H - \lambda(\sigma)T'_1\big), \tag{6.5}$$

with respect to the splitting $T = T'_1 + T'_2$, *where* $T'_1 = T_1 - \sigma K$ *and* $T'_2 = T_2 + \sigma K$, *and it is Lipschitz continuous on* D *with constant*

$$\theta(\sigma) = \frac{1}{\sqrt{1 + \left(\dfrac{\mu_1+\mu_2}{\tilde{\kappa}_1(\sigma)}\right)^2}} \;<\; 1. \tag{6.6}$$

Proof. Iterations $x_k = S_{\lambda,\tau}(x_{k-1})$ have the meaning that

$$0 \in \frac{1}{\lambda}[H+\tau K][x_k - x_{k-1}] + T_1(x_{k-1}) + T_2(x_k).$$

This condition can equally well be written as

$$0 \in \frac{1}{\lambda}H[x_k - x_{k-1}] + [T_1 - \sigma K](x_{k-1}) + [T_2 + \sigma K](x_k) \tag{6.7}$$

under the correspondence $\sigma = \tau/\lambda$, $\tau = \sigma\lambda$. Thus, the same iterations can be written as $x_k = S'_\lambda(x_{k-1})$ for $S'_\lambda = (H + \lambda T'_2)^{-1}(H - \lambda T'_1)$. The splitting $T = T'_1 + T'_2$ satisfies (A) with Lipschitz constant $\tilde{\kappa}_1(\sigma)$, so Theorem 2.4 applies. The optimal step size coming out of that result is $\lambda(\sigma)$ as given by (6.3), and it yields for $S'_{\lambda(\sigma)}$ the contraction rate $\theta(\sigma)$ defined in (6.6). □

The observation to be made from Theorem 6.1 is that instead of pursuing asymmetric implementations directly, a good strategy is to first subtract off from T_1 to get T'_1 whatever multiple σ of the asymmetric part K of the implementation matrix $H+K$ is appropriate in order to reduce the Lipschitz constant $\tilde{\kappa}_1(\sigma)$ as far as possible. This multiple is added to T_2 to get T'_2. Thereafter, it's just a matter of taking the optimal step size $\lambda(\sigma)$ for the altered splitting $T = T'_1 + T'_2$ with respect to the symmetric part H of the implementation matrix, in accordance with the earlier results. The net effect will be the same as the asymmetric iterations (6.4) but executed symmetrically and at an optimized rate.

REFERENCES

[1] P.-L. Lions and B. Mercier, *Splitting algorithms for the sum of two nonlinear operators*, SIAM J. Numer. Anal., 16 (1979), pp. 964–979.

[2] A. A. Goldstein, *Convex programming in Hilbert space*, Bull. Amer. Math. Soc., 70 (1964), pp. 709–710.

[3] E. S. Levitin and B. T. Polyak, *Constrained minimization methods*, U.S.S.R. Comput. Math. and Math. Phys., 6 (1966), pp. 1–50.

[4] H. Brézis and M. Sibony, *Méthodes d'approximation et d'itération pour les opérateurs monotones*, Arch. Rational Mech. Anal., 27 (1969), pp. 59–82.

[5] M. Sibony, *Methodes iteratives pour les equations et inequations aux derivées partielles non-linéaires de type monotone*, Calcolo, 7 (1970), pp. 65–183.

[6] H. Gajewski and R. Kluge, *Projektionsverfahren für nichtlinearen variationsungleichungen*, Math. Nachr., 46 (1970), pp. 363–373.

[7] S. Dafermos, *Traffic equilibrium and variational inequalities*, Transportation Sci., 14 (1980), pp. 42–54.

[8] R. T. Rockafellar, *Monotone mappings and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.

[9] R. T. Rockafellar, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Math. Oper. Res., 1 (1976), pp. 97–116.

[10] J. Eckstein and D. P. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone mappings*, Math. Programming, 55 (1992), pp. 293–318.

[11] J. E. Spingarn, *Partial inverse of a monotone mapping*, Appl. Math. Optim., 10 (1983), pp. 247–265.

[12] J. Eckstein, *Splitting Methods for Monotone Operators with Applications to Parallel Optimization*, Ph.D. thesis, CICS-TH-140, MIT, Cambridge, MA, 1989.

[13] G. Cohen, *Optimization by decomposition and coordination: A unified approach*, IEEE Trans. Automat. Control, AC-23 (1978), pp. 222–232.

[14] G. Cohen, *Auxiliary problem principle and decomposition of optimization problems*, J. Optim. Theory Appl., 32 (1980), pp. 277–305.

[15] G. Cohen, *Auxiliary problem principle extended to variational inequalities*, J. Optim. Theory Appl., 59 (1988), pp. 369–390.

[16] J. S. Pang and D. Chan, *Iterative methods for variational and complementarity problems*, Math. Programming, 24 (1982), pp. 284–313.

[17] M. Patriksson, *A Unified Framework of Descent Algorithms for Nonlinear Programs and Variational Inequalities*, Doctoral thesis, Linköping University, 1993.

[18] R. T. Rockafellar, *Linear-quadratic programming and optimal control*, SIAM J. Control Optim., 25 (1987), pp. 781–814.

[19] R. T. Rockafellar, *Multistage convex programming and discrete-time optimal control*, Control Cybernet., 17 (1988), pp. 225–246.

[20] R. T. Rockafellar and R. J-B Wets, *Generalized linear-quadratic problems of deterministic and stochastic optimal control in discrete time*, SIAM J. Control Optim., 28 (1990), pp.

810–822.

[21] R. Temam and I. Ekeland, *Analyse Convex et Problèmes Variationnels*, Dunod Gauthier-Villars, Paris, 1974.

[22] G. B. Passty, *Ergodic convergence to a zero of the sum of monotone mappings in Hilbert space*, J. Math. Anal. Appl., 72 (1979), pp. 383–390.

[23] D. Gabay, *Applications of the method of multipliers to variational inequalities*, in Augmented Lagrangian Methods: Applications to the Solution of Boundary Value Problems, M. Fortin and R. Glowinski, eds., North–Holland, Amsterdam, 1983.

[24] P. Tseng, *Applications of a splitting algorithm to decomposition in convex programming and variational inequalities*, SIAM J. Control Optim., 29 (1991), pp. 119–138.

[25] P. Tseng, *Further applications of a splitting algorithm to decomposition in variational inequalities and convex programming*, Math. Programming, 48 (1990), pp. 249–263.

[26] K. Mouallif, V. H. Nguyen, and J.-J. Strodiot, *A perturbed parallel decomposition method for a class of nonsmooth convex minimization problems*, SIAM J. Control Optim., 29 (1991), pp. 829–847.

[27] S. Makler–Scheimberg, V. H. Nguyen, and J.-J. Strodiot, *A Family of Perturbed Parallel Decomposition Methods for Variational Inequalities*, 1993, preprint.

[28] C. Dupuis and J.-M. Darveau, *The convergence conditions of diagonalization and projection methods for fixed demand asymmetric network equilibrium problems*, Oper. Res. Lett., 5 (1986), pp. 149–155.

[29] D. P. Bertsekas and E. M. Gafni, *Projection methods for variational inequalities with application to the traffic assignment problem*, Math. Programming Study, 17 (1982), pp. 139–159.

[30] L. Zanni, *On the convergence rate of two projection methods for variational inequalities in R^n*, Calcolo, 29 (1992), pp. 193–212.

[31] L. Zanni, *Convergence properties of a projection method for affine variational inequality problems*, Matematiche, 49 (1994), pp. 359–377.

[32] A. Renaud, *Algorithmes de Régularisation et Décomposition pour les Problèmes Variationnels Monotones*, Doctoral thesis, L'École Nationale Supérieure des Mines de Paris, Paris, France, 1993.

[33] P. Marcotte and J. H. Wu, *On the convergence of projection methods: Application to the decomposition of affine variational inequalities*, J. Optim. Theory Appl., 85 (1995), pp. 347–362.

[34] Z. Q. Luo and P. Tseng, *Error bound and convergence analysis of matrix splitting algorithms for the affine variational inequality problem*, SIAM J. Optim., 2 (1992), pp. 43–54.

[35] P. Tseng, *On the linear convergence of iterative methods for the variational inequality problem*, J. Comput. Appl. Math., 60 (1995), pp. 237–252.

[36] P. T. Harker and J.-S. Pang, *Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications*, Math. Programming, 48 (1990), pp. 161–220.

[37] G. H-G. Chen, *Forward-Backward Splitting Techniques: Theory and Applications*, Doctoral thesis, University of Washington, Seattle, WA, 1994.

[38] S. Dafermos, *An iterative scheme for variational inequalities*, Math. Programming, 26 (1983), pp. 40–47.

[39] G. J. Minty, *On the maximal domain of a 'monotone' function*, Michigan Math. J., 8 (1961), pp. 135–137.

[40] R. T. Rockafellar, *On the maximality of sums of nonlinear monotone mappings*, Trans. Amer. Math. Soc., 149 (1970), pp. 75–88.

[41] R. T. Rockafellar, *Local boundedness of nonlinear monotone mappings*, Michigan Math. J., 16 (1969), pp. 397–407.

[42] P. Marcotte and F. Guélat, *Adaptation of a modified Newton method for solving the asymmetric traffic equilibrium problem*, Transportation Sci., 22 (1988), pp. 112–124.

[43] B. T. Polyak, *Introduction to Optimization*, Optimization Software, New York, 1987.

SIAM J. OPTIM.
Vol. 7, No. 2, pp. 445–462, May 1997

010

CONVERGENCE OF NEWTON'S METHOD FOR SINGULAR SMOOTH AND NONSMOOTH EQUATIONS USING ADAPTIVE OUTER INVERSES[*]

XIAOJUN CHEN[†], ZUHAIR NASHED[‡], AND LIQUN QI[†]

Abstract. We present a local convergence analysis of generalized Newton methods for singular smooth and nonsmooth operator equations using adaptive constructs of outer inverses. We prove that for a solution x^* of $F(x) = 0$, there exists a ball $S = S(x^*, r)$, $r > 0$ such that for any starting point $x_0 \in S$ the method converges to a solution $\bar{x}^* \in S$ of $\Gamma F(x) = 0$, where Γ is a bounded linear operator that depends on the Fréchet derivative of F at x_0 or on a generalized Jacobian of F at x_0. Point $\bar{x}^*$ may be different from x^* when x^* is not an isolated solution. Moreover, we prove that the convergence is quadratic if the operator is smooth and superlinear if the operator is locally Lipschitz. These results are sharp in the sense that they reduce in the case of an invertible derivative or generalized derivative to earlier theorems with no additional assumptions. The results are illustrated by a system of smooth equations and a system of nonsmooth equations, each of which is equivalent to a nonlinear complementarity problem.

Key words. Newton's method, convergence theory, nonsmooth analysis, outer inverses, nonlinear complementarity problems

AMS subject classifications. 65J15, 65H10, 65K10, 49M15

PII. S1052623493246288

1. Introduction.

Let X and Y be Banach spaces and let $L(X, Y)$ denote the set of all bounded linear operators on X into Y. Let $F : X \to Y$ be a continuous function. We consider the nonlinear operator equation

$$F(x) = 0, \qquad x \in X. \tag{1.1}$$

When $X = Y$, it is well known that if F is Fréchet differentiable and F' is locally Lipschitz and invertible at a solution x^*, then there exists a ball $S(x^*, r), r > 0$ such that for any $x_0 \in S(x^*, r)$, the Newton method

$$x_{k+1} = x_k - F'(x_k)^{-1} F(x_k) \tag{1.2}$$

is quadratically convergent to x^*. See, e.g., [9, 27, 34].

In the nonsmooth case, $F'(x_k)$ may not exist. The generalized Newton method proposes to use generalized Jacobians of F to play the role of F' in the Newton method (1.2) in the finite dimensional case. Let F be a locally Lipschitzian mapping from R^n into R^m. Then Rademacher's theorem implies that F is almost everywhere differentiable. Let D_F be the set where F is differentiable. Denote

$$\partial_B F(x) = \{ \lim_{\substack{x_i \to x \\ x_i \in D_F}} \nabla F(x_i) \}.$$

[*] Received by the editors March 26, 1993; accepted for publication December 15, 1995. This work was supported by the Australian Research Council and National Science Foundation grant DMS-901526.

http://www.siam.org/journals/siopt/7-2/24628.html

[†] School of Mathematics, University of New South Wales, Sydney 2052, Australia (x.chen@unsw.edu.au, l.qi@unsw.edu.au).

[‡] Department of Mathematical Sciences, University of Delaware, Newark, DE 19716 (nashed@math.udel.edu).

The generalized Jacobian of F at $x \in R^n$ in the sense of Clarke [8] is equal to the convex hull of $\partial_B F(x)$,

$$\partial F(x) = \mathrm{conv}\partial_B F(x),$$

which is a nonempty convex compact set. The Newton method for nonsingular nonsmooth equations using the generalized Jacobian is defined by

$$x_{k+1} = x_k - V_k^{-1} F(x_k), \qquad V_k \in \partial F(x_k). \tag{1.3}$$

A local superlinear convergence theorem is given in [33], where it is assumed that all $V \in \partial F(x^*)$ are nonsingular.

Qi [31] suggested a modified version of method (1.3) in the form

$$x_{k+1} = x_k - V_k^{-1} F(x_k), \qquad V_k \in \partial_B F(x_k) \tag{1.4}$$

and gave a local superlinear convergence theorem for method (1.4). His theorem reduced the nonsingularity requirement on all members of $\partial F(x^*)$ to all members of $\partial_B F(x^*)$.

Another modification is an iteration function method introduced by Han, Pang, and Rangaraj [13] using an iteration function $G(\cdot;\cdot) : R^n \times R^n \to R^n$. If F has a one-sided directional derivative

$$F'(x; d) := \lim_{t \downarrow 0} \frac{F(x + td) - F(x)}{t} \tag{1.5}$$

and $G(x; d) = F'(x; d)$, a variant of the iteration function method can be defined by

$$\begin{cases} \text{solve} \quad F(x_k) + F'(x_k; d) = 0, \\ \text{set} \qquad x_{k+1} = x_k + d. \end{cases} \tag{1.6}$$

See also Pang [28] and Qi [31].

Methods (1.2), (1.3), (1.4), and (1.6) are very useful, but they are not applicable to the singular case. At each step in (1.2), (1.3), and (1.4), the inverse of a Jacobian or a generalized Jacobian is required; in (1.6) a nonlinear equation is solved at each step (in the singular case, it may have no solutions). Often, the inverse cannot be guaranteed to exist; singularity occurs in many applications. For example, we consider the nonlinear complementarity problem (NCP): for a given $f : R^n \to R^n$, find $x \in R^n$ such that

$$x \geq 0, \quad f(x) \geq 0, \quad \text{and} \quad x^T f(x) = 0.$$

Mangasarian [19] formulated the NCP in the case when f is Fréchet differentiable as an equivalent system of smooth equations:

$$\hat{F}_i(x) = (f_i(x) - x_i)^2 - f_i(x)|f_i(x)| - x_i|x_i| = 0, \qquad i = 1, 2, \ldots, n, \tag{1.7}$$

where $f = (f_1, \ldots, f_n)^T$. Let

$$\mathrm{sgn}(\alpha) = \begin{cases} 1, & \alpha \geq 0, \\ -1, & \alpha < 0, \end{cases}$$

and let δ_{ij} denote the Kronecker function. It is easy to show that the Jacobian of $\hat{F} := (\hat{F}_1, \dots, \hat{F}_n)^T$ at x is given by

$$\frac{\partial \hat{F}_i(x)}{\partial x_j} = 2f_i(x)\frac{\partial f_i(x)}{\partial x_j}(1 - \operatorname{sgn}(f_i(x))) + 2x_i\delta_{ij}(1 - \operatorname{sgn}(x_i)) - 2f_i(x)\delta_{ij} - 2x_i\frac{\partial f_i(x)}{\partial x_j},$$

$$i, j = 1, 2, \dots, n.$$

The Jacobian $\nabla \hat{F}(x)$ is singular when there is some degeneracy; i.e., $x_i = f_i(x) = 0$ for some i.

The NCP can also be formulated as a system of nonsmooth equations [28]:

$$\tilde{F}(x) = \min(f(x), x) = 0, \tag{1.8}$$

where the "min" operator denotes the componentwise minimum of two vectors. It is hard to guarantee that all members of $\partial_B F(x)$ are nonsingular when there is some nonsmoothness; i.e., $x_i = f_i(x)$ and $e_i \neq \nabla f_i(x)$ for some i, where e_i is the ith row of the identity matrix $I \in R^{n \times n}$.

In [5], Chen and Qi studied a parameterized Newton method:

$$x_{k+1} = x_k - (V_k + \lambda_k I)^{-1} F(x_k), \qquad V_x \in \partial_B F(x_k),$$

where λ_k is a parameter to ensure the existence of the inverse of $V_k + \lambda_k I$. The local superlinear convergence theorem in [5] requires all $V_* \in \partial_B F(x^*)$ to be nonsingular.

In Newton-like methods for solving smooth and nonsmooth equations, e.g., quasi-Newton methods and splitting methods, the Jacobian is often required to be nonsingular at a solution x^* to which the method is supposed to converge [4, 5, 6, 9, 15, 16, 27, 28, 32, 40]. Hence, it is interesting to know what happens with the Newton methods when $F'(x^*)$ or some $V_* \in \partial_B F(x^*)$ are singular at x^*. In this case the solution set is locally a manifold of positive dimension; hence, x^* is not an isolated solution.

Let $A \in L(X, Y)$. We denote the range and nullspace of A by $R(A)$ and $N(A)$, respectively. A linear operator $A^\sharp : Y \to X$ is said to be an outer inverse of A if $A^\sharp A A^\sharp = A^\sharp$.

In this paper, for $X = R^n$ and $Y = R^m$, we consider a generalized Newton method

$$x_{k+1} = x_k - V_k^\sharp F(x_k), \tag{1.9}$$

where $V_k \in \partial_B F(x_k)$ and $V_k^\sharp$ denotes an outer inverse of V_k.

Newton's method for singular smooth equations using outer inverses

$$x_{k+1} = x_k - F'(x_k)^\sharp F(x_k) \tag{1.10}$$

has been considered by Ben-Israel [1], Deuflhard and Heindl [10], and Nashed [25] and more recently by Nashed and Chen [26]. Reference [26] presented a Kantorovich-type theorem (semilocal convergence) for Newton-like methods for singular smooth equations using outer inverses: if some conditions hold at the starting point x_0, method (1.10) converges to a solution of $F'(x_0)^\sharp F(x) = 0$.

This paper establishes new results on Newton's method for smooth and nonsmooth equations. In particular, we consider the behavior of methods (1.9) and (1.10) when the singularity occurs at a solution x^*, which is close to the starting point.

In section 2 we state the definitions and properties of generalized gradients, semismooth functions, and outer inverses. These results are used to analyze convergence of methods (1.9) and (1.10).

In section 3, by using a Kantorovich-type theorem, we give a locally quadratic convergence theorem for Newton's method (1.10) in the following sense: for a solution x^* of (1.1), there is a ball $S(x^*, r)$ with $r > 0$ such that for any $x_0 \in S(x^*, r)$, Newton's method (1.10) with $F'(x_k)^\sharp = (I + F'(x_0)^\sharp(F'(x_k) - F'(x_0)))^{-1}F'(x_0)^\sharp$ converges quadratically to a solution $\bar{x}^*$ of $F'(x_0)^\sharp F(x) = 0$. Here, $\bar{x}^*$ may be different from x^*, because of singularity; there is no guarantee for uniqueness of the solutions. This is a major difference between singular and nonsingular equations.

In section 4, by using a Mysovskii-type theorem, we prove the superlinear convergence of method (1.9) for nonsmooth equations. Difficulties in the analysis of method (1.9) for singular nonsmooth equations that have not been previously resolved in the literature arise from the fact that there are some singular elements $V_x \in \partial_B F(x)$, so rank(V_x) are different and $V_x^\sharp V_x \neq I$. Previous results for nonsingular equations require that all $V_x \in \partial_B F(x)$ have full rank and $V_x V_x^{-1} = V_x^{-1} V_x = I$. We develop new techniques for considering singular nonsmooth equations. It is noteworthy that the solution to which our method converges (in both smooth and nonsmooth cases) need not be the original solution, and our method applies even when the solution set is locally a manifold of positive dimension.

In section 5 we illustrate the singularity issue by numerical examples. We give numerical results for computing three examples of the NCP by methods (1.9) and (1.10) via a system of smooth equations (1.7) and a system of nonsmooth equations (1.8), respectively.

2. Definitions and lemmas on outer inverses and semismooth functions.

Outer inverses of linear operators play a pivotal role in the formulation and convergence analysis of the iterative methods studied in this paper. Their role is derived from projectional properties of outer inverses and, more importantly, from perturbation and stability analysis of outer inverses. The strategy is based on Banach-type lemmas and perturbation bounds for outer inverses which show that the set of outer inverses (to a given bounded linear operator) admits selections that behave like bounded linear inverses, in contrast to inner inverses or generalized inverses, which do not depend continuously on perturbations of the operators. This strategy was first used in [26] to generate adaptive constructs of outer inverses that would lead to sharp convergence results. Lemmas 2.1–2.4 below summarize important perturbation bounds and projectional properties of outer inverses which are used in the convergence analysis. For detailed proofs and related properties and references, see [26]. For definition and properties of generalized inverses in Banach spaces, see [24] or [25].

Lemma 2.1 (Banach-type lemma for outer inverses). *Let $A \in L(X, Y)$ and let $A^\sharp \in L(Y, X)$ be an outer inverse of A. Let $B \in L(X, Y)$ be such that $||A^\sharp(B - A)|| < 1$. Then $B^\sharp := (I + A^\sharp(B - A))^{-1}A^\sharp$ is a bounded outer inverse of B with $N(B^\sharp) = N(A^\sharp)$ and $R(B^\sharp) = R(A^\sharp)$. Moreover,*

$$||B^\sharp - A^\sharp|| \leq \frac{||A^\sharp(B - A)||||A^\sharp||}{1 - ||A^\sharp(B - A)||}$$

and

$$||B^\sharp A|| \leq \frac{1}{1 - ||A^\sharp(B - A)||}.$$

LEMMA 2.2. *Let $A \in L(X,Y)$. If $A^\sharp$ is a bounded outer inverse of A, then the following topological direct sum decompositions hold:*

$$X = R(A^\sharp) \oplus N(A^\sharp A),$$

$$Y = N(A^\sharp) \oplus R(AA^\sharp).$$

LEMMA 2.3. *Let $A, B \in L(X,Y)$ and let $A^\sharp$ and $B^\sharp \in L(Y,X)$ be outer inverses of A and B, respectively. Then $B^\sharp(I - AA^\sharp) = 0$ if and only if $N(A^\sharp) \subset N(B^\sharp)$.*

LEMMA 2.4. *Let $A \in L(X,Y)$ and let $A^\dagger$ be a bounded generalized inverse of A. Let $B \in L(X,Y)$ satisfy the condition $||A^\dagger(B - A)|| < 1$, and define $B^\sharp := (I + A^\dagger(B - A))^{-1}A^\dagger$. Then $B^\sharp$ is a generalized inverse of B if and only if*

$$\mathrm{dim}N(B) = \mathrm{dim}N(A)$$

and

$$\mathrm{codim}R(B) = \mathrm{codim}R(A).$$

Note that if A and B are Fredholm operators with the same index, the two dimensionality conditions are equivalent. Thus Lemma 2.4 and Theorem 3.4 below are not restricted to finite dimensional spaces. A simple example of a Fredholm operator is an operator of the form $I + K$, where K is a compact operator.

For nonsmooth problems, we consider functions which are semismooth. We now recall two definitions and an important property related to this class of functions.

DEFINITION 2.5. *A function $F : R^n \to R^m$ is said to be B-differentiable at a point x if F has a one-sided directional derivative $F'(x;h)$ at x (see (1.5)) and*

$$\lim_{h\to 0} \frac{F(x+h) - F(x) - F'(x;h)}{\| h \|} = 0. \tag{2.1}$$

We may write (2.1) as $F(x+h) = F(x) + F'(x;h) + o(\| h \|)$.

DEFINITION 2.6. *A function $F : R^n \to R^m$ is semismooth at x if F is locally Lipschitz at x and*

$$\lim_{\substack{V\in\partial F(x+th') \\ h'\to h, t\downarrow 0}} \{Vh'\}$$

exists for every $h \in R^n$.

LEMMA 2.7 (see [31]). *If $F : R^n \to R^m$ is semismooth at x, then F is directionally differentiable at x, and for any $V \in \partial F(x+h)$,*

$$Vh - F'(x;h) = o(||h||).$$

Shapiro [36] showed that a locally Lipschitzian function F is B-differentiable at x if and only if it is directionally differentiable. Hence, F is B-differentiable at x if F is semismooth at x. For a comprehensive analysis of the role of semismooth functions, see [21, 31, 33].

3. Local convergence for smooth equations. $S(x,r)$ denotes the open ball in X with center x and radius r, and $\bar{S}(x,r)$ is its closure. For a fixed $A \in L(X,Y)$, we denote the set of nonzero outer inverses of A by

$$\Omega(A) := \{B \in L(Y,X) : BAB = B, B \neq 0\}.$$

In this section we give two local convergence theorems for method (1.10) by using the following Kantorovich-type theorem (semilocal convergence).

THEOREM 3.1. *Let $F : D \subset X \to Y$ be Fréchet differentiable. Assume that there exist an $x_0 \in D$, $F'(x_0)^\sharp \in \Omega(F'(x_0))$ and constants $\eta, K > 0$ such that for all $x, y \in D$ the following conditions hold:*

$$||F'(x_0)^\sharp F(x_0)|| \leq \eta, \tag{3.1}$$

$$||F'(x_0)^\sharp (F'(x) - F'(y))|| \leq K||x-y||, \tag{3.2}$$

$$h := K\eta \leq \frac{1}{2}, \quad S(x_0, t^*) \subset D, \tag{3.3}$$

where $t^* = (1 - \sqrt{1-2h})/K$. Then, the sequence $\{x_k\}$ defined by method (1.10) *with $F'(x_k)^\sharp = (I + F'(x_0)^\sharp(F'(x_k) - F'(x_0)))^{-1} F'(x_0)^\sharp$ lies in $S(x_0, t^*)$ and converges to a solution x^* of $F'(x_0)^\sharp F(x) = 0$.*

(See Theorem 3.1 and Corollary 3.1 in [26].)

THEOREM 3.2. *Let $F : D \subset X \to Y$ be Fréchet differentiable and assume that $F'(x)$ satisfies a Lipschitz condition*

$$||F'(x) - F'(y)|| \leq L||x-y||, \qquad x \in D. \tag{3.4}$$

Assume that there exists an $x^* \in D$ such that $F(x^*) = 0$. Let $p > 0$ be a positive number such that $S(x^*, \frac{1}{p}) \subset D$. Suppose that the following condition holds:

(a) *There is an $F'(x^*)^\sharp \in \Omega(F'(x^*))$ such that $||F'(x^*)^\sharp|| \leq p$, and for any $x \in S(x^*, \frac{1}{3Lp})$, the set $\Omega(F'(x))$ contains an element of minimal norm.*

Then there exists a ball $S(x^*, r) \subset D$ with $0 < r < \frac{1}{3Lp}$ such that for any $x_0 \in S(x^*, r)$, the sequence $\{x_k\}$ defined by method (1.10) *with*

$$F'(x_0)^\sharp \in \textit{argmin}\{||B|| : B \in \Omega(F'(x_0))\} \tag{3.5}$$

and with $F'(x_k)^\sharp = (I + F'(x_0)^\sharp(F'(x_k) - F'(x_0)))^{-1} F'(x_0)^\sharp$ converges quadratically to $\bar{x}^* \in S(x_0, \frac{1}{Lp}) \cap \{R(F'(x_0)^\sharp) + x_0\}$, which is a solution of $F'(x_0)^\sharp F(x) = 0$. Here, $R(F'(x_0)^\sharp) + x_0 := \{x + x_0 : x \in R(F'(x_0)^\sharp)\}$.

Proof. Let

$$\bar{r} = \frac{1}{3Lp} \quad \text{and} \quad \epsilon = \frac{4}{27Lp^2}.$$

We first prove that there exists a ball $S(x^*, r) \subset D, 0 < r \leq \bar{r}$ such that for any $x_0 \in S(x^*, r)$, all conditions of Theorem 3.1 hold.

Since F is continuous at x^*, there exists a ball $S(x^*, r) \subset D, 0 < r \leq \bar{r}$, such that for any $x \in S(x^*, r)$,

$$||F(x)|| < \epsilon.$$

From (3.4), there is an $F'(x^*)^\sharp \in \Omega(F'(x^*))$ such that

$$||F'(x^*)^\sharp(F'(x) - F'(x^*))|| \le pLr < 1.$$

By Lemma 2.1, we have that

$$F'(x)^\sharp = (I + F'(x^*)^\sharp(F'(x) - F'(x^*)))^{-1}F'(x^*)^\sharp$$

is an outer inverse of $F'(x)$ and

$$||F'(x)^\sharp|| \le \frac{||F'(x^*)^\sharp||}{1 - pLr} \le \frac{p}{1 - pLr} =: \beta.$$

Hence, for any $x_0 \in S(x^*, r)$, the outer inverse

$$F'(x_0)^\sharp \in \operatorname{argmin}\{||B|| : B \in \Omega(F'(x_0))\}$$

satisfies $||F'(x_0)^\sharp|| \le \beta$. Let $K = \frac{3}{2}\beta L$. Then, for $x, y \in D$,

$$||F'(x_0)^\sharp(F'(x) - F'(y))|| \le \beta||F'(x) - F'(y)|| \le K||x - y||,$$

and

$$h = K||F'(x_0)^\sharp F(x_0)|| \le \frac{3}{2}L\beta^2\epsilon \le \frac{2}{9(1 - pLr)^2} \le \frac{1}{2}.$$

Furthermore, for any $x \in S(x_0, t^*)$ with $t^* = (1 - \sqrt{1 - 2h})/K$, we have

$$||x^* - x|| \le ||x_0 - x^*|| + ||x_0 - x|| \le \frac{1}{3Lp} + \frac{2}{3L\beta} \le \frac{1}{3Lp} + \frac{2(1 - Lpr)}{3Lp} \le \frac{1}{Lp}.$$

This implies $S(x_0, t^*) \subset S(x^*, \frac{1}{Lp}) \subset D$. Hence, all conditions of Theorem 3.1 hold at x_0. Thus, the sequence $\{x_k\}$ lies in $S(x_0, t^*)$ and converges to a solution $\bar{x}^*$ of $F'(x_0)^\sharp F(x) = 0$.

Now we prove that the convergence rate is quadratic.

Since $F'(x_k)^\sharp = (I + F'(x_0)^\sharp(F'(x_k) - F'(x_0)))^{-1}F'(x_0)^\sharp$ by Lemma 2.1, $R(F'(x_0)^\sharp) = R(F'(x_k)^\sharp)$. By

$$x_{k+1} - x_k = F'(x_k)^\sharp F(x_k) \in R(F'(x_k)^\sharp),$$

we have

$$x_{k+1} \in R(F'(x_k)^\sharp) + x_k = R(F'(x_{k-1})^\sharp) + x_k = R(F'(x_0)^\sharp) + x_0$$

and $\bar{x}^* \in R(F'(x_k)^\sharp) + x_{k+1}$ for any $k \ge 0$. This implies that

$$\bar{x}^* \in R(F'(x_0)^\sharp) + x_0 = R(F'(x_k)^\sharp) + x_0$$

and

$$\begin{aligned}&F'(x_k)^\sharp F'(x_k)(\bar{x}^* - x_{k+1})\\ &= F'(x_k)^\sharp F'(x_k)(\bar{x}^* - x_0) - F'(x_k)^\sharp F'(x_k)(x_{k+1} - x_0) = \bar{x}^* - x_{k+1}.\end{aligned}$$

From Lemma 2.3, we have $F'(x_k)^\sharp = F'(x_k)^\sharp F'(x_0)F'(x_0)^\sharp$. Using $F'(x_0)^\sharp F(\bar{x}^*) = 0$ and $N(F'(x_0)^\sharp) = N(F'(x_k)^\sharp)$, we obtain $F'(x_k)^\sharp F(\bar{x}^*) = 0$ and

$$\begin{aligned}
&||\bar{x}^* - x_{k+1}|| = ||F'(x_k)^\sharp F'(x_k)(\bar{x}^* - x_{k+1})|| \\
&= ||F'(x_k)^\sharp F'(x_k)(\bar{x}^* - x_k + F'(x_k)^\sharp (F(x_k) - F(\bar{x}^*)))|| \\
&= ||F'(x_k)^\sharp (F'(x_k)(\bar{x}^* - x_k) - \int_0^1 F'(x_k + t(x_k - \bar{x}^*))dt(\bar{x}^* - x_k))|| \\
&= ||F'(x_k)^\sharp F'(x_0)||||F'(x_0)^\sharp (F'(x_k) - \int_0^1 F'(x_k + t(x_k - \bar{x}^*))dt)(\bar{x}^* - x_k)|| \\
&\le \frac{1}{1 - Kt^*} \cdot \frac{K}{2} ||\bar{x}^* - x_k||^2.
\end{aligned}$$

Hence, $x_k \to \bar{x}^*$ quadratically. □

Lemma 3.3. *Let $A \in L(X, Y), A \neq 0$, where X and Y are finite dimensional normed spaces. Then the infimum of $||B||$ over $\Omega(A)$ is attained.*

Proof. Let A be a fixed nonzero linear operator from X into Y. For any nonzero outer inverse B of A, we have $||B|| = ||BAB|| \le ||B||^2||A||$; hence, $||B|| \ge \frac{1}{||A||}$. Let α :=inf $\{||B|| : B \in \Omega(A)\}$. There exists $\{B_k\} \subset \Omega(A)$ such that lim $\|B_k\| = \alpha$; since $\{B_k\}$ is bounded, it has a limit point B. Then $B_k A B_k = B_k$ and $||B_k|| \ge \frac{1}{||A||}$. Hence, $BAB = B$ and $||B|| = \alpha$. Thus, $\Omega(A)$ contains an element of minimal operator norm. □

Theorem 3.4. *Let F satisfy the assumptions of Theorem* 3.2 *except that condition* (a) *is replaced by the following condition:*

(b) *The generalized inverse $F'(x^*)^\dagger$ exists, $\|F'(x^*)^\dagger\| \le p$, and for any $x \in S(x^*, \frac{1}{3Lp})$,*

$$\dim N(F'(x)) = \dim N(F'(x^*))$$

and

$$\text{codim} R(F'(x)) = \text{codim} R(F'(x^*)).$$

Then, the conclusion of Theorem 3.2 *holds with*

$$F'(x_0)^\sharp \in \{B : B \in \Omega(F'(x_0)), ||B|| \le ||F'(x_0)^\dagger||\}. \tag{3.6}$$

Proof. Condition (a) of Theorem 3.2 ensures that for any $x \in S(x^*, r), 0 < r \le 1/3Lp$, the outer inverse $F'(x)^\sharp \in \text{argmin}\{\|B\| : B \in \Omega(F'(x))\}$ satisfies $\|F'(x)^\sharp\| \le p/(1 - Lpr)$. Now we show that under condition (b) for any $x \in S(x^*, r), 0 < r \le 1/3Lp$, the outer inverse $F'(x)^\sharp \in \{B : B \in \Omega(F'(x)), ||B|| \le ||F'(x)^\dagger||\}$ satisfies $\|F'(x)^\sharp\| \le p/(1 - Lpr)$.

From (3.4),

$$||F'(x^*)^\dagger (F'(x) - F'(x^*))|| \le p||F'(x) - F'(x^*)|| \le pL||x - x^*|| \le pLr < 1.$$

By Lemma 2.4,

$$F'(x)^\dagger = (I + F'(x^*)^\dagger (F'(x) - F'(x^*)))^{-1} F'(x^*)^\dagger$$

is the generalized inverse of $F'(x)$. By Lemma 2.1,

$$||F'(x)^\dagger|| \le \frac{||F'(x^*)^\dagger||}{1 - pLr} \le \frac{p}{1 - pLr} =: \beta.$$

Hence, for any $x_0 \in S(x^*, r)$, the outer inverse

$$F'(x_0)^\sharp \in \{B : B \in \Omega(F'(x_0)), ||B|| \leq ||F'(x_0)^\dagger||\}$$

satisfies $||F'(x_0)^\sharp|| \leq \beta$. By the same argument in the proof of Theorem 3.2, we can show that the conclusion of Theorem 3.2 holds with

$$F'(x_0)^\sharp \in \{B : B \in \Omega(F'(x_0)), \|B\| \leq \|F(x_0)^\dagger\|\}. \qquad \square$$

Remark 3.5. Let $X = R^n$ and $Y = R^m$. Then condition (a) of Theorem 3.2 holds automatically. Condition (b) of Theorem 3.4 holds if and only if $F'(x)$ is of a constant rank in $S(x^*, \frac{1}{3Lp})$. In the case of infinite dimensional spaces, condition (a) depends on the norm being used. Operator extremal properties of various generalized inverses have been studied by Engl and Nashed [11].

Remark 3.6. Rall [34] assumed that $F'(x^*)^{-1}$ exists, $||F'(x^*)^{-1}|| \leq p$,

$$||F'(x) - F'(y)|| \leq L||x - y||,$$

and $S(x^*, \frac{1}{Lp}) \subset D$ and proved that there is a ball $S(x^*, r)$ such that Kantorovich conditions hold at each $x_0 \in S(x^*, r)$. Under Rall's conditions, all conditions of Theorem 3.4 hold. Therefore, Theorem 3.4 reduces to Rall's theorem for nonsingular equations. In [39], Yamamoto and Chen compared three local convergence balls for Newton-like methods for nonsingular equations. Their results also can be generalized to singular equations using the technique in the proof of Theorem 3.2.

4. Local convergence for nonsmooth equations. In this section, we consider method (1.9) for singular nonsmooth equations with $X = R^n$ and $Y = R^m$. The discussion in this section is presented in finite dimensional spaces since for technical reasons we wish to confine ourselves to the notion of the generalized derivative of locally Lipschitzian mappings. Furthermore, because we could not restrict $||V_x - V_y||$ by $||x - y||$ for nonsmooth operators, Lemma 2.1 could not be used to construct V_k from V_0. A Kantorovich-type theorem is difficult to establish for local analysis of singular nonsmooth equations. For overcoming the difficulty, we use a Mysovskii-type theorem [27] to give a local convergence theorem for singular nonsmooth equations.

First, we give a Mysovskii theorem (semilocal convergence) for singular nonsmooth equations.

THEOREM 4.1. *Let $F : R^m \to R^n$ be locally Lipschitz. Assume that there exist $x_0 \in D$, $V_0 \in \partial_B F(x_0)$, $V_0^\sharp \in \Omega(V_0)$, and constants $\eta > 0$ and $\alpha \in (0, 1)$ such that for any $V_x \in \partial_B F(x), x \in D$, there exists an outer inverse $V_x^\sharp \in \Omega(V_x)$ satisfying $N(V_x^\sharp) = N(V_0^\sharp)$. Also assume that for this outer inverse the following conditions hold:*

$$||V_0^\sharp F(x_0)|| \leq \eta,$$

$$||V_y^\sharp(F(y) - F(x) - V_x(y - x))|| \leq \alpha||y - x|| \quad \text{ if } y = x - V_x^\sharp F(x). \tag{4.1}$$

Let $S := S(x_0, r) \subseteq D$ with $r = \eta/(1 - \alpha)$. Then, the sequence $\{x_k\}$ defined by (1.9) *with $V_k^\sharp$ satisfying $N(V_k^\sharp) = N(V_0^\sharp)$ lies in $\bar{S} = \bar{S}(x_0, r)$ and converges to a solution x^* of $V_0^\sharp F(x) = 0$ in $\bar{S}$.*

Proof. First we show that the sequence defined by method (1.9) lies in S. For $k = 1$, we have

$$||x_1 - x_0|| = ||V_0^\sharp F(x_0)|| \leq \eta = r(1 - \alpha),$$

and thus $x_1 \in S$. Suppose now $x_1, x_2, \ldots, x_k \in S$. Let $V_k^\sharp$ be an outer inverse of $V_k \in \partial_B F(x_k)$ such that $N(V_k^\sharp) = N(V_{k-1}^\sharp) = N(V_0^\sharp)$. Then, by Lemma 2.3, we have $V_k^\sharp(I - V_{k-1}V_{k-1}^\sharp) = 0$ and thus

$$\begin{aligned}
&||x_{k+1} - x_k|| = ||V_k^\sharp F(x_k)|| \\
&= ||V_k^\sharp(F(x_k) - V_{k-1}(x_k - x_{k-1}) - V_{k-1}V_{k-1}^\sharp F(x_{k-1}))|| \\
&= ||V_k^\sharp(F(x_k) - V_{k-1}(x_k - x_{k-1}) - F(x_{k-1}))|| \\
&\le \alpha||x_k - x_{k-1}|| \le \alpha^k||x_1 - x_0|| \le \alpha^k\eta = r\alpha^k(1 - \alpha).
\end{aligned}$$

Hence,

$$||x_{k+1} - x_0|| \le \sum_{j=0}^{k} ||x_{j+1} - x_j|| \le \sum_{j=0}^{k} r\alpha^j(1 - \alpha) \le r.$$

This proves that $\{x_k\} \subseteq S$. Hence, for any positive integers k and p,

$$||x_{k+p+1} - x_k|| \le \sum_{j=k}^{k+p} ||x_{j+1} - x_j|| \le \sum_{j=k}^{k+p} r\alpha^j(1 - \alpha) \le r\alpha^k.$$

So, the method (1.9) converges to a point $x^* \in S$. Since F is Lipschitz on $\bar{S}$, $||V_k||$ is uniformly bounded on $\bar{S}$. Thus, by Lemma 2.3,

$$||V_0^\sharp F(x^*)|| = \lim_{k\to\infty} ||V_0^\sharp F(x_k)|| = \lim_{k\to\infty} ||V_0^\sharp V_k V_k^\sharp F(x_k)||$$

$$\le \lim_{k\to\infty} ||V_0^\sharp||||V_k V_k^\sharp F(x_k)|| = \lim_{k\to\infty} ||V_0^\sharp||||V_k(x_{k+1} - x_k)|| = 0.$$

Therefore, $V_0^\sharp F(x^*) = 0$. □

Remark 4.2. Suppose that $m = n$ and all $V_x \in \partial F(x)$, $x \in S$ are nonsingular. Then, we have $V_x^{-1} \in \Omega(V_x)$ and $N(V_x^{-1}) = N(V_0^{-1})$. Moreover, x^* is a solution of $F(x) = 0$. Hence, Theorem 4.1 generalizes Theorem 3.3 in [33] to singular equations. Moreover, our assumptions are weaker than assumptions of Theorem 3.3 in [33] in the nonsingular case.

THEOREM 4.3. *Let $F : R^n \to R^m$ be locally Lipschitz. Let p be a positive constant. Assume that there exist a $\Gamma \in R^{n\times m}$ and an $x^* \in D$ such that $\Gamma F(x^*) = 0$; for any $V_* \in \partial_B F(x^*)$, there is an outer inverse $V_*^\sharp \in \Omega(V_*)$ satisfying $N(V_*^\sharp) = N(\Gamma)$ and $\| V_*^\sharp \| \le p$. Then, there exists a positive number r such that for any $x \in S(x^*, r)$ and $V_x \in \partial_B F(x)$ there is an outer inverse $V_x^\sharp \in \Omega(V_x)$ such that $N(V_x^\sharp) = N(\Gamma)$. Moreover assume that (4.1) holds for this outer inverse. Then, there is a $\delta \in (0, r/2]$ such that for any $x_0 \in S(x^*, \delta)$, the sequence $\{x_k\}$ defined by (1.9) with $V_k^\sharp \in \Omega(V_k)$ and $N(V_k^\sharp) = N(\Gamma)$ lies in $S(x^*, r)$ and converges to a solution $\bar{x}^*$ of $\Gamma F(x) = 0$. Furthermore, if F is semismooth at $\bar{x}^*$ and $R(V_k^\sharp) = R(V_0^\sharp)$, then the convergence rate is superlinear.*

Proof. First, we claim that for $\hat{\epsilon} \in (0, 1/p)$ there is a ball $S(x^*, r) \subset D$ with $r > 0$ such that for any $V_x \in \partial_B F(x)$, $x \in S(x^*, r)$, we have

$$\| V_x - V_* \| < \hat{\epsilon} \text{ for a } V_* \in \partial_B F(x^*). \tag{4.2}$$

If (4.2) is not true, then there is a sequence $\{y_k : y_k \in D_F\}$ with $y_k \to x^*$ such that

$$\| \nabla F(y_k) - V_* \| \geq \hat{\epsilon} \text{ for all } V_* \in \partial_B F(x^*). \tag{4.3}$$

By passing to a subsequence, we may assume that $\{\nabla F(y_k)\}$ converges to a $V_* \in \partial_B F(x^*)$. This contradicts (4.3); hence, (4.2) holds.

Suppose that $x \in S(x^*, r)$. Then, there is a $V_* \in \partial_B F(x^*)$ such that

$$\| V_x - V_* \| \leq \hat{\epsilon} < \frac{1}{p}. \tag{4.4}$$

From the assumptions of this theorem, there is a $V_*^\sharp \in \Omega(V_*)$ which satisfies $||V_*^\sharp|| \leq p$ and $N(V_*^\sharp) = N(\Gamma)$. Hence, from Lemma 2.1, we have that $V_x^\sharp = (I + V_*^\sharp(V_x - V_*))^{-1} V_*^\sharp$ is an outer inverse and

$$N(V_x^\sharp) = N(V_*^\sharp) = N(\Gamma), \qquad R(V_x^\sharp) = R(V_*^\sharp), \quad \| V_x^\sharp V_* \| \leq \frac{1}{1 - \hat{\epsilon} p} =: \beta.$$

Since F is continuous and $\|V_*^\sharp\|$ is bounded for any $\epsilon \in (0, r(1-\alpha)/2\beta]$, there exists a $\delta \in (0, r/2)$ such that for any $x \in S(x^*, \delta), ||V_*^\sharp F(x)|| < \epsilon$. Therefore, for any $x_0 \in S(x^*, \delta)$, $V_0 \in \partial_B F(x_0)$, there exists $V_* \in \partial_B F(x^*)$ such that (4.2) holds. Moreover, there exist $V_0^\sharp \in \Omega(V_0)$ and $V_*^\sharp \in \Omega(V_*)$ such that

$$N(V_0^\sharp) = N(V_*^\sharp) = N(\Gamma)$$

and

$$||V_0^\sharp F(x_0)|| = ||V_0^\sharp V_* V_*^\sharp F(x_0)|| \leq ||V_0^\sharp V_*|| ||V_*^\sharp F(x_0)|| \leq \beta\epsilon \leq r(1-\alpha)/2.$$

Since $x_0 \in S(x^*, r/2)$, we have $S(x^0, r/2) \subset S(x^*, r)$. Hence, all conditions of Theorem 4.1 hold with $\eta = r(1-\alpha)/2$. By Theorem 4.1 for any $x_0 \in S(x^*, \delta)$, the sequence $\{x_k\}$ defined by method (1.9) with $V_k^\sharp$ satisfying (4.1) lies in $S(x^*, r)$ and converges to a solution $\bar{x}^*$ of $V_0^\sharp F(x) = 0$. Since $N(V_0^\sharp) = N(\Gamma)$, we have $\Gamma F(\bar{x}^*) = 0$.

Now, we prove that the convergence rate is superlinear.

Since $x_k \in S(x^*, r)$, there is a $V_*^\sharp \in \Omega(V_*)$ such that

$$N(V_k^\sharp) = N(V_*^\sharp) \quad \text{and} \quad \|V_k - V_*\| < 1/p.$$

From Lemma 2.3, $V_k^\sharp = V_k^\sharp V_* V_*^\sharp$ and

$$\|V_k^\sharp\| \leq \|V_k^\sharp V_*\| \|V_*^\sharp\| \leq \beta p,$$

i.e., $\|V_k^\sharp\|$ is bounded. Since $N(V_k^\sharp) = N(\Gamma)$ and $\Gamma F(\bar{x}^*) = 0$, $V_k^\sharp F(\bar{x}^*) = 0$. Since $R(V_k^\sharp) = R(V_0^\sharp)$, $x_{k+1} - \bar{x}^* \in R(V_k^\sharp)$. Hence,

$$\begin{aligned}
&\| x_{k+1} - \bar{x}^* \| \\
&= ||V_k^\sharp V_k(x_{k+1} - \bar{x}^*)|| \\
&= ||V_k^\sharp V_k(x_k - V_k^\sharp(F(x_k) - F(\bar{x}^*)) - \bar{x}^*)|| \\
&= ||V_k^\sharp(V_k(x_k - \bar{x}^*) - F(x_k) + F(\bar{x}^*))|| \\
&\leq \| V_k^\sharp \| (\| F(x_k) - F(\bar{x}^*) - F'(\bar{x}^*; x_k - \bar{x}^*) \| \\
&\quad + \| V_k(x_k - \bar{x}^*) - F'(\bar{x}^*; x_k - \bar{x}^*) \|).
\end{aligned}$$

By (2.1) and Lemma 2.7, we have

$$\| F(x) - F(\bar{x}^*) - F'(\bar{x}^*; x - \bar{x}^*) \| = o(\| x - \bar{x}^* \|)$$

and

$$\| V_x(x - \bar{x}^*) - F'(\bar{x}^*; x - \bar{x}^*) \| = o(\| x - \bar{x}^* \|).$$

This implies

$$\| x_{k+1} - \bar{x}^* \| = o(\| x_k - \bar{x}^* \|).$$

Hence, method (1.9) converges to $\bar{x}^*$ superlinearly. □

Remark 4.4. Suppose that there is a $V_0^\sharp \in \Omega(V_0)$ satisfying $N(V_0^\sharp) = N(\Gamma)$. If there is a $V_k \in \partial_B F(x_k)$ such that

$$\|V_0^\sharp(V_k - V_0)\| < 1, \tag{4.5}$$

then $V_k^\sharp = (I + V_0^\sharp(V_k - V_0))^{-1} V_0^\sharp$ is an outer inverse of V_k with $N(V_k^\sharp) = N(V_0^\sharp) = N(\Gamma)$ and $R(V_k^\sharp) = R(V_0^\sharp)$. Because of the nonsmoothness, we do not have the benefit of a condition such as (3.4) to ensure (4.5). This is a major difference between smooth and nonsmooth equations.

Remark 4.5. Superlinear convergence results in [31, 33] assume that all $V_* \in \partial_B F(x^*)$ are nonsingular. Under this assumption, x^* is the unique solution of $F(x) = 0$ in a neighborhood $\mathcal{N}_*$ of x^*, and it was only shown that $\|x_{k+1} - x^*\| = o(\|x_k - x^*\|)$ for a sequence $\{x_k\} \subset \mathcal{N}_*$. In the singular case, the set of solutions may be locally a manifold of positive dimension. There is no neighborhood $\mathcal{N}_*$ of x^* such that method (1.9) converges to x^* for any close starting point $x_0 \in \mathcal{N}_*$. Condition (4.1) is imposed in Theorem 4.3 to guarantee the existence of a neighborhood $\mathcal{N}_*$ such that method (1.9) converges to a solution of $V_0^\sharp F(x) = 0$ for any starting point $x_0 \in \mathcal{N}_*$. The following corollary shows that condition (4.1) can be replaced by special choices of starting points.

COROLLARY 4.6. *Let p be a positive number, Γ be an $n \times m$ matrix, and x^* be a solution of $\Gamma F(x) = 0$. Suppose that F is semismooth at x^* and, for all $V_* \in \partial_B F(x^*)$, there exists a $V_*^\sharp \in \Omega(V_*)$ such that $N(V_*^\sharp) = N(\Gamma)$ and $||V_*^\sharp|| \le p$. Then method* (1.9) *with $V_k^\sharp$ satisfying $N(V_k^\sharp) = N(\Gamma)$, $R(V_k^\sharp) = R(V_0^\sharp)$, and $x_0 \in R(V_0^\sharp) + x^*$ is convergent to x^* superlinearly in a neighborhood of x^*.*

Proof. From the proof of Theorem 4.3, we have that there is a ball $S(x^*, \bar{r})$ such that for any $V_x \in \partial_B F(x), x \in S(x^*, \bar{r})$, there is an outer inverse $V_x^\sharp \in \Omega(V_x)$ satisfying $N(V_x^\sharp) = N(\Gamma)$ and $||V_x^\sharp V_*|| \le \beta$ for a $V_* \in \partial_B F(x^*)$. Choosing $x_0 \in R(V_0^\sharp) + x^*$, we have $x_{k+1} - x^* \in R(V_k^\sharp)$ since $R(V_k^\sharp) = R(V_0^\sharp)$ and $x_{k+1} - x_k \in R(V_k^\sharp)$. Then we can show the superlinear convergence by the last part of the proof of Theorem 4.3 (superlinear convergence). □

Remark 4.7. If we assume that $m = n$ and all $V_* \in \partial_B F(x^*)$ are nonsingular, then we can take $\Gamma = I \in R^{n \times n}$. Furthermore, there is a neighborhood $\mathcal{N}_*$ of x^* such that for any $x \in \mathcal{N}_*$, all $V_x \in \partial_B F(x)$ are nonsingular. Then for any $x_k \in \mathcal{N}_*$, we can take $V_k^\sharp = V_k^{-1}$, which satisfies $N(V_k^\sharp) = N(\Gamma) = \{0\}$, $R(V_k^\sharp) = R(V_0^\sharp) = R^n$, and $x_k \in R(V_0^\sharp) + x^*$. Hence, Theorem 4.3 and Corollary 4.6 generalize the local convergence theorems given in [31, 33].

REMARK 4.8. In Theorem 4.3 and Corollary 4.6, $V_k^\sharp$ should be chosen to satisfy $N(V_k^\sharp) = N(V_0^\sharp)$ and $R(V_k^\sharp) = R(V_0^\sharp)$ at each step of (1.9). There exists an outer inverse $V_k^\sharp$ such that $N(V_k^\sharp) = N(V_0^\sharp)$ and $R(V_k^\sharp) = R(V_0^\sharp)$ if and only if $N(V_0^\sharp)$ and $R(V_k V_0^\sharp)$ are complementary subspaces of R^m. If such an outer inverse exists, it is unique [2; p. 62]. Now we give a method for numerical construction of such an outer inverse based on $V_0^\sharp$ and V_k. Let $s =$rank$(V_0^\sharp)$ and rank$(V_k) \geq s$. Let U be a matrix whose columns form a basis for $R(V_0^\sharp)$, and let W be a matrix whose rows form a basis for the orthogonal complement of $N(V_0^\sharp)$. Then WV_kU is an $s \times s$ matrix with rank s, so it is invertible. Let $V_k^\sharp := U(WV_kU)^{-1}W$. Then $V_k^\sharp$ is an outer inverse of V_k with $N(V_k^\sharp) = N(V_0^\sharp)$ and $R(V_k^\sharp) = R(V_0^\sharp)$.

5. Examples and numerical experiments. In this section we give methods for constructing outer inverses which are needed in the theorems and illustrate our results with three examples from nonlinear complementarity problems. The first example compares the theorems given in this paper with earlier results. The second example shows how outer inverses apply while generalized inverses fail for Newton's method. The third example tests the methods (1.9) and (1.10) for problems with different dimensions. The performance of algorithms is given by using Matlab 4.2c on a Sun 2000 workstation.

5.1. Calculation of outer inverses. Methods for constructing outer inverses of a given matrix or a linear operator are given in [2, 23, 24, 26]. For the case of an $m \times n$ matrix A with rank $r > 0$, we have a method using singular value decomposition (SVD). Let $A = V\Sigma U^T$, where Σ is a diagonal matrix of the same size as A and with nonnegative diagonal elements in decreasing order; V and U are $m \times m$ and $n \times n$ orthogonal matrices, respectively. Let $\epsilon > 0$ be a computational error control, $\Sigma_s^\sharp$=diag$(v_1, v_2, \ldots, v_s, 0, \ldots, 0) \in R^{n\times m}$, where $s \leq$min(m, n) and

$$v_i = \begin{cases} \sigma_{i,i}^{-1}, & \sigma_{i,i} > |\epsilon|, \\ 0, & \text{otherwise.} \end{cases}$$

Then $U\Sigma_s^\sharp V^T$ is an outer inverse of A.

As we know, orthogonal-triangular decomposition (QR) is less expensive than SVD. Here we give a new method to construct an outer inverse of A by using QR.

Let $A = QR$ be a factorization, where Q is an $m \times m$ orthogonal matrix, R is an $m \times n$ upper triangular matrix of the form

$$R = \begin{pmatrix} R_{11} & R_{12} \\ 0 & 0 \end{pmatrix},$$

and R_{11} is an $r \times r$ matrix of rank r. Then $A^\sharp = R^\sharp Q^T$ is an outer inverse of A, where

$$R^\sharp = \begin{pmatrix} \bar{R}_{11}^{-1} & 0 \\ 0 & 0 \end{pmatrix},$$

and $\bar{R}_{11}$ is an $\bar{r} \times \bar{r}$ matrix of rank $\bar{r} \leq r$.

Remark 5.1. Outer inverses of a matrix A are more stable than the generalized inverses when some singular values of A are close to zero because we can choose $\Sigma^\sharp$ and $\bar{R}^\sharp$ such that their elements are bounded. For details of the perturbation analysis that demonstrates stability of certain selections of outer inverses, see [22, 24, 26].

Remark 5.2. For the smooth case, we need not construct an outer inverse at each step but only at the starting point. In practice, method (1.10) is implemented in the following form: $x_1 = x_0 - F'(x_0)^\sharp F(x_0)$. For $k \geq 1$, we let $x_{k+1} = x_k + d$, where d is the unique solution of the linear system

$$(I + F'(x_0)^\sharp (F'(x_k) - F'(x_0)))d = -F'(x_0)^\sharp F(x_k).$$

Obviously, if $m = n$ and $F'(x_k)$ is invertible, then the method reduces to $F'(x_k)d = -F(x_k)$.

5.2. Numerical examples. As we stated in section 1, an NCP can be formulated as a system of smooth equations by (1.7) and also as a system of nonsmooth equations by (1.8). We can solve the smooth equations (1.7) by method (1.10) and the nonsmooth equations (1.8) by method (1.9). The singularity occurs very often in solving these two systems. It is interesting to see how to overcome the singularity by methods (1.9) and (1.10) and Theorems 3.2 and 4.3.

Example 1. We consider the following example [12]. Let

$$f(x) = (1 - x_2, x_1)^T.$$

The solution set of the associated NCP is $W = \{(0, \alpha), |\ 0 \leq \alpha \leq 1\}$. For each $x^* \in W$, the Jacobian of $\hat{F}$ defined by (1.7) is

$$\hat{F}'(x^*) = \begin{pmatrix} -2(1-\alpha) & 0 \\ -2\alpha & 0 \end{pmatrix},$$

which is singular. Hence, previous local convergence theorems of Newton's method [9, 27, 34] are not applicable for this example. Now we apply Theorem 3.2 to this problem. We take $x^* = (0, 0.5)$; then,

$$\hat{F}'(x^*)^\sharp = \begin{pmatrix} a & -1-a \\ a & -1-a \end{pmatrix} \in \Omega(F'(x^*))$$

for any $a \in (-\infty, \infty)$. We take $a = -1$, $L = 4$, and $p = 1$. Then we can show that there is a ball $S(x^*, r)$ with $0 < r < 0.5$ such that $||\hat{F}'(x^*)^\sharp||_\infty \leq p$ and, for any $x \in S(x^*, r)$, $||\hat{F}'(x) - \hat{F}'(y)||_\infty \leq L||x - y||_\infty$. Hence, all conditions of Theorem 3.2 hold.

Now we consider the nonsmooth equation (1.8). For any $x^* \in W$,

$$V_* = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \in \partial_B \tilde{F}(x^*).$$

This implies that F is not strongly BD-regular at all solutions in the sense of [31, 33]. Hence, the local convergence theorems in [31, 33] are not applicable for this example. However, we can choose an outer inverse as

$$V_*^\sharp = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \in \Omega(V_*).$$

Take $x^* = (0.0, 0.0)$. Then, F is nonsmooth at x^*. There is a ball $S(x^*, r)$ with $0 < r < 0.5$ such that for any $x \in S(x^*, r)$, the generalized Jacobian is

$$V(x) = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \quad \text{or} \quad V(x) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

(For determination of a $V_x \in \partial_B F(x)$, see [7, 31].)

Hence, we can take outer inverses $V_x^\sharp = V_*^\sharp$. Note that this is a linear complementarity problem; all conditions of Theorem 4.3 hold. Furthermore, for any $x_0 = (\alpha, \beta) \in S(x^*, r)$, $x_1 = x_0 - V_0^\sharp \tilde{F}(x_0) = (0, \beta)$. If $\beta \geq 0$, then $(0, \beta) \in W$.

Example 2. The generalized inverse $A^\dagger$ of a matrix A is an outer inverse, but $A^\dagger$ may not be a good outer inverse for Newton's method. This example, given by one of the referees, illustrates that conditions of Theorem 4.3 and Corollary 4.6 fail when we use generalized inverses. However, these conditions hold for a number of outer inverses.

Consider the piecewise linear equation

$$F(x_1, x_2) = \min(2x_1 + x_2 - 2, -2x_1 + x_2 - 2).$$

The solution of $F(x) = 0$ is the union of the two rays:

$$\{(x_1, x_2) : x_1 \leq 0, x_2 = -2x_1 + 2\} \cup \{(x_1, x_2) : x_1 \geq 0, x_2 = 2x_1 + 2\}.$$

This particular function, though nonsmooth, is well behaved in that its set of zeros is a 1-dimensional manifold, just as the solution of a linear equation in two variable is (usually) a line.

For $x_1 \leq 0$, let $V = V_1 = [2, 1] \in \partial_B F(x)$ and

$$V_1^\dagger = \begin{bmatrix} 2/5 \\ 1/5 \end{bmatrix}.$$

Similarly, for $x_1 > 0$, let $V = V_2 = [-2, 1] \in \partial_B F(x)$;

$$V_2^\dagger = \begin{bmatrix} -2/5 \\ 1/5 \end{bmatrix}.$$

It can be seen that for any starting point $x^0 = (x_1^0, x_2^0)$ such that $x_2^0 \leq -(1/2)x_1^0 + 2$ and $x_2^0 \leq (1/2)x_1^0 + 2$ (for instance $x^0 = (0, 0)$), the Newton's method defined by

$$x^{k+1} = x^k - V_i^\dagger F(x^k),$$

where $i = 1$ if $x_1^k \leq 0$ and $i = 2$ otherwise, converges linearly but not superlinearly to $\bar{x} = (0, 2)$. The convergence is only linear because, although $N(V_1^\dagger) = N(V_2^\dagger) = \{0\}$, we have $R(V_1^\dagger) \neq R(V_2^\dagger)$. We also see $\|V_2^\dagger(V_1 - V_2)\| = 8/5 > 1$. Thus, Lemma 2.1 cannot be applied; likewise, (4.1) fails to hold in this case.

Now we consider the use of outer inverses. It is easy to verify

$$\Omega(V_1) = \left\{ \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, 2\alpha + \beta = 1, |\alpha| + |\beta| \neq 0 \right\}$$

and

$$\Omega(V_2) = \left\{ \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, -2\alpha + \beta = 1, |\alpha| + |\beta| \neq 0 \right\}.$$

For any $V_2^\sharp \in \Omega(V_2)$ with $\alpha < 1/4$, $\|V_2^\sharp(V_1 - V_2)\|_2 < 1$. By Lemma 2.1, $V_1^\sharp = (I + V_2^\sharp(V_1 - V_2))^{-1} V_2^\sharp$ is an outer inverse of V_1 with $N(V_1^\sharp) = N(V_2^\sharp)$ and $R(V_1^\sharp) = R(V_2^\sharp)$. For instance, we choose

$$V_2^\sharp = \begin{pmatrix} -1/5 \\ 3/5 \end{pmatrix}.$$

Table 1

n	(1.10) for smooth eq.			(1.9) for nonsmooth eq.		
	k	$\|\|F(x_k)^\sharp F(x_k)\|\|$	cputime	k	$\|\|V_k^\sharp F(x_k)\|\|$	cputime
50	13	2.83×10^{-15}	31.85	5	1.0×10^{-16}	3.97
100	12	1.25×10^{-14}	144.15	4	1.0×10^{-16}	15.18
200	8	3.10×10^{-10}	586.97	4	1.0×10^{-16}	137.20
350	14	4.91×10^{-14}	4.92×10^3	4	1.0×10^{-16}	577.95

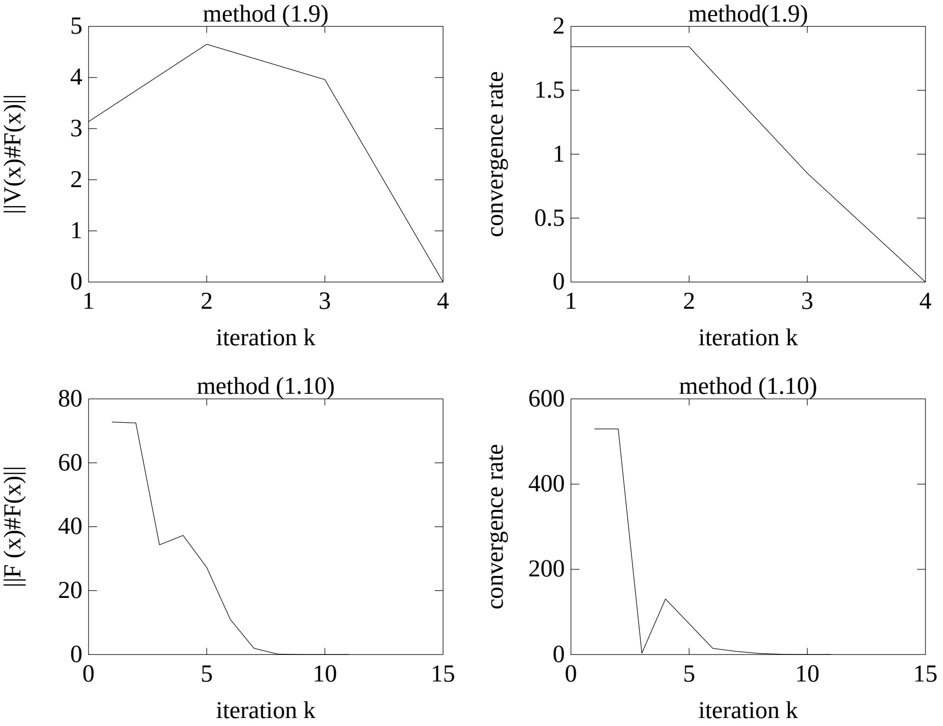


Fig. 1. *Computational results for Example* 3.

Then,

$$V_1^\sharp = (I + V_2^\sharp(V_1 - V_2))^{-1} V_2^\sharp = \begin{pmatrix} -1 \\ 3 \end{pmatrix}.$$

Furthermore, for the starting point $x^0 = (0, 0)$,

$$x^1 = x^0 - V_1^\sharp F(x^0) = \begin{pmatrix} -2 \\ 6 \end{pmatrix}$$

is a solution of $F(x) = 0$.

Example 3. To compare methods (1.9) and (1.10), we randomly generate an NCP with

$$f(x) = Ax^2 + Bx + c,$$

where A and B are $n \times n$ matrices, $c \in R^n$ is a vector, and $x^2 = (x_i^2) \in R^n$.

We first randomly generate singular matrices A and B and a nonnegative vector x^*, which has some zero elements. Then we choose c such that x^* is a solution of an NCP with $f(x) = Ax^2 + Bx + c$. The problem is randomly generated but with known solution characteristic and singularity, so we can test the efficiency of methods (1.9) and (1.10). Table 1 summarizes computational results with different n. Also, we choose $n = 100$, $x_0 = x^*$+random vector and show $||F'(x_k)^\sharp F(x_k)||, ||V_k^\sharp F(x_k)||$ and convergence rate $||F(x_{k+1})||/||x_{k+1} - x_k||$ in Figure 1.

6. Concluding remarks. In this paper, we discussed local convergence of Newton's method for singular smooth and nonsmooth equations, respectively. These results generalize and extend earlier results on nonsingular smooth and nonsmooth equations. Singularity occurs in many areas of optimization and numerical analysis. Pang–Gabriel [29] mentioned that the singularity badly affected the convergence of the NE/SQP method for the NCP in a number of numerical examples. The results in this paper present a strategy to treat singularity and to guarantee convergence of Newton's method and related iterative methods. Some of our results are also stronger than earlier results in the nonsingular case since they involve weaker assumptions.

Acknowledgments. We are grateful to the referees for their constructive comments.

REFERENCES

[1] A. Ben-Israel, *A Newton-Raphson method for the solution of equations*, J. Math. Anal. Appl., 15 (1966), pp. 243–253.

[2] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, Wiley, New York, 1974.

[3] B. Chen and P. Harker, *A noninterior-point continuation method for linear complementarity problems*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1168–1190.

[4] X. Chen, *On the convergence of Broyden-like methods for nonlinear equations with nondifferentiable terms*, Ann. Inst. Statist. Math., 42 (1990), pp. 387–401.

[5] X. Chen and L. Qi, *Parameterized Newton method and Broyden-like method for solving nonsmooth equations*, Comput. Optim. Appl., 3 (1994), pp. 157–179.

[6] X. Chen and T. Yamamoto, *On the convergence of some quasi-Newton methods for nonlinear equations with nondifferentiable operators*, Computing, 48 (1992), pp. 87–94.

[7] X. Chen and T. Yamamoto, *Newton-like methods for solving underdetermined nonlinear equations*, J. Comput. Appl. Math., 55 (1995), pp. 311–324.

[8] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley and Sons, New York, 1983.

[9] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1983.

[10] P. Deuflhard and G. Heindl, *Affine invariant convergence theorems for Newton's method and extensions to related methods*, SIAM J. Numer. Anal., 16 (1979), pp. 1–10.

[11] H. W. Engl and M. Z. Nashed, *New extremal characterizations of generalized inverses of linear operators*, J. Math. Anal. Appl., 82 (1981), pp. 566–586.

[12] M. Ferris and S. Lucidi, *Globally Convergent Methods for Nonlinear Equations*, Computer Sciences Department, University of Wisconsin—Madison, Madison, WI, 1991, preprint 1030.

[13] S. P. Han, J. S. Pang, and N. Rangaraj, *Globally convergent Newton methods for nonsmooth equations*, Math. Oper. Res., 17 (1992), pp. 586–607.

[14] W. M. Häussler, *A Kantorovich-type convergence analysis for the Gauss-Newton method*, Numer. Math., 48 (1986), pp. 119–125.

[15] M. Heinkenschloss, C. T. Kelley, and H. T. Tran, *Fast algorithms for nonsmooth compact fixed point problems*, SIAM J. Numer. Anal., 29 (1992), pp. 1769–1792.

[16] C. M. Ip and J. Kyparisis, *Local convergence of quasi -Newton methods for B-differentiable equations*, Math. Programming, 56 (1992), pp. 71–89.

[17] M. Kojima and S. Shindo, *Extensions of Newton and quasi-Newton methods to systems of PC^1 equations*, J. Oper. Res. Soc. Japan, 29 (1986), pp. 352–374.

[18] B. Kummer, *Newton's method for non-differentiable functions*, in Advances in Mathematical Optimization, J. Guddat, B. Bank, H. Hollatz, P. Kall, D. Klatte, B. Kummer, K. Lommatzsch, L. Tammer, M. Vlach, and K. Zimmerman, eds., Akademi-Verlag, Berlin, 1988, pp. 114–125.

[19] O. L. Mangasarian, *Equivalence of the complementarity problem to a system of nonlinear equations*, SIAM J. Appl. Math., 31 (1976), pp. 89–92.

[20] O. L. Mangasarian and M. V. Solodov, *Nonlinear complementarity as unconstrained and constrained minimization*, Math. Programming, 62 (1993), pp. 277–297.

[21] R. Mifflin, *Semismooth and semiconvex functions in constrained optimization*, SIAM J. Control Optim., 15 (1977), pp. 191–207.

[22] R. H. Moore and M. Z. Nashed, *Approximations to generalized inverses of linear operators*, SIAM J. Appl. Math., 27 (1974), pp. 1–16.

[23] M. Z. Nashed, ed., *Generalized Inverses and Applications*, Academic Press, New York, 1976.

[24] M. Z. Nashed, *Generalized inverse mapping theorems and related applications of generalized inverses,* in Nonlinear Equations in Abstract Spaces, V. Lakshmikantham, ed., Academic Press, New York, 1978, pp. 217–252.

[25] M. Z. Nashed, *Inner, outer, and generalized inverses in Banach and Hilbert spaces*, Numer. Funct. Anal. Optim., 9 (1987), pp. 261–325.

[26] M. Z. Nashed and X. Chen, *Convergence of Newton-like methods for singular operator equations using outer inverses*, Numer. Math., 66 (1993), pp. 235–257.

[27] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[28] J-S. Pang, *Newton's method for B–differentiable equations*, Math. Oper. Res., 15 (1990), pp. 311–341.

[29] J-S. Pang and S. A. Gabriel, *NE/SQP: A robust algorithm for the nonlinear complementarity problem*, Math. Programming, 60 (1993), pp. 295–337.

[30] J-S. Pang and L. Qi, *Nonsmooth equations: Motivation and algorithms*, SIAM J. Optim., 3 (1993), pp. 443–465.

[31] L. Qi, *Convergence analysis of some algorithms for solving nonsmooth equations*, Math. Oper. Res., 18 (1993), pp. 227–244.

[32] L. Qi and X. Chen, *A globally convergence successive approximation method for severely nonsmooth equations*, SIAM J. Control Optim., 33 (1995), pp. 402–418.

[33] L. Qi and J. Sun, *A nonsmooth version of Newton's method,* Math. Programming, 58 (1993), pp. 353–367.

[34] L. B. Rall, *A note on the convergence of Newton's method*, SIAM J. Numer. Anal., 1 (1974), pp. 34–36.

[35] S. M. Robinson, *Newton's method for a class of nonsmooth functions*, Set-Valued Anal., 2 (1994), pp. 291–305.

[36] A. Shapiro, *On concepts of directional differentiability*, J. Optim. Theory Appl., 66 (1990), pp. 477–487.

[37] T. Yamamoto, *On some convergence theorems for the Gauss-Newton method for solving nonlinear least squares problem*, in Proc. Summer Symposium on Computation in Mathematical Sciences–Fusion of Symbolic and Numerical Computations, IIAS-SIS Fujitsu Ltd., 1987, pp. 94–103.

[38] T. Yamamoto, *Uniqueness of the solution in a Kantorovich-type theorem of Häußler for the Gauss-Newton method*, Japan J. Appl. Math., 6 (1989), pp. 77–81.

[39] T. Yamamoto and X. Chen, *Ball-convergence theorems and error estimates for certain iterative methods for nonlinear equations*, Japan J. Appl. Math., 7 (1990), pp. 131–143.

[40] N. Yamashita and M. Fukushima, *Modified Newton methods for solving semismooth reformulations of monotone complementarity problems*, Math. Programming, to appear.

SIAM J. OPTIM.
Vol. 7, No. 2, pp. 463–480, May 1997

011

NEWTON AND QUASI-NEWTON METHODS FOR A CLASS OF NONSMOOTH EQUATIONS AND RELATED PROBLEMS[*]

DEFENG SUN[†] AND JIYE HAN[†]

Abstract. The paper presents concrete realizations of quasi-Newton methods for solving several standard problems including complementarity problems, special variational inequality problems, and the Karush–Kuhn–Tucker (KKT) system of nonlinear programming. A new approximation idea is introduced in this paper. The Q-superlinear convergence of the Newton method and the quasi-Newton method are established under suitable assumptions, in which the existence of $F'(x^*)$ is not assumed. The new algorithms only need to solve a linear equation in each step. For complementarity problems, the QR factorization on the quasi-Newton method is discussed.

Key words. nonsmooth equations, Newton method, quasi-Newton method, Q-superlinear convergence

AMS subject classifications. 90C30, 90C33

PII. S1052623494274970

1. Introduction.

In recent years, many authors have considered various forms of Newton methods for solving nonsmooth equations (NE) (see, e.g., [17, 18, 19, 20, 11, 12, 13, 21, 22, 23, 26]). Some authors have also considered the application of the quasi-Newton methods to nonsmooth equations. In Kojima and Shindo [11], the quasi-Newton method was applied to piecewise smooth equations. When the iteration sequence moves to a new C^1-piece, a new approximate starting matrix is needed. Ip and Kyparisis [9] considered the local convergence of quasi-Newton methods directly applied to B-differentiable equations (in the sense of Robinson [25]). The superlinearly convergent theorems are established under the assumption that F is strongly F-differentiable [15] at the solution.

The main object of this paper is to construct a practical quasi-Newton method for nonsmooth equations, especially for those which are of concrete background. In order to complete this, we first give a slight modification of the generalized Newton method [21, 22, 13]. Based on the modified generalized Newton method, we give a quasi-Newton method for solving a class of nonsmooth equations, which arises from the complementarity problem, variational inequality problem, the Karush–Kuhn–Tucker (KKT) system of nonlinear programming, and related problems. In each step, we only need to solve a linear equation. The Q-superlinear convergence is established under mild conditions.

The characteristics of the quasi-Newton method for solving (4.12) established in section 4 include the following: (i) without assuming the existence of $F'(x^*)$, we prove the Q-superlinearly convergent property; (ii) only one approximate starting matrix is needed; and (iii) from the QR factorization of the kth iterate matrix we need at most $O((I(k)+1)n^2)$ arithmetic operations to get the QR factorization of the $(k+1)$th iterate matrix (for the definition of $I(k)$, see (5.8)).

The remainder of this paper is organized as follows. In section 2, we give some preliminaries on nonsmooth functions. In section 3, we propose a modified generalized

[*] Received by the editors September 30, 1994; accepted for publication (in revised form) November 14, 1995. This work was supported by the National Natural Science Foundation of China under project 19371084.

http://www.siam.org/journals/siopt/7-2/27497.html

[†] Institute of Applied Mathematics, Academia Sinica, Beijing 100080, P. R. China (sun@maths.unsw.edu.au, jyhan%amath3@amath3.amt.ac.cn).

Newton method. In section 4, we give a quasi-Newton method for solving a class of nonsmooth equations. In section 5, we discuss the implementation of the quasi-Newton method for the nonlinear complementarity problem. The KKT system of variational inequality problems with upper and lower bounds are discussed in section 6. The computational results are given in section 7.

2. Preliminaries.

In general, assume that $F : R^n \to R^m$ is locally Lipschitzian. In order to reduce the nonsingularity assumption of the generalized Newton method [22], the concept $\partial_B F(x)$ was introduced by Qi [21]:

$$\partial_B F(x) = \Big\{ \lim_{\substack{x^k \to x \\ x^k \in D_F}} F'(x^k) \Big\}, \tag{2.1}$$

where D_F is the set where F is differentiable. Let ∂F be the generalized Jacobian of F in the sense of Clarke [4]. Then $\partial F(x)$ is the convex hull of $\partial_B F(x)$,

$$\partial F(x) = \text{conv } \partial_B F(x). \tag{2.2}$$

For $m = 1$, $\partial_B F(x)$ was introduced by Shor [28]. Here, we denote

$$\partial_b F(x) = \partial_B F_1(x) \times \partial_B F_2(x) \times \cdots \times \partial_B F_m(x). \tag{2.3}$$

When $m = 1$, $\partial_b F(x) = \partial_B F(x)$.

We say that F is *semismooth* at x if

$$\lim_{\substack{V \in \partial F(x+th') \\ h' \to h,\ t \downarrow 0}} \{Vh'\} \tag{2.4}$$

exists for any $h \in R^n$. Semismoothness was originally introduced by Mifflin [14] for functionals. Convex functions, smooth functions, and piecewise linear functions are examples of semismooth functions. Scalar productions and sums of semismooth functions are still semismooth functions (see [14]). In [23], Qi and Sun extended the definition of semismooth functions to $F : R^n \to R^m$. It was proved in [23] that F is semismooth at x if and only if all its component functions are so.

Condition (2.4) is stronger than the assumption that for any $h \in R^n$,

$$\lim_{\substack{V \in \partial F(x+th) \\ t \downarrow 0}} \{Vh\} \tag{2.5}$$

exists. Under the latter assumption, Qi and Sun [Proposition 2.1, 22] proved that the classical derivative

$$F'(x; h) = \lim_{t \downarrow 0} \frac{F(x+th) - F(x)}{t}$$

exists and is equal to the limit in (2.5); i.e.,

$$F'(x; h) = \lim_{\substack{V \in \partial F(x+th) \\ t \downarrow 0}} \{Vh\}. \tag{2.6}$$

If the right-hand side limit in (2.6) is uniformly convergent for all h with unit norm, then from Theorem 2.3 of [22] we have that F is semismooth at x. In [13], Kummer discussed sufficient and necessary conditions for the convergence of the Newton method based on generalized derivatives. One of the conditions for guaranteeing convergence (see Theorem 2 of [13]) is (specialized to the fourth case discussed in [13]) that for any $V \in \partial F(x+h),\ h \to 0$,

$$F(x+h) - F(x) - Vh = o(\|h\|). \tag{2.7}$$

Since F is locally Lipschitz continuous, from [27] we know that if $F'(x;h)$ exists, then $F'(x;h)$ coincides with the B-derivative of F at x; i.e.,

$$\lim_{h\to 0} \frac{F(x+h) - F(x) - F'(x;h)}{\|h\|} = 0. \tag{2.8}$$

So, if $F'(x;h)$ exists, then (2.7) implies that for any $V \in \partial F(x+h),\ h \to 0$,

$$Vh - F'(x;h) = o(\|h\|). \tag{2.9}$$

Again, (2.9) implies the semismoothness of F at x from Theorem 2.3 of [22]. But in [13], Kummer also discussed the case that $F'(x;h)$ may not exist. In this paper we will only consider the case that $F'(x;h)$ exists. Under the existence assumption of $F'(x;h)$, similar to the above discussion from Theorem 2.3 of [22], we can prove that in finite dimensional space the condition (CA*) in Theorem 2 of [13] implies (2.9) (by assuming $F(x) = 0$), which is essentially equivalent to the semismoothness of F at x. Semismoothness is a useful tool in proving the Q-superlinear convergence of the generalized Newton method for nonsmooth equations [21, 22, 23]. We also need it in this paper. In addition, Kummer [13] discussed the approximation of Newton matrices and errors when solving the auxiliary problems. In this paper we will put our main attention on constructing concrete quasi-Newton methods for solving special nonsmooth equations and will not discuss the inexact solution of the subproblems.

LEMMA 2.1 (see [22]). *Suppose that $F : R^n \to R^m$ is a locally Lipschitzian function and semismooth at x. Then*

(1) *for any $V \in \partial F(x+h),\ h \to 0$,*

$$Vh - F'(x;h) = o(\|h\|);$$

(2) *for any $h \to 0$,*

$$F(x+h) - F(x) - F'(x;h) = o(\|h\|).$$

In the rest of this paper, let $\|\cdot\|$ denote the l_2 vector norm or its induced matrix norm.

LEMMA 2.2. *Suppose that $F : R^n \to R^n$ is a locally Lipschitzian function. If all $V \in \partial_b F(x)$ are nonsingular, then there exists a positive constant β such that*

$$\|V^{-1}\| \le \beta$$

for any $V \in \partial_b F(x)$. Furthermore, there exists a neighborhood $N(x)$ of x such that for any $y \in N(x)$, all $W \in \partial_b F(y)$ are nonsingular and satisfy

$$\|W^{-1}\| \le \frac{10}{9}\beta. \tag{2.10}$$

Proof. From the definition of $\partial_b F$ we can easily know that $\partial_b F(\cdot)$ is bounded and closed in a neighborhood of x. Then the proof of the theorem is similar to that of [21, 22]. We omit the detail here. □

3. Newton method for nonsmooth equations. Suppose that $F : R^n \to R^n$ is locally Lipschitzian. We are interested in finding a solution of the equation

$$F(x) = 0. \tag{3.1}$$

Qi and Sun [22], Qi [21], and Kummer [13] considered various forms of the Newton method for solving (3.1) when F is not F-differentiable. Here we will consider the following slightly modified Newton method

$$x^{k+1} = x^k - V_k^{-1} F(x^k), \quad k = 0, 1, \dots, \tag{3.2}$$

where $V_k \in \partial_b F(x^k)$. This method is useful to establish the superlinear convergence of quasi-Newton methods given in section 4. Similar to that of [21, 22], we can give the following convergence theorem.

THEOREM 3.1. *Suppose that x^* is a solution of* (3.1)*, F is locally Lipschitzian and semismooth at x^*, and all $V_* \in \partial_b F(x^*)$ are nonsingular. Then the iteration method* (3.2) *is well defined and converges to x^* Q-superlinearly in a neighborhood of x^*.*

Proof. By Lemma 2.2, (3.2) is well defined in a neighborhood of x^* for the first step $k = 0$. Since $V_k \in \partial_b F(x^k)$, the ith row V_k^i of V_k satisfies

$$V_k^i \in \partial_B F_i(x^k).$$

From the semismoothness of F we know that F_i is semismooth at x^*. By Lemma 2.1,

$$V_k^i(x^k - x^*) - F_i'(x^*; x^k - x^*) = o(\|x^k - x^*\|), \ i = 1, \dots, n.$$

Therefore,

$$V_k(x^k - x^*) - F'(x^*; x^k - x^*) = o(\|x^k - x^*\|). \tag{3.3}$$

From Lemma 2.1 and (3.3) we have

$$\begin{aligned} \|x^{k+1} - x^*\| &= \|x^k - x^* - V_k^{-1} F(x^k)\| \\ &\le \|V_k^{-1}[F(x^k) - F(x^*) - F'(x^*; x^k - x^*)]\| \\ &\qquad + \|V_k^{-1}[V_k(x^k - x^*) - F'(x^*; x^k - x^*)]\| \\ &= o(\|x^k - x^*\|). \end{aligned}$$ □

From the theoretical point of view, there is no need to allow Newton matrices in $\partial_b F(\cdot)$ only since, due to the semismoothness assumptions, even each matrix of conv $\partial_b F(\cdot)$ could be used. The latter would lead to more general statements than those in Theorem 3.1. On the other hand, from the computational point of view, the assumption that all matrices $V \in$ conv $\partial_b F(x)$ are nonsingular is too strong and not necessary. So here we only restrict $V \in \partial_b F(x)$ and will not discuss the more general case that $V \in$ conv $\partial_b F(x)$. See [20] and section 6 for further discussions on the nonsingularity assumption of $V \in \partial_b F(x)$. For general statements on Newton methods for nonsmooth equations, see Qi and Sun [22] and Kummer [13].

4. Quasi-Newton method for nonsmooth equations and its specializations. In this section, we will first consider a quasi-Newton method for general nonsmooth equations and then discuss its specializations to a class of nonsmooth equations and related problems.

Consider the following quasi-Newton method:

$$x^{k+1} = x^k - V_k^{-1}F(x^k), \quad V_k \in R^{n\times n}, \ k = 0, 1, \ldots . \tag{4.1}$$

THEOREM 4.1. *Suppose that $F : R^n \to R^n$ is a locally Lipschitzian function in the open convex set $D \subset R^n$ and $x^* \in D$ is a solution of $F(x) = 0$. Suppose that F is semismooth at x^* and all $W_* \in \partial_b F(x^*)$ are nonsingular. There exist positive constants ε, Δ such that if $x^0 \in D$, $\|x^0 - x^*\| \le \varepsilon$, and there exists $W_k \in \partial_b F(x^k)$ such that*

$$\|V_k - W_k\| \le \Delta, \tag{4.2}$$

then the sequence of points generated by (4.1) *is well defined and converges to x^* Q-linearly in a neighborhood of x^*.*

Proof. From Lemma 2.2, there exists a positive constant β such that $\|W_*^{-1}\| \le \beta$ for all $W_* \in \partial_b F(x^*)$ and there exists a neighborhood $N_0(x^*)$ ($\subseteq D$) of x^* such that

$$\|W^{-1}\| \le \frac{10}{9}\beta$$

for any $y \in N_0(x^*)$, $W \in \partial_b F(y)$. Choose $\Delta > 0$ such that

$$6\beta\Delta \le 1. \tag{4.3}$$

Recall that a map is semismooth at x^* if and only if each of its components is semismooth at x^*. So from (1) and (2) of Lemma 2.1, for any $W^i \in \partial_b F_i(x)$, $x \to x^*$,

$$\|F_i(x) - F_i(x^*) - W^i(x - x^*)\| = o(\|x - x^*\|). \tag{4.4}$$

Therefore, for any $W \in \partial_b F(x)$, $x \to x^*$, we have

$$\|F(x) - F(x^*) - W(x - x^*)\| = o(\|x - x^*\|). \tag{4.5}$$

Then we can choose a positive constant ε small enough such that for any $x \in N(x^*) = \{y | \|y - x^*\| \le \varepsilon\} \subseteq N_0(x^*)$, $W \in \partial_b F(x)$, we have

$$\|F(x) - F(x^*) - W(x - x^*)\| \le \Delta\|x - x^*\|. \tag{4.6}$$

If $\|x^k - x^*\| \le \varepsilon$, then $W_k \in \partial_b F(x^k)$ is nonsingular and $\|W_k^{-1}\| \le \frac{10}{9}\beta$. By Theorem 2.3.2 of Ortega and Rheinboldt [15], V_k is invertible and

$$\|V_k^{-1}\| \le \frac{\|W_k^{-1}\|}{1 - \|W_k^{-1}(W_k - V_k)\|} \le \frac{\frac{10}{9}\beta}{1 - \frac{5}{27}} < \frac{3}{2}\beta. \tag{4.7}$$

Then when $\|x^k - x^*\| \le \varepsilon$, we have

$$\begin{aligned} \|x^{k+1} - x^k\| &= \|x^k - V_k^{-1}F(x^k) - x^*\| \\ &\le \|V_k^{-1}\|\|F(x^k) - F(x^*) - V_k(x^k - x^*)\| \\ &\le \|V_k^{-1}\|[\|F(x^k) - F(x^*) - W_k(x^k - x^*)\| \\ &\qquad + \|V_k - W_k\|\|x^k - x^*\|]. \end{aligned} \tag{4.8}$$

Substituting (4.2), (4.6), and (4.7) into (4.8) gives

$$\begin{aligned}\|x^{k+1}-x^*\| &\le \frac{3}{2}\beta[\Delta\|x^k-x^*\|+\Delta\|x^k-x^*\|]\\ &\le 3\beta\Delta\|x^k-x^*\|\\ &\le \frac{1}{2}\|x^k-x^*\|. \end{aligned}\tag{4.9}$$

This shows that the sequence of points generated by (4.1) is well defined and converges to x^* Q-linearly in a neighborhood of x^*. □

In [20], Pang and Qi extended Theorem 2.2 in Dennis and Moré [5] to nonsmooth equations. Here, we can do a similar extension and point out that some quasi-Newton methods belong to our frame form.

THEOREM 4.2. *Suppose that $F: R^n \to R^n$ is a locally Lipschitzian function in the open convex set $D \subset R^n$. Assume that F is semismooth at some $x^* \in D$ and all $W_* \in \partial_b F(x^*)$ are nonsingular. Let $\{V_k\}$ be a sequence of nonsingular matrices in $R^{n\times n}$, and suppose for some x^0 in D that the sequence of points generated by* (4.1) *remains in D and satisfies $x^k \ne x^*$ for all k, and $\lim_{k\to\infty} x^k = x^*$. Then $\{x^k\}$ converges Q-superlinearly to x^*, and $F(x^*)=0$ if and only if there exists $W_k \in \partial_b F(x^k)$ such that*

$$\lim_{k\to\infty}\frac{\|(V_k-W_k)s^k\|}{\|s^k\|}=0,\tag{4.10}$$

where $s^k = x^{k+1}-x^k$.

Proof. Write $e^k = x^k - x^*$. Then both sequence $\{e^k\}$ and $\{s^k\}$ converge to zero. From (4.1) we have

$$\begin{aligned}F(x^*) &= [F(x^k)+W_k s^k]-[F(x^k)-F(x^*)-W_k e^k]-W_k e^{k+1}\\ &= [F(x^k)+V_k s^k]+[(V_k-W_k)s^k]\\ &\quad -[F(x^k)-F(x^*)-W_k e^k]-W_k e^{k+1}\\ &= [(V_k-W_k)s^k]-[F(x^k)-F(x^*)-W_k e^k]-W_k e^{k+1}.\end{aligned}\tag{4.11}$$

From the semismoothness of F at x^* and (4.5) we know that the term in the second square bracket approaches zero as $k \to \infty$. So if (4.10) holds, then $H(x^*)=0$. From Lemma 2.2, $\{\|W_k^{-1}\|\}$ is bounded. Thus, from (4.5), (4.10), (4.11), and the boundedness of $\{\|W_k^{-1}\|\}$, we have

$$\|e^{k+1}\| \le o(\|s^k\|)+o(\|e^k\|) \le o(\|e^k\|)+o(\|e^{k+1}\|),$$

which means that

$$\lim_{k\to\infty}\frac{\|e^{k+1}\|}{\|e^k\|}=0.$$

Conversely, suppose that $H(x^*)=0$ and $\{x^k\}$ converges Q-superlinearly to x^*. Then reversing the above discussion easily establishes condition (4.10). □

As applications to Theorems 4.1 and 4.2, we will first consider the following nonsmooth equations, which arise from complementarity problems, special variational inequality problems, and the KKT system of nonlinear programming:

$$F(x) = x - P_X[x - f(x)] = 0, \tag{4.12}$$

where $f : R^n \to R^n$ is a continuously differentiable function, $P_Y(\cdot)$ is the orthogonal projection operator onto a nonempty closed convex set Y, and $X = \{x \in R^n |\ l \le x \le u\}$, where $l, u \in \{R \cup \{\infty\}\}^n$. To solve equation (4.12) is the original motivation in investigating nonsmooth equations. When $f \in C^1$, F is a semismooth function. The results of the Newton method for solving (4.12) are fruitful, but not for the quasi-Newton method. In this section, we will give a new quasi-Newton method for solving equation (4.12).

Quasi-Newton method (Broyden's case [1]).

```
Given $f : R^n \to R^n,\ x^0 \in R^n,\ A_0 \in R^{n\times n}$
Do for  $k = 0, 1, \dots :$
  Define
      $f^k(x) = f(x^k) + A_k(x - x^k)$
(4.13)   $F^k(x) = x - P_X[x - f^k(x)]$
  Choose  $V_k \in \partial_b F^k(x^k)$
  Solve  $V_k s^k + F(x^k) = 0$ for $s^k$
      $x^{k+1} = x^k + s^k$
      $y^k = f(x^{k+1}) - f(x^k)$
(4.14)   $A_{k+1} = A_k + \frac{(y^k - A_k s^k){s^k}^T}{{s^k}^T s^k}.$
```

For any matrix $B \in R^{n\times n}$, let B^i be the ith row of B. For an arbitrary function $f \in C^1$, if $V \in \partial_b F(x)$, then V satisfies

$$V^i = \begin{cases} I^i & \text{if } x_i - f_i(x) < l_i \text{ (or } > u_i), \\ \lambda_i I^i + (1-\lambda_i) f_i'(x) & \text{if } x_i - f_i(x) = l_i \text{ (or } = u_i), \\ f_i'(x) & \text{if } l_i < x_i - f_i(x) < u_i, \end{cases} \tag{4.15}$$

where $\lambda_i \in \{0, 1\}$ and I is the unit matrix of $R^{n\times n}$. On the other hand, any V of the above form is an element of $\partial_b F(x)$.

Corollary 4.1. *Suppose that $f : R^n \to R^n$ is continuously differentiable, x^* is a solution of* (4.12)*, $f'(x)$ is Lipschitz continuous in a neighborhood of x^*, and the Lipschitz constant is γ. Suppose that all $W_* \in \partial_b F(x^*)$ are nonsingular. There exist positive constants ε, δ such that if $\|x^0 - x^*\| \le \varepsilon$ and $\|A_0 - f'(x^*)\| \le \delta$, then the* *sequence $\{x^k\}$ generated by the quasi-Newton method (Broyden's case) is well defined and converges Q-superlinearly to x^*.*

Proof. First we prove the Q-linear convergence of $\{x^k\}$. Choose ε and Δ as in the proof of Theorem 4.1 and restrict ε to be small enough such that for any $y \in N(x^*) = \{x | \|x - x^*\| \leq \varepsilon\}$, we have

$$\|f'(y) - f'(x^*)\| \leq \gamma \|y - x^*\|, \tag{4.16}$$

$$3\gamma\varepsilon \leq \Delta. \tag{4.17}$$

Denote $\delta := \Delta/2$. From the definition of $F^k(x)$ and (4.15), the jth row V_k^j of V_k satisfies

$$V_k^j = \begin{cases} I^j & \text{if } \ x_j^k - f_j^k(x^k) < l_j \ (\text{or} > u_j), \\ \lambda_j^k I^j + (1-\lambda_j^k) A_k^j & \text{if } \ x_j^k - f_j^k(x^k) = l_j \ (\text{or} = u_j), \\ A_k^j & \text{if } \ l_j < x_j^k - f_j^k(x^k) < u_j, \end{cases} \tag{4.18}$$

where $\lambda_j^k \in \{0, 1\}$. For such constants λ_j^k we define a companion matrix W_k such that the jth row W_k^j of W_k satisfies

$$W_k^j = \begin{cases} I^j & \text{if } \ x_j^k - f_j^k(x^k) < l_j \ (\text{or} > u_j), \\ \lambda_j^k I^j + (1-\lambda_j^k) f_j'(x^k) & \text{if } \ x_j^k - f_j^k(x^k) = l_j \ (\text{or} = u_j), \\ f_j'(x^k) & \text{if } \ l_j < x_j^k - f_j^k(x^k) < u_j. \end{cases} \tag{4.19}$$

From $f(x^k) = f^k(x^k)$ and (4.19) we get

$$W_k \in \partial_b F(x^k).$$

From (4.18) and (4.19) for any $x \in R^n$ we get

$$|(W_k^j - V_k^j)x| \leq |(A_k^j - f_j'(x^k))x|,$$

which means that

$$\|(W_k - V_k)x\| \leq \|(A_k - f'(x^k))x\|. \tag{4.20}$$

Thus,

$$\begin{aligned} \|W_k - V_k\| &\leq \|A_k - f'(x^k)\| \\ &\leq \|A_k - f'(x^*)\| + \|f'(x^k) - f'(x^*)\|. \end{aligned} \tag{4.21}$$

The local Q-linear convergence proof consists of showing by induction that

$$\|A_k - f'(x^*)\| \leq (2 - 2^{-k})\delta, \tag{4.22}$$

$$\|V_k - W_k\| \leq \Delta. \tag{4.23}$$

For $k = 0$, (4.22) is trivially true. The proof of (4.23) is identical to the proof at the induction step, so we omit it here.

Now assume that (4.22) and (4.23) hold for $k = 0, 1, \ldots, i-1$. From the proof of Theorem 4.1, for $k = 0, 1, \ldots, i-1$, we have

$$\|e^{k+1}\| \le \frac{1}{2}\|e^k\|. \tag{4.24}$$

For $k = i$, we have from Lemma 8.2.1 of [6] (also see [5]), (4.24), and the induction hypothesis that

$$\begin{aligned} \|A_i - f'(x^*)\| &\le \|A_{i-1} - f'(x^*)\| + \frac{\gamma}{2}(\|e^i\| + \|e^{i-1}\|) \\ &\le (2 - 2^{-(i-1)})\delta + \frac{3\gamma}{4}\|e^{i-1}\|. \end{aligned} \tag{4.25}$$

From (4.24) and $\|e^0\| \le \varepsilon$ we get

$$\|e^{i-1}\| \le 2^{-(i-1)}\|e^0\| \le 2^{-(i-1)}\varepsilon.$$

Substituting this into (4.25) and using (4.17) gives

$$\begin{aligned} \|A_i - f'(x^*)\| &\le (2 - 2^{-(i-1)})\delta + \frac{3\gamma}{4}\varepsilon \cdot 2^{-(i-1)} \\ &\le (2 - 2^{-(i-1)} + 2^{-i})\delta = (2 - 2^{-i})\delta, \end{aligned}$$

which verifies (4.22).

To complete the induction, we verify (4.23). Substituting (4.22) into (4.21) for $k = i$ and using $\|e^0\| \le \varepsilon$, (4.16), (4.17), and (4.24) gives

$$\begin{aligned} \|W_i - V_i\| &\le (2 - 2^{-i})\delta + 2^{-i}\varepsilon\gamma \\ &= (2 - 2^{-i})\frac{\Delta}{2} + \frac{1}{3} \cdot 2^{-i}\Delta \\ &< \Delta. \end{aligned}$$

This proves (4.23). So the Q-linear convergence follows from Theorem 4.1.

Next we will prove the Q-superlinear convergence of $\{x^k\}$ under the assumptions. Let $E_k = A_k - f'(x^*)$. From the last part of the proof of Theorem 8.2.2 of [6] (also see [5]) we get

$$\lim_{k\to\infty} \frac{\|E_k s^k\|}{\|s^k\|} = 0. \tag{4.26}$$

From (4.20) and (4.16), we have

$$\begin{aligned} \|(V_k - W_k)s^k\| &\le \|(A_k - f'(x^k))s^k\| \\ &\le \|(A_k - f'(x^*))s^k\| + \|(f'(x^k) - f'(x^*))s^k\| \\ &\le \|E_k s^k\| + \gamma\|e^k\|\|s^k\|. \end{aligned} \tag{4.27}$$

Substituting (4.26) into (4.27) and using the linear convergence of $\{x^k\}$ gives

$$\lim_{k\to\infty}\frac{\|(V_k-W_k)s^k\|}{\|s^k\|}=0,$$

which, from Theorem 4.2, means that $\{x^k\}$ converges to x^* Q-superlinearly. □

Recall that when X is the nonnegative orthant, i.e., $X=R^n_+$, $F(x)$ defined by (4.12) is essentially equivalent to the function $H(x)$ in [9] and [17]. In [9], Ip and Kyparisis discussed the convergence properties of quasi-Newton methods directly applied to nonsmooth equations. For nonlinear complementarity problems, they described the sufficient conditions to guarantee the convergence of the quasi-Newton method (see Theorem 5.2 of [9]). A restrictive assumption in [9] is that F is strongly F-differentiable at x^*. This condition, which restricts the class f to which Theorem 5.2 of [9] applies, is satisfied if $f_i'(x^*)=I^i$ for all $i\in\{j|f_j(x^*)=x^*_j,\ j=1,\dots,n\}$. Here, to guarantee the convergence of our new quasi-Newton method, we need the nonsingularity of $\partial_b F(x^*)$ instead of needing the existence and invertibility of $F'(x^*)$. For nonlinear complementarity problems, the nonsingularity assumption of $\partial_b F(x^*)$ is equivalent to the b-regularity assumption in [19]. For a detailed discussion on b-regularity, see [19].

Next we consider the following nonsmooth equation:

$$F(x)=\min(f(x),g(x))=0, \tag{4.28}$$

where $f,\ g: R^n\to R^n$ are continuously differentiable and the "min" operator denotes the componentwise minimum of two vectors. Such a system arises from nonsmooth partial differentiable equations [3, 2, 15] and implicit complementarity problems (see, e.g., [16]). When $g(x)=x$, (4.28) is the function $H(x)$ discussed in [9] and [17] and is equivalent to (4.12) for $X=R^n_+$. Here we will give a new quasi-Newton method (Broyden's case) for solving (4.28). In particular, the new resulting method with $g(x)=x$ coincides with the quasi-Newton method for solving (4.12) with $X=R^n_+$. In both methods, the concept $\partial_b F(\cdot)$ has an important role.

Quasi-Newton method (Broyden's case [1]).

```
Given  $x^0 \in R^n,\ A_0,\ B_0 \in R^{n\times n}$
Do for  $k = 0, 1, \dots$ :
  Define
      $f^k(x) = f(x^k) + A_k(x - x^k)$
      $g^k(x) = g(x^k) + B_k(x - x^k)$
      $F^k(x) = \min(f^k(x), g^k(x))$
  Choose  $V_k \in \partial_b F^k(x^k)$
  Solve  $V_k s^k + F(x^k) = 0$ for $s^k$
      $x^{k+1} = x^k + s^k$
      $y^k = f(x^{k+1}) - f(x^k)$
```

$$z^k = g(x^{k+1}) - g(x^k)$$

$$A_{k+1} = A_k + \frac{(y^k - A_k s^k) s^{k^T}}{s^{k^T} s^k}$$

$$B_{k+1} = B_k + \frac{(z^k - B_k s^k) s^{k^T}}{s^{k^T} s^k}.$$

COROLLARY 4.2. *Suppose that $f,\ g : R^n \to R^n$ are continuously differentiable, x^* is a solution of* (4.28)*, $f'(x)$, $g'(x)$ are Lipschitz continuous in a neighborhood of x^*, and the common Lipschitz constant is γ. Suppose that all $W_* \in \partial_b F(x^*)$ are nonsingular. There exist positive constants ε, δ such that if $\|x^0 - x^*\| \le \varepsilon$, $\|A_0 - f'(x^*)\| \le \delta$, and $\|B_0 - g'(x^*)\| \le \delta$, then the sequence $\{x^k\}$ generated by the quasi-Newton method (Broyden's case) is well defined and converges Q-superlinearly to x^*.*

Proof. The proof is similar to that of Corollary 4.1. Here we only give an outline of the proof. It is not difficult to give the detail.

Choose ε and Δ as in the proof of Theorem 4.1 and restrict ε to be small enough such that for any $y \in N(x^*) = \{x | \|x - x^*\| \le \varepsilon\}$, we have

$$\|f'(y) - f'(x^*)\| \le \gamma \|y - x^*\|,\ \|g'(y) - g'(x^*)\| \le \gamma \|y - x^*\|, \tag{4.29}$$

$$6\gamma\varepsilon \le \Delta. \tag{4.30}$$

Denote $\delta := \Delta/4$. From the definition of $F^k(x)$ there exists $\lambda_j^k \in \{0, 1\}$ such that the jth row V_k^j of V_k satisfies

$$V_k^j = \begin{cases} A_k^j & \text{if } f_j^k(x^k) < g_j^k(x^k), \\ \lambda_j^k A_k^j + (1 - \lambda_j^k) B_k^j & \text{if } f_j^k(x^k) = g_j^k(x^k), \\ B_k^j & \text{if } f_j^k(x^k) > g_j^k(x^k). \end{cases} \tag{4.31}$$

For such constants λ_j^k we define a companion matrix W_k such that the jth row W_k^j of W_k satisfies

$$W_k^j = \begin{cases} f_j'(x^k) & \text{if } f_j^k(x^k) < g_j^k(x^k), \\ \lambda_j^k f_j'(x^k) + (1 - \lambda_j^k) g_j'(x^k) & \text{if } f_j^k(x^k) = g_j^k(x^k), \\ g_j'(x^k) & \text{if } f_j^k(x^k) > g_j^k(x^k). \end{cases} \tag{4.32}$$

From $f(x^k) = f^k(x^k)$, $g(x^k) = g^k(x^k)$, and the definition of $\partial_b F(x^k)$, we get

$$W_k \in \partial_b F(x^k).$$

From (4.31) and (4.32), for any $x \in R^n$ we get

$$\|(V_k - W_k)x\| \le \|(A_k - f'(x^k))x\| + \|(B_k - g'(x^k))x\|. \tag{4.33}$$

Thus,

$$
\begin{aligned}
\|W_k - V_k\| &\le \|A_k - f'(x^k)\| + \|B_k - g'(x^k)\| \\
&\le \|A_k - f'(x^*)\| + \|f'(x^k) - f'(x^*)\| \\
&\quad + \|B_k - g'(x^*)\| + \|g'(x^k) - g'(x^*)\|.
\end{aligned} \tag{4.34}
$$

The local Q-linear convergence proof consists of showing by induction that

$$\|A_k - f'(x^*)\| \le (2-2^{-k})\delta, \ \|B_k - g'(x^*)\| \le (2-2^{-k})\delta,$$

$$\|V_k - W_k\| \le \Delta.$$

The induction proof is similar to that of Corollary 4.1. We omit it here.

To prove the Q-superlinear convergence of $\{x^k\}$, let $E_k = A_k - f'(x^*)$ and $H_k = B_k - g'(x^*)$. From the last part of the proof of Theorem 8.2.2 of [6] (also see [5]) we get

$$\lim_{k\to\infty} \frac{\|E_k s^k\|}{\|s^k\|} = 0, \quad \lim_{k\to\infty} \frac{\|H_k s^k\|}{\|s^k\|} = 0. \tag{4.35}$$

From (4.33) and (4.29), we have

$$
\begin{aligned}
\|(V_k - W_k)s^k\| &\le \|(A_k - f'(x^k))s^k\| + \|(B_k - g'(x^k))s^k\| \\
&\le \|E_k s^k\| + \gamma\|e^k\|\|s^k\| + \|H_k s^k\| + \gamma\|e^k\|\|s^k\|.
\end{aligned} \tag{4.36}
$$

Thus, from (4.35), (4.36), and the linear convergence of $\{x^k\}$, we get

$$\lim_{k\to\infty} \frac{\|(V_k - W_k)s^k\|}{\|s^k\|} = 0,$$

which, from Theorem 4.2, means that $\{x^k\}$ converges to x^* Q-superlinearly. □

In [21], Qi discussed a Newton method for solving (4.28) and provided a method to compute $\partial_B F$. Here, by using the concept $\partial_b F$, we give a quasi-Newton method. The main condition to guarantee the local Q-superlinear convergence is the nonsingularity assumption of $\partial_b F(x^*)$. When $g(x) = x$, this nonsingularity assumption is exactly the b-regularity in [19].

5. Implementation of the quasi-Newton method. The implementation of the quasi-Newton method discussed in section 4 for solving equation (4.12) has no difference to the smooth case except for the implementation of the QR factorization of the iterate matrix V_k. The entire QR factorization of V_k costs $O(n^3)$ arithmetic operations. If we do this in every step, then the advantage of quasi-Newton method loses a lot. In this section, we will show how to update the QR factorization of V_k into the QR factorization of V_{k+1} at most in $O((I(k)+1)n^2)$ operations (see (5.8) for the definition of $I(k)$). For simplicity, we will assume that $X = R^n_+$.

For a given vector $x \in R^n$, denote the index sets

$$\alpha(x) = \{i : x_i > f_i(x)\},$$

$$\beta(x) = \{i : x_i = f_i(x)\},$$

$$\gamma(x) = \{i : x_i < f_i(x)\}.$$

Suppose for each k that we choose $V_k \in \partial_b F^k(x^k)$ such that the ith row V_k^i of V_k satisfies

$$V_k^i = \begin{cases} A_k^i & \text{if} \quad i \in \alpha(x^k), \\ I^i & \text{if} \quad i \in \beta(x^k) \cup \gamma(x^k). \end{cases} \tag{5.1}$$

Denote a matrix $\overline{V}_k$ such that its ith row $\overline{V}_k^i$ satisfies

$$\overline{V}_k^i = \begin{cases} A_{k+1}^i & \text{if} \quad i \in \alpha(x^k), \\ I^i & \text{if} \quad i \in \beta(x^k) \cup \gamma(x^k). \end{cases} \tag{5.2}$$

From (5.1), (5.2), and (4.14), we get

$$\overline{V}_k = V_k + \frac{(\overline{y}^k - V_k s^k){s^k}^T}{{s^k}^T s^k}, \tag{5.3}$$

where $\overline{y}^k$ satisfies

$$\overline{y}_i^k = \begin{cases} y_i^k & \text{if} \quad i \in \alpha(x^k), \\ s_i^k & \text{if} \quad i \in \beta(x^k) \cup \gamma(x^k). \end{cases} \tag{5.4}$$

It is well known that we can update the QR factorization of V_k into the QR factorization of $\overline{V}_k$ in $O(n^2)$ operations (see, e.g., [7, 8]).

The ith row V_{k+1}^i of V_{k+1} satisfies

$$V_{k+1}^i = \begin{cases} A_{k+1}^i & \text{if} \quad i \in \alpha(x^{k+1}), \\ I^i & \text{if} \quad i \in \beta(x^{k+1}) \cup \gamma(x^{k+1}). \end{cases} \tag{5.5}$$

Therefore,

$$V_{k+1} = \overline{V}_k + \Delta\overline{V}_k, \tag{5.6}$$

where $\Delta\overline{V}_k$ satisfies

$$\Delta\overline{V}_k^i = \begin{cases} 0 & \text{if} \quad i \in \alpha(x^k) \cap \alpha(x^{k+1}), \\ 0 & \text{if} \quad i \in \{\beta(x^k) \cup \gamma(x^k)\} \cap \{\beta(x^{k+1}) \cup \gamma(x^{k+1})\}, \\ V_{k+1}^i - \overline{V}_k^i & \text{otherwise.} \end{cases} \tag{5.7}$$

Denote

$$I(k) = n - (|\alpha(x^k) \cap \alpha(x^{k+1})| + |\{\beta(x^k) \cup \gamma(x^k)\} \cap \{\beta(x^{k+1}) \cup \gamma(x^{k+1})\}|). \tag{5.8}$$

Since the number of the nonzero rows of $\Delta\overline{V}_k$ is at most $I(k)$, we can update the QR factorization of $\overline{V}_k$ into the QR factorization of V_{k+1} at most in $O(I(k)n^2)$ operations (see, e.g., [7, 8]).

Therefore, we get the following theorem.

THEOREM 5.1. *The cost of updating the QR factorization of V_k into the QR factorization of V_{k+1} is at most $O((I(k)+1)n^2)$ arithmetic operations.*

Josephy [10] considered the quasi-Newton method for solving generalized equations (see Robinson [24]). For nonlinear complementarity problems, in every step his method needs to solve a linear complementarity problems, which requires more cost than solving a linear equation. Kojima and Shindo [11] extended the quasi-Newton method to piecewise smooth equations. They applied the classical Broyden's method as the points x^k stayed within a given C^1-piece. When the points x^k arrived at a new piece, a new starting matrix was used and it was needed to perform the entire QR factorization (or other factorizations) in $O(n^3)$ operations in general. Thus a potentially large number of matrices need to be stored and need to be performed to get an entire QR factorization (or other factorizations). Here, our method needs only one approximate matrix, and except for the first step we only need less effort to solve a linear equation, which may be solved in much less than $O(n^3)$ operations. The smaller the measure of $I(k)$ is, the less computing effort is needed in the $(k+1)$th step (note that $I(k)$ is related to the nonsmoothness of F). Ip and Kyparisis [9] discussed the local convergence of the classical Broyden's quasi-Newton method for solving nonsmooth equations. Although the form used in [9] is very simple, the convergence remains open without assuming the existence of $F'(x^*)$.

6. The KKT system of variational inequality problems. For a given closed set $X \subseteq R^n$ and a mapping $f : X \to R^n$, the variational inequality problem which is denoted by VI(X, f) is to find a vector $x^* \in X$ such that

$$(x - x^*)^T f(x^*) \geq 0 \quad \text{for all } x \in X.$$

If $X = R^n_+$, then VI(X, f) is equivalent to the complementarity problem which is to find $x^* \in R^n_+$ such that

$$f(x^*) \in R^n_+ \text{ and } {x^*}^T f(x^*) = 0.$$

When f is a gradient mapping, say $f(x) = \nabla\theta(x)$ for some real-valued function θ, VI(X, f) is equivalent to the problem of finding a stationary point for the following minimization problem:

$$\text{minimize } \theta(x)$$

$$\text{subject to } x \in X.$$

Here we shall assume that X has the form

$$X = \{x \in R^n |\ g(x) \leq 0,\ h(x) = 0,\ l \leq x \leq u\}, \tag{6.1}$$

where $g : R^n \to R^m$ and $h : R^n \to R^p$ are assumed to be twice continuously differentiable, and $l, u \in \{R \cup \{\infty\}\}^n$. By introducing multipliers $(\lambda, \mu, v, w) \in R^{m+p+2n}$ corresponding to the constraints in X, the (VI) Lagrangian (vector-valued) function (see, e.g., Tobin [29]) can be defined by

$$L(x, \lambda, \mu, v, w) = f(x) + \sum_{i=1}^{m} \nabla g_i(x)\lambda_i + \sum_{j=1}^{p} \nabla h_j(x)\mu_j - v + w.$$

If $l_i = -\infty$ (or $u_i = +\infty$) for some i, the corresponding v_i (w_i, respectively) is absent in the above formula. Then the KKT system of VI(X, f) can be written as

$$\begin{cases} L(x, \lambda, \mu, v, w) = 0, \\ \lambda \geq 0, \ -g(x) \geq 0, \text{ and } \lambda^T g(x) = 0, \\ -h(x) = 0, \\ v \geq 0, \ x - l \geq 0, \text{ and } v^T(x - l) = 0, \\ w \geq 0, \ u - x \geq 0, \text{ and } w^T(x - u) = 0. \end{cases} \tag{6.2}$$

Define

$$\tilde{L}(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \nabla g_i(x)\lambda_i + \sum_{j=1}^{p} \nabla h_j(x)\mu_j$$

and

$$H(x, \lambda, \mu) = \begin{pmatrix} x - P_{[l,u]}[x - \tilde{L}(x, \lambda, \mu)] \\ \lambda - P_{R_+^n}[\lambda - (-g(x))] \\ -h(x) \end{pmatrix}. \tag{6.3}$$

Suppose that $(x^*, \lambda^*, \mu^*, v^*, w^*) \in R^{n+m+p+2n}$ is a solution of the KKT system (6.2), then (x^*, λ^*, μ^*) satisfies $H(x^*, \lambda^*, \mu^*) = 0$; conversely, if $(x^*, \lambda^*, \mu^*) \in R^{n+m+p}$ is a solution of $H(x, \lambda, \mu) = 0$, then $(x^*, \lambda^*, \mu^*, v^*, w^*)$ is a solution of the KKT system (6.2), where v^*, w^* are defined as

$$v^* = P_{R_+^n}[\tilde{L}(x^*, \lambda^*, \mu^*)] \text{ and } w^* = P_{R_+^n}[-\tilde{L}(x^*, \lambda^*, \mu^*)]. \tag{6.4}$$

So finding a solution of the KKT system of VI is equivalent to solving $H(x, \lambda, \mu) = 0$. Let $z = (x, \lambda, \mu)$, $K = [l, u] \times R_+^n \times R^p$, and

$$\tilde{f}(z) = \begin{pmatrix} \tilde{L}(z) \\ -g(x) \\ -h(x) \end{pmatrix}.$$

Then $H(x, \lambda, \mu) = 0$ can be written as

$$H(z) = z - P_K[z - \tilde{f}(z)] = 0, \tag{6.5}$$

which is a special form of (4.12).

Now suppose that z^* is a solution of $H(z) = 0$ and f is continuously differentiable at x^*; we will discuss a sufficient condition on the nonsingularity assumption of $\partial_b H(z^*)$. Let

$$I(z^*) = \{i | \ 1 \leq i \leq m, \ g_i(x^*) = 0\},$$

$$I^+(z^*) = \{i \in I(z^*) |\ \lambda_i^* > 0\},$$

$$G^+(z^*) = \{d \in R^n | \quad \nabla g_i(x^*)^T d = 0 \text{ for } i \in I^+(z^*)$$

$$\text{and } \nabla h_i(x^*)^T d = 0 \text{ for } i = 1, \ldots, p\},$$

and

$$R(z^*) = \{d \in R^n |\ d_i = 0 \text{ if } x_i^* = l_i \text{ (or } u_i) \text{ and } (\tilde{L}(z^*))_i \neq 0 \text{ for } i = 1, \ldots, n\}.$$

THEOREM 6.1. *Suppose that z^* is a solution of $H(z) = 0$ and that it satisfies $d^T \nabla_{xx}^2 \tilde{L}(z^*) d > 0$ for all $d \in G^+(z^*) \cap R(z^*) \backslash \{0\}$. If $\{\nabla g_i(x^*),\ i \in I(z^*)\}$ and $\{\nabla h_i(x^*),\ i = 1, \ldots, p\}$ are linearly independent, then all $V \in \partial_b H(z^*)$ are nonsingular.*

Proof. Combining (4.15) and the proof of Theorem 4.1 in Robinson [24], we can get the result. □

7. Numerical examples. In this section, we report computational results obtained for two small nonlinear complementarity problems using the above Newton method and quasi-Newton method. For the quasi-Newton method, the initial matrices are generated by the difference approximation method. In Table 1, "N" and "QN" represent the Newton method and quasi-Newton method, respectively, and "P 1" and "P 2" represent Problem 1 and Problem 2, respectively.

Problem 1 (a nondegenerate nonlinear complementarity problem [10, 9]). Consider the following problem: find $x \in R^4$ such that $x \geq 0,\ f(x) \geq 0$, and $x^T f(x) = 0$, where $f : R^4 \to R^4$ is given by

$$f_1(x) = 3x_1^2 + 2x_1x_2 + 2x_2^2 + x_3 + 3x_4 - 6,$$

$$f_2(x) = 2x_1^2 + x_1 + x_2^2 + 3x_3 + 2x_4 - 2,$$

$$f_3(x) = 3x_1^2 + x_1x_2 + 2x_2^2 + 2x_3 + 3x_4 - 1,$$

$$f_4(x) = x_1^2 + 3x_2^2 + 2x_3 + 3x_4 - 3.$$

This problem has the solution

$$x^* = \left(\frac{1}{2}\sqrt{6} \approx 1.2247, 0, 0, 0.5\right), \quad f(x^*) = \left(0, 2 + \frac{1}{2}\sqrt{6} \approx 3.2247, 5, 0\right).$$

Since $\beta(x^*) = \emptyset,\ \ x^*$ is nondegenerate (see [9]) and it is easy to check that $F'(x^*)$ (here $\partial_b F'(x^*) = \{F'(x^*)\}$) is nonsingular.

Problem 2 (a degenerate nonlinear complementarity problem [11, 9]). Consider the following problem: find $x \in R^4$ such that $x \geq 0,\ f(x) \geq 0$, and $x^T f(x) = 0$, where $f : R^4 \to R^4$ is given by

$$f_1(x) = 3x_1^2 + 2x_1x_2 + 2x_2^2 + x_3 + 3x_4 - 6,$$

$$f_2(x) = 2x_1^2 + x_1 + x_2^2 + 10x_3 + 2x_4 - 2,$$

$$f_3(x) = 3x_1^2 + x_1x_2 + 2x_2^2 + 2x_3 + 9x_4 - 9,$$

$$f_4(x) = x_1^2 + 3x_2^2 + 2x_3 + 3x_4 - 3.$$

TABLE 1
Results for Problems 1 *and* 2*, where* D = *degenerate solution and* ND = *nondegenerate solution.*

Algorithm	Starting point	Number of Iterations		sum of $I(k)$	
		P 1	P 2	P 1	P 2
N	(1,0,0,0)	3	3(D)		
QN	(1,0,0,0)	4	4(D)	0	2
N	(1,0,1,0)	4	1(ND)		
QN	(1,0,1,0)	5	1(ND)	1	0
N	(1,0,0,1)	4	4(D)		
QN	(1,0,0,1)	5	5(D)	1	2
N	(1,0.2,0.5,1)	4	4(D)		
QN	(1,0.2,0.5,1)	6	6(D)	0	2
N	(1,0,1,-1)	3	3(D)		
QN	(1,0,1,-1)	5	5(D)	1	2
N	(1.5,-0.5,4.5,-1.0)	4	4(D)		
QN	(1.5,-0.5,4.5,-1.0)	6	6(D)	1	0
N	(1.1,-0.1,3.1,-0.1)	4	3(ND)		
QN	(1.1,-0.1,3.1,-0.1)	5	4(ND)	1	0
N	(0.85,0.2,0.5,1)	4	5(D)		
QN	(0.85,0.2,0.5,1)	7	7(D)	1	2

This problem has the following two solutions:

$$x_D^* = \left(\frac{1}{2}\sqrt{6} \approx 1.2247, 0, 0, 0.5\right), \quad f(x_D^*) = \left(0, 2 + \frac{1}{2}\sqrt{6} \approx 3.2247, 0, 0\right),$$

and

$$x_{ND}^* = (1, 0, 3, 0), \quad f(x_{ND}^*) = (0, 31, 0, 4).$$

Since $\beta(x_{ND}^*) = \emptyset$ for the solution x_{ND}^*, it is a nondegenerate solution (see [9]). On the other hand, $\beta(x_D^*) = \{3\}$ for the solution x_D^*, so it is a degenerate solution (see [9]). It is easy to check that $\partial_b F(x_{ND}^*)$ and $\partial_b F(x_D^*)$ are nonsingular.

From Table 1 we see that even for Problem 2 when the starting point is close to a solution, the sequence will converge to the corresponding solution no matter whether it is degenerate or not.

In this paper two small examples are used to show the effectiveness of the Newton method and the quasi-Newton method for solving some nonsmooth equations. More examples are needed to show the efficiency of the above algorithms. For problem (4.12) with a general convex set X, especially when X is a polyhedral set, how to construct appropriate Newton methods and quasi-Newton methods is our further research topic.

Acknowledgments. We are grateful to two referees for their helpful suggestions and comments on this paper, to Prof. Jong-Shi Pang for his useful comments on an earlier version of this paper, and to Prof. Liqun Qi and Dr. Xiaojun Chen for their helpful suggestions on this subject. Finally, we thank Prof. Masakazu Kojima for his help in revising this paper.

REFERENCES

[1] C. G. BROYDEN, *A class of methods for solving nonlinear simultaneous equations*, Math. Comp., 19 (1965), pp. 577–593.

[2] X. CHEN AND T. YAMAMOTO, *On the convergence of some quasi-Newton methods for solving nonlinear equations with nondifferentiable operators*, Computing, 49 (1992), pp. 87–94.

[3] X. Chen and L. Qi, *A parameterized Newton method and a quasi-Newton method for solving nonsmooth equations*, Comput. Optim. Appl., 3 (1994), pp. 157–179.

[4] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley and Sons, New York, 1983.

[5] J. E. Dennis and J. J. Moré, *A characterization of superlinear convergence and its application to quasi-Newton methods*, Math. Comp., 28 (1974), pp. 549–560.

[6] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1983.

[7] P. E. Gill and and M. Murray, *Quasi-Newton methods for unconstrained optimization*, J. Inst. Math. Appl., 9 (1972), pp. 91–108.

[8] G. H. Golub and C. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1983.

[9] C.-M. Ip and T. Kyparisis, *Local convergence of quasi-Newton methods for B-differentiable equations*, Math. Programming, 56 (1992), pp. 71–89.

[10] N. H. Josephy, *Quasi-Newton Methods for Generalized Equations*, Technical summary report 1966, Mathematical Research Center, University of Wisconsin, Madison, WI, 1979.

[11] M. Kojima and S. Shindo, *Extensions of Newton and quasi-Newton methods to systems of PC^1 equations*, J. Oper. Res. Soc. Japan, 29 (1986), pp. 352–374.

[12] B. Kummer, *Newton's method for non-differentiable functions*, in Advances in Mathematical Programming, J. Guddat, B. Bank, H. Hollatz, P. Kall, D. Klatte, B. Kummer, K. Lommatzsch, L. Tammer, M. Vlach, and K. Zimmerman, eds., Academi Verlag, Berlin, 1988, pp. 114–125.

[13] B. Kummer, *Newton's method based on generalized derivatives for nonsmooth functions: Convergence analysis*, in Lecture Notes in Economics and Mathematical Systems 382: Advances in Optimization, W. Oettli and D. Pallaschke, eds., Springer, Berlin, 1992, pp. 171–194.

[14] M. Mifflin, *Semismooth and semiconvex functions in constrained optimization*, SIAM J. Control Optim., 15 (1977), pp. 957–972.

[15] J. M. Ortega and W. C. Rheinboldt, *Iterative solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[16] J.-S. Pang, *The implicit complementarity problem*, in Nonlinear Programming 4, O. L. Mangasarian, S. M. Robinson, and P. R. Meyer, eds., Academic Press, New York, 1981, pp. 487–518.

[17] J.-S. Pang, *Newton's method for B-differentiable equations*, Math. Oper. Res., 15 (1990), pp. 311–341.

[18] J.-S. Pang, *A B-differentiable equation-based, globally and locally quadratically convergent algorithm for nonlinear programs, complementarity and variational inequality problems*, Math. Programming, 51 (1991), pp. 101–131.

[19] J.-S. Pang and S. A. Gabriel, *NE/SQP: A robust algorithm for the nonlinear complementarity problem*, Math. Programming, 60 (1993), pp. 295–337.

[20] J.-S. Pang and L. Qi, *Nonsmooth equations: Motivation and algorithms*, SIAM J. Optim., 3 (1993), pp. 443–465.

[21] L. Qi, *Convergence analysis of some algorithms for solving nonsmooth equations*, Math. Oper. Res., 18 (1993), pp. 227–244.

[22] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Programming, 58 (1993), pp. 353–368.

[23] L. Qi and J. Sun, *A Nonsmooth Version of Newton's Method and an Interior Point Algorithm for Convex Programming*, Applied Mathematics Preprint 89/33, School of Mathematics, The University of New South Wales, Sydney, Australia, 1991.

[24] S. M. Robinson, *Strongly regular generalized equations*, Math. Oper. Res., 5 (1980), pp. 43–62.

[25] S. M. Robinson, *Local structure of feasible sets in nonlinear programming, Part III: Stability and sensitivity*, Math. Programming Study, 30 (1987), pp. 45–66.

[26] S. M. Robinson, *Newton's methods for a class of nonsmooth functions*, Set-Valued Analysis, 2 (1994), pp. 292–305.

[27] A. Shapiro, *On concepts of directional differentiability*, J. Optim. Theory Appl., 66 (1990), pp. 477–487.

[28] N. Z. Shor, *A class of almost-differentiable functions and a minimization method for functions of this class*, Kibernetic, 4 (1972), pp. 65–70.

[29] R. L. Tobin, *Sensitivity analysis for variational inequalities*, J. Optim. Theory Appl., 48 (1986), pp. 191–204.

SIAM J. Optim.
Vol. 7, No. 2, pp. 481–507, May 1997

EXACT PENALIZATION AND NECESSARY OPTIMALITY CONDITIONS FOR GENERALIZED BILEVEL PROGRAMMING PROBLEMS[*]

J. J. YE[†], D. L. ZHU[‡], AND Q. J. ZHU[§]

Abstract. The generalized bilevel programming problem (GBLP) is a bilevel mathematical program where the lower level is a variational inequality. In this paper we prove that if the objective function of a GBLP is uniformly Lipschitz continuous in the lower level decision variable with respect to the upper level decision variable, then using certain uniform parametric error bounds as penalty functions gives single level problems equivalent to the GBLP. Several local and global uniform parametric error bounds are presented, and assumptions guaranteeing that they apply are discussed. We then derive Kuhn–Tucker-type necessary optimality conditions by using exact penalty formulations and nonsmooth analysis.

Key words. generalized bilevel programming problems, variational inequalities, exact penalty formulations, uniform parametric error bounds, necessary optimality conditions, nonsmooth analysis

AMS subject classifications. 49K99, 90C, 90D65

PII. S1052623493257344

1. Introduction.

We consider the following *mathematical programming problem with variational inequality constraints* (which is called the *generalized bilevel programming problem* (GBLP)):

$$\text{(1)}\quad \text{GBLP} \qquad \text{minimize } f(x,y) \qquad \text{subject to } x \in X \text{ and } y \in S(x)$$

where $f : R^{n+m} \to R$, X is a nonempty and closed subset of R^n, and for each $x \in X$, $S(x)$ is the *solution set of a variational inequality with parameter* x,

$$S(x) = \{y \in U(x) : \langle F(x,y), y - z\rangle \leq 0 \ \ \forall z \in U(x)\}.$$

Here $U : X \to R^m$ is a set-valued map and $F : R^{n+m} \to R^m$ is a function. Throughout this paper, we make the blanket assumption that $\mathrm{Gr}S := \{(x,y) : x \in X, y \in S(x)\}$, the graph of S, is not empty.

One can interpret the above problem as a *hierarchical decision process* where there are two decision makers and the upper level decision maker always has the first choice as follows: given a decision vector x for the upper level decision maker (the *leader*), $S(x)$ is viewed as the lower level decision maker's (the *follower's*) decision set, i.e., the set of decision vectors that the follower may use. Assuming that the game is *cooperative* (i.e., the follower's decision set $S(x)$ is not a singleton), the follower allows the leader to choose the lower level decision from $S(x)$. Having complete knowledge of the follower's possible reactions, the leader selects decision vectors $x \in X$ and $y \in S(x)$, minimizing his objective function $f(x,y)$.

[*] Received by the editors October 18, 1993; accepted for publication (in revised form) December 15, 1995.
http://www.siam.org/journals/siopt/7-2/25734.html

[†] Department of Mathematics and Statistics, University of Victoria, Victoria, BC, Canada V8W 3P4 (janeye@uvic.ca). The research of this author was supported by National Science and Engineering Research Council of Canada grant WFA 0123160.

[‡] Center for Research on Transportation, University of Montréal, C.P. 6128, Succursale Centre-Ville, Montréal, Québec, Canada H3T 1V6 (daoli@crt.umontreal.ca).

[§] Department of Mathematics and Statistics, Western Michigan University, Kalamazoo, MI 49008-5152 (zhu@math-stat.wmich.edu).

If $F(x,y)$ is the partial gradient of a real-valued differentiable function (i.e., $F(x,y) = -\nabla_y g(x,y)$, where $g : R^{n+m} \to R$ is differentiable in y and $U(x)$ is convex), then the variational inequality with parameter x,

$$\langle F(x,y), y-z \rangle \leq 0 \qquad \forall z \in U(x), \tag{2}$$

is the first-order necessary optimality condition for the following optimization problem with parameter x:

$$\mathrm{P}_x \qquad \text{minimize}_y\ g(x,y) \qquad \text{subject to } y \in U(x) \tag{3}$$

(see, e.g., [13]). Furthermore, if $g(x,y)$ is pseudoconvex in y (i.e., $\langle \nabla_y g(x,y), y - z\rangle \leq 0$ implies $g(x,y) \leq g(x,z)$ for all $y, z \in U(x)$), then a vector $y \in U(x)$ is a solution to (2) if and only if it is a global optimal solution to (3). In this case, the mathematical programming problem with variational inequality constraints (1) is the *classical bilevel programming problem* (CBLP), or Stackelberg game (see, e.g., [1, 6, 17, 26, 29, 30, 31, 32]),

$$\text{CBLP} \qquad \text{minimize } f(x,y) \qquad \text{subject to } x \in X \text{ and } y \in \Sigma(x),$$

where $\Sigma(x)$ is the set of solutions for the problem P_x. The correspondence between lower level problems breaks down if F is not the partial gradient of a function with respect to y. Since problem (1) includes problems that are not classical bilevel programming problems, we call problem (1) a *generalized bilevel programming problem* (GBLP). The problem has been studied under the name "mathematical programs with equilibrium constraints" by other authors (see [12] and [19]).

In this paper we assume that

$$U(x) = \{y \in R^m : \ c(x,y) \leq 0\},$$

where $c : R^{n+m} \to R^d$ is a function. Throughout this paper we assume that f, c, and F are continuous. Under these assumptions, it is known [12, Lem. 1] that the solution set $S(x)$ of the variational inequality with parameter x is closed. Refer to [12] for the results on the existence of solutions for GBLP and CBLP.

Reducing a (generalized or classical) bilevel programming problem to a single level optimization problem is a useful strategy from both theoretical and computational points of view. There are several equivalent single level formulations for the GBLP. The Karush–Kuhn–Tucker (KKT) approach is to interpret the variational inequality constraint $y \in S(x)$ with y being a solution of the following optimization problem:

$$\text{minimize } \langle F(x,y), z\rangle \qquad \text{subject to } z \in U(x),$$

and to replace this minimization problem by its KKT necessary optimality conditions. These conditions are also sufficient if the feasible region $U(x)$ is convex. Assuming that $U(x)$ is convex, $c(x,y)$ is differentiable in y and one of the usual constraint qualifications, such as the Mangasarian–Fromowitz, condition is satisfied by the system of constraints $c(x,y) \leq 0$ in terms of variable y at a feasible point (x^*, y^*). Then (x^*, y^*) is a solution to the GBLP if and only if there exists $u^* \in R^d$ such that (x^*, y^*, u^*) is a solution to the following problem:

$$\begin{aligned} \text{KS} \qquad & \min f(x,y) \\ & \text{s.t. } F(x,y) + \nabla_y c(x,y)^T u = 0, \\ & \qquad \langle u, c(x,y)\rangle = 0, \\ & \qquad u \geq 0, c(x,y) \leq 0, \\ & \qquad x \in X, y \in R^m. \end{aligned} \tag{4}$$

To handle general GBLPs and CBLPs where $U(x)$ *may or may not be convex*, the value function and the gap function can be used to derive equivalent single level problems. Consider the CBLP. Define the *value function* $V(x) : X \to [-\infty, \infty]$ by

$$V(x) := \inf\{g(x, y) : y \in U(x)\}. \tag{5}$$

Then, for any $x \in X$, we have

$$g(x, y) - V(x) \geq 0 \ \forall y \in U(x), \text{ and } g(x, y) - V(x) = 0 \text{ if and only if } y \in \Sigma(x). \tag{6}$$

Thus, CBLP is equivalent to the following single level optimization problem:

$$\begin{array}{lll} \text{VS} & \min & f(x, y) \\ & \text{s.t.} & g(x, y) - V(x) = 0, \\ & & c(x, y) \leq 0, \\ & & x \in X, y \in R^m. \end{array} \tag{7}$$

Following [14] and [25], define the *gap function* $G_0(x, y) : X \times R^m \to [-\infty, \infty]$ by

$$G_0(x, y) := \sup\{\langle F(x, y), y - z\rangle : z \in U(x)\}. \tag{8}$$

It is easy to see that, for any $x \in X$,

$$G_0(x, y) \geq 0 \, \forall y \in U(x) \text{ and } G_0(x, y) = 0 \text{ if and only if } y \in S(x). \tag{9}$$

Hence GBLP is equivalent to the following single level optimization problem:

$$\begin{array}{lll} \text{GS} & \min & f(x, y) \\ & \text{s.t.} & G_0(x, y) = 0, \\ & & c(x, y) \leq 0, \\ & & x \in X, y \in R^m, \end{array} \tag{10}$$

Using the single level equivalent formulations KS, VS, and GS (see (4), (7), and (10)), one can derive Fritz John-type necessary optimality conditions for the original GBLP or CBLP. (See, e.g., [30] for the derivation of Fritz John-type necessary optimality conditions for CBLP.) In deriving Kuhn–Tucker-type necessary optimality conditions, however, we need to find constraint qualifications. Unfortunately, the usual constraint qualifications such as the Mangasarian–Fromowitz condition, never hold for problems VS and GS. To see this, for convenience, we assume that $U(x) = R^m$, $X = R^n$ and that $g(x, y), V(x)$, and $G_0(x, y)$ are Lipschitz continuous. Now suppose that (x^*, y^*) is a solution of GBLP. Then (6) and (9) imply the inclusions $0 \in \partial(g(x^*, y^*) - V(x^*))$ and $0 \in \partial G_0(x^*, y^*)$, respectively. These imply that there always exist abnormal multipliers for problems VS and GS. This is equivalent to saying that the Mangasarian–Fromowitz condition will never hold (see, e.g., [30, Prop. 3.1] for the equivalence). This phenomenon is intrinsic in bilevel problems. Even when using the KKT approach, the usual constraint qualifications will never hold for KS as long as the lower level problem is constrained. The following is a precise statement of this fact.

PROPOSITION 1.1. *Let (x^*, y^*, u^*) be a solution of KS. Suppose that $I := \{0 \leq i \leq d : c_i(x^*, y^*) = 0\} \neq \emptyset$. Then the Mangasarian–Fromowitz condition does not hold at (x^*, y^*, u^*).*

Proof. The complementary slackness condition (4) implies that $u_i^* = 0 \; \forall i \in I^c := \{0 \leq i \leq d : c_i(x^*, y^*) \neq 0\}$. So

$$c_i(x^*, y^*) = 0, \qquad i \in I$$

and

$$-u_i^* = 0, \qquad i \in I^c$$

are active constraints for KS at (x^*, y^*, u^*). Set

$$\widetilde{c}_i(x, y, u) := c_i(x, y),$$

$$\widehat{c}_i(x, y, u) := -u_i,$$

and

$$h(x, y, u) := \langle u, c(x, y) \rangle.$$

Suppose that there exists a vector $v \in R^{n+m+d}$ such that

$$\langle v, \nabla \widetilde{c}_i(x^*, y^*, u^*) \rangle = \sum_{j=1}^{n+m} v_j \nabla_j c_i(x^*, y^*) < 0 \; \forall i \in I$$

and

$$\langle v, \nabla \widehat{c}_i(x^*, y^*, u^*) \rangle = -v_{n+m+i} < 0 \; \forall i \in I^c,$$

where $\nabla_j c_i(x, y)$ denotes the gradient of c_i with respect to the jth component of the vector (x, y). Then

$$\begin{aligned} &\langle v, \nabla h(x^*, y^*, u^*) \rangle \\ &= \sum_{i \in I} u_i \sum_{j=1}^{n+m} v_j \nabla_j c_i(x^*, y^*) + \sum_{i \in I^c} v_{n+m+i} c_i(x^*, y^*) < 0. \end{aligned}$$

Thus, the Mangasarian–Fromowitz condition cannot hold at (x^*, y^*, u^*). □

The difficulty here is obviously due to the equality constraints (4), (7), and (10), which reflect the bilevel nature of the problem.

The partial calmness condition is identified in [30] as an appropriate constraint qualification for problem VS. It is also proved that the existence of a uniformly weak sharp minimum is a sufficient condition for partial calmness, and a parametric linear lower level problem is always partially calm.

Recently, using the theory of exact penalization for mathematical programming problems with subanalytic constraints and the theory of error bounds for quadratic inequality systems, Luo et al. [19] successfully derived various penalty functions for the single level equivalent mathematical programming problem KS. By using the theory of parametric normal equations, Luo et al. [19] also obtained some necessary and sufficient stationary point conditions for GBLP.

In this paper we use the *uniform parametric error bound* as a tool to establish (local or global) exact penalty formulations of several single level mathematical programming problems (including KS, VS, and GS) that are equivalent to GBLP. Since the exact penalty formulations move the troublesome equality constraints (4), (7), and (10) to the objective function, we can get Kuhn–Tucker-type necessary optimality conditions under the usual constraint qualifications. The concept of a uniform parametric error bound generalizes the uniformly weak sharp minimum defined in [30]. Thus, the uniform parametric error bounds derived in this paper provide many more exact penalty formulations than those in [30] for VS. Using the uniform parametric error bound as a tool, the conditions we derived in this paper are very general and distinct (cf. Theorem 6.5) from the ones derived in [19].

The paper is arranged as follows. In the next section we introduce uniform parametric error bounds and show that they provide local and global exact penalty formulations of GBLP. In section 3, we discuss several useful uniform parametric error bounds. Kuhn–Tucker-type necessary optimality conditions for problem GBLP associated with various uniform parametric error bounds are derived in section 4. In section 5, the relationships between various uniform parametric error bounds are discussed and some examples are given showing that the various equivalent single level optimization formulations with uniform parametric error bounds and their corresponding necessary optimality conditions complement each other. In section 6, we show that uniform parametric error bounds can be used to derive exact penalty formulations for KS.

2. Partial calmness and exact penalization.

In this section we introduce uniform parametric error bounds and show that they are useful in deriving exact penalty formulations for GBLP.

Consider the following mathematical programming problem:

$$\begin{array}{lll} \text{MP} & \text{minimize} & f(x) \\ & \text{subject to} & h(x)=0, \\ & & g(x)\le 0, \\ & & x\in C, \end{array}$$

where $f : R^n \to R$, $h : R^n \to R$, $g : R^n \to R^m$ are lower semicontinuous and C is a closed subset in R^n. The corresponding *perturbed problem* is

$$\begin{array}{lll} \text{MP}(\epsilon) & \text{minimize} & f(x) \\ & \text{subject to} & h(x)=\epsilon, \\ & & g(x)\le 0, \\ & & x\in C, \end{array}$$

where $\epsilon \in R$. The following definition was introduced in [30].

Definition 2.1 (partial calmness). *Let x^* solve* MP. *The problem* MP *is said to be* partially calm *at x^* provided that there exist constants $\mu > 0, \delta > 0$ such that, for all $\epsilon \in \delta B$ and all $x \in x^* + \delta B$ that are feasible for* MP*(ϵ), one has*

$$f(x) - f(x^*) + \mu|h(x)| \geq 0.$$

Here B denotes the open unit ball in R^n. The constants μ and δ are called the modulus and radius, respectively.

The partial calmness condition is similar to, but different from, the calmness condition introduced by Clarke and Rockafellar (see, e.g., [5]; see also Definition 4.1) in that only the equality constraint $h(x) = 0$ is perturbed.

The concept of calmness was shown to be closely related to "exact penalization" in [5, Prop. 6.4.3]. More precisely, if x^* is a local solution of MP and the problem MP is calm at x^*, then x^* is a local solution for a penalized problem. In the following proposition we show that the concept of partial calmness is equivalent to local exact penalization.

PROPOSITION 2.2. *Assume that f is continuous. Suppose x^* is a local minimum of* MP *and* MP *is partially calm at x^*. Then there exists $\mu^* > 0$ such that x^* is a local minimum of the following penalized problems for all $\mu \geq \mu^*$:*

$$\begin{array}{lll} \mathrm{MP}_\mu & \textit{minimize} & f(x) + \mu|h(x)| \\ & \textit{subject to} & g(x) \leq 0, \\ & & x \in C. \end{array}$$

Any local minima of MP_μ *with $\mu > \mu^*$ with respect to the neighborhood of x^* in which x^* is a local minimum are also local minima of* MP.

Proof. Suppose that x^* is a local minimum of MP but not MP_μ for any $\mu > 0$. Then, for each positive integer k, there exists a point $x_k \in x^* + (1/k)B \subset C$ and $g(x_k) \leq 0$ such that

$$f(x_k) + k|h(x_k)| < f(x^*). \tag{11}$$

Since x^* is a local minimum of MP, the above inequality implies that $|h(x_k)| > 0$. Therefore,

$$0 < |h(x_k)| < \frac{f(x^*) - f(x_k)}{k}. \tag{12}$$

Taking the limit as k goes to infinity in (12), one has

$$|h(x_k)| \to 0 \text{ as } k \to \infty.$$

But then the inequality (11) contradicts the hypothesis that MP is partially calm at x^*. Thus for some $\mu^* > 0$, x^* must be a local minimum of MP_{μ^*}.

It is obvious that a local minimum of MP_{μ^*} must be a local minimum for MP_μ whenever $\mu \geq \mu^*$.

Conversely, let $\mu > \mu^*$ and x_μ be a local minimum of MP_μ in the neighborhood of x^* in which x^* is a local minimum. Then

$$\begin{aligned} f(x_\mu) + \mu|h(x_\mu)| &= f(x^*) \qquad \text{since } x^* \text{ is a local minimum of } \mathrm{MP}_\mu, \\ &\leq f(x_\mu) + \frac{1}{2}(\mu + \mu^*)|h(x_\mu)| \qquad \text{since } \frac{1}{2}(\mu + \mu^*) > \mu^*, \end{aligned}$$

which implies that

$$(\mu - \mu^*)|h(x_\mu)| \leq 0.$$

Therefore, $h(x_\mu) = 0$, which implies that x_μ is also a local minimum of MP. □

Remark 2.3. Notice that in the above result, no continuity assumption is required for the function $h(x)$. When the function h is continuous, it is easy to see that if MP is partially calm at a solution x^* of MP with modulus μ and radius ϵ, then there exists a $\hat{\delta} \leq \delta$ such that x^* is a $\hat{\delta}$-local solution to the penalized problem MP_μ; i.e.,

$$f(x) + \mu|h(x)| \geq f(x^*) \quad \forall x \in C \text{ s.t. } g(x) \leq 0, x \in x^* + \hat{\delta}B.$$

Therefore, in our definition of partial calmness, the restriction on the size of perturbation $\epsilon \in \delta B$ can be removed when h is continuous, and it then corresponds to the definition of calmness given by Burke [2]. Furthermore, the infimum of μ^* in Proposition 2.2 can be taken as the modulus of partial calmness.

For any $x \in X, y \in R^m$, define the parametric distance function

$$d_{S(x)}(y) := \inf\{\|y - z\| : z \in S(x)\}$$

to be the distance from the point y to the set $S(x)$. The GBLP is equivalent to a *mathematical programming problem involving a parametric distance function constraint*:

$$\begin{array}{lll} \text{DP} & \text{minimize} & f(x,y) \\ & \text{subject to} & d_{S(x)}(y) = 0, \\ & & c(x,y) \leq 0, \\ & & x \in X, y \in R^m. \end{array}$$

It is known (see [5, Prop. 2.4.3]) that if the objective function of a constrained optimization problem is Lipschitz continuous then the distance function is an exact penalty term. In what follows, we extend this result to the mathematical programming problem with variational inequality constraints, GBLP. The constraint implied in the parametric distance function is, in fact, in the lower level decision variable. It is natural that we only need to assume that the objective function is locally Lipschitz in the lower level decision variable uniformly in the upper level decision variable to prove the exact penalty property of the parametric distance function. We need the following definition.

From now on we shall use $N(z)$ to denote a neighborhood of z.

Definition 2.4. *Let $(x^*, y^*) \in R^{n+m}$. The function $f(x,y)$ is said to be locally Lipschitz near y^* uniformly in $x \in N(x^*)$ if there exists $L > 0$ and a neighborhood $N(y^*)$ of y^* such that*

$$|f(x,y') - f(x,y)| \leq L|y' - y| \qquad \forall y', y \in N(y^*), x \in N(x^*).$$

The following result generalizes Proposition 2.4.3 of Clarke [5] to GBLP. We omit the proof of the global result, since it is essentially the same as the local one and the converse part of the proof in Proposition 2.2.

Theorem 2.5. *Let (x^*, y^*) be a local solution of problem* DP. *Assume that f is locally Lipschitz near y^* uniformly in x on a neighborhood of x^* with constant L. Then problem* DP *is partially calm at (x^*, y^*) with modulus L.*

Furthermore, let (x^*, y^*) be a global solution of GBLP and assume that $f(x, \cdot)$ is Lipschitz continuous in y with constant $L > 0$ uniformly for all $x \in X$. Then (x^*, y^*) is a global solution of the penalized problem

$$\begin{array}{lll} \text{DP}_\mu & \textit{minimize} & f(x,y) + \mu d_{S(x)}(y) \\ & \textit{subject to} & c(x,y) \leq 0, \\ & & x \in X, y \in R^m \end{array}$$

for any $\mu \geq L$, and any other global solution of DP_μ *for any $\mu > L$ is also a global solution of GBLP.*

Proof. Let $\delta > 0$ be such that (x^*, y^*) is a local solution of DP in $(x^*, y^*) + 2\delta B \subset X \times Y$. For any $0 \leq \epsilon < \delta$, let $(x,y) \in (x^*, y^*) + \delta B$ be feasible for DP_ϵ; i.e., $d_{S(x)}(y) = \epsilon$ and $c(x,y) \leq 0$, $(x,y) \in (x^*, y^*) + \delta B$. Since $S(x)$ is closed, one can choose a $y' \in S(x)$ such that $\|y' - y\| = \epsilon$. Since (x, y') is feasible for DP and

$$\begin{aligned} \|(x,y') - (x^*, y^*)\| &\leq \|(x,y') - (x,y)\| + \|(x,y) - (x^*, y^*)\| \\ &\leq \epsilon + \delta < 2\delta, \end{aligned}$$

we have

$$f(x,y') \geq f(x^*, y^*). \tag{13}$$

Since $f(x,\cdot)$ is locally Lipschitz near y^*,

$$f(x,y) - f(x,y') \geq -L\epsilon. \tag{14}$$

Combining (13) and (14) yields

$$f(x,y) - f(x^*,y^*) + L\epsilon \geq 0;$$

i.e., DP is partially calm at (x^*,y^*) with modulus L. □

Theorem 2.5 shows that the distance function provides an exact penalty equivalent formulation for GBLP under very mild conditions. However, the parametric distance function is usually an implicit nonsmooth function of the data in the original problem. It is difficult to compute or estimate its Clarke generalized gradient.

To overcome this difficulty, we shall use the parametric distance function $d_{S(x)}(y)$ establishing some equivalent exact penalty formulations of GBLP. These equivalent formulations have penalty functions with computable Clarke generalized gradients.

We call a function $r(x,y) : R^{n+m} \to R$ a *merit function* provided

$$r(x,y) \geq 0 \quad \forall (x,y) \in \mathrm{Gr}U \text{ and } r(x,y) = 0 \text{ if and only if } (x,y) \in \mathrm{Gr}S. \tag{15}$$

A merit function is called a *uniform parametric error bound* for the inclusion $y \in S(x)$ with modulus $\delta > 0$ in the set $Q \subset \mathrm{Gr}U$ if it satisfies

$$d_{S(x)}(y) \leq \delta r(x,y) \qquad \forall (x,y) \in Q. \tag{16}$$

A merit function provides the following equivalent formulation of GBLP:

$$\begin{array}{lll} \mathrm{RP} & \text{minimize} & f(x,y) \\ & \text{subject to} & r(x,y) = 0, \\ & & c(x,y) \leq 0, \\ & & x \in X, y \in R^m. \end{array}$$

Its corresponding penalized problem is

$$\begin{array}{lll} \mathrm{RP}_\mu & \text{minimize} & f(x,y) + \mu r(x,y) \\ & \text{subject to} & c(x,y) \leq 0, \\ & & x \in X, y \in R^m. \end{array}$$

Next we show that if $r(x,y)$ is a uniform parametric error bound and f is Lipschitz near y^* uniformly in x, then there exists $\mu > 0$ such that the problem RP_μ is an exact penalty equivalence of RP. As in Theorem 2.5 we omit the proof for the global result.

THEOREM 2.6. *Let (x^*,y^*) be a local solution of problem* GBLP *and r be a uniform parametric error bound with modulus $\delta > 0$ in a neighborhood of (x^*,y^*). Suppose that f is locally Lipschitz near y^* uniformly for all x in a neighborhood of x^*. Then there exists $\mu^* > 0$ such that (x^*,y^*) is a local solution of the penalized problem* RP_μ *for all $\mu \geq \delta\mu^*$ and any local solution to* RP_μ *with $\mu > \delta\mu^*$ with respect to the neighborhood of (x^*,y^*) is also a local solution to* RP.

Furthermore, let (x^*,y^*) be a global solution of GBLP *and r be a uniform parametric error bound in* GrU. *Assume that $f(x,\cdot)$ is Lipschitz continuous with constant $L > 0$ uniformly for all $x \in X$. Then (x^*,y^*) is a global solution of* RP_μ *for all $\mu \geq \delta L$, and any other global solution of* RP_μ *for all $\mu > \delta L$ is also a global solution of* GBLP.

Proof. Being a local solution of GBLP, (x^*,y^*) is also a local solution of DP. DP is partially calm by Theorem 2.5. Thus, by Proposition 2.2, there exists a $\mu^* > 0$

such that (x^*, y^*) is also a solution to DP_{μ^*}. Hence, for all (x, y) in a neighborhood of (x^*, y^*) which are feasible for $\mathrm{RP}_{\delta\mu^*}$, one has

$$\begin{aligned} f(x^*, y^*) + \delta\mu^* \cdot r(x^*, y^*) &= f(x^*, y^*) + \mu^* \cdot d_{S(x^*)}(y^*) \qquad \text{since } y^* \in S(x^*), \\ &\leq f(x, y) + \mu^* \cdot d_{S(x)}(y) \qquad \text{since } (x^*, y^*) \text{ solves } \mathrm{DP}_{\mu^*}, \\ &\leq f(x, y) + \delta\mu^* \cdot r(x, y) \quad \text{by inequality (16)}. \end{aligned}$$

Therefore, (x^*, y^*) is also a local solution of $\mathrm{RP}_{\delta\mu^*}$. The proof for the converse is similar to that of the converse part of Proposition 2.2. □

Remark 2.7. As in Remark 2.3 when the uniform parametric error bound r is continuous, the constant μ^* in Theorem 2.6 can be taken as the modulus of partial calmness, which is the Lipschitz constant of $f(x, \cdot)$ by virtue of Theorem 2.5.

Sometimes a uniform parametric error bound is not nicely behaved but its square is; e.g., $\sqrt{|x|}$ is not Lispchitz continuous near 0 but $|x|$ is. Therefore, we are interested in the following formulations which are equivalent to GBLP when $r(x, y)$ is a merit function.

$$\begin{array}{lll} \mathrm{RSP} & \text{minimize} & f(x,y) \\ & \text{subject to} & r^2(x,y) = 0, \\ & & c(x,y) \leq 0, \\ & & x \in X, y \in R^m. \end{array}$$

Its penalized problem is

$$\begin{array}{lll} \mathrm{RSP}_\mu & \text{minimize} & f(x,y) + \mu r^2(x,y) \\ & \text{subject to} & c(x,y) \leq 0, \\ & & x \in X, y \in R^m. \end{array}$$

Although the penalty term $r^2(x, y)$ might be better behaved, it is smaller than $r(x, y)$ for all (x, y) that are close to (x^*, y^*). Hence, to formulate an equivalent exact penalty formulation for the problem RSP, one needs to impose a stronger condition on f. The following definition gives such a condition.

Definition 2.8. *Let $x_0 \in X$. The mapping $f(x, y) : R^n \times R^m \to R$ is* upper Hölder continuous with exponent 2 near every $y \in S(x)$ uniformly for x in a neighborhood of x_0 *provided there exists $L > 0$ such that*

$$f(x, y') - f(x, y) \geq -L\|y' - y\|^2 \qquad \forall y' \in N(y), y \in S(x), x \in N(x_0).$$

The constant L is called the modulus.

We prove that $r^2(x, y)$ provides an exact penalty formulation for GBLP if $r(x, y)$ is a uniform parametric error bound and f is upper Hölder continuous with exponent 2 near every $y \in S(x)$ uniformly in x in a neighborhood of x^*.

Theorem 2.9. *Let (x^*, y^*) be a local solution of the problem* RSP. *Assume that r is a uniform parametric error bound with modulus δ in a neighborhood of (x^*, y^*) and that f is upper Hölder continuous with exponent* 2 *and modulus $L > 0$ near every $y \in S(x)$ uniformly in x in a neighborhood of x^*. Then (x^*, y^*) is a local solution of the penalized problem* RSP_μ *for all $\mu \geq \delta^2 L$, and any local solution to* RSP_μ *with $\mu > \delta^2\mu^*$ in the neighborhood of (x^*, y^*) is also a local solution to* RSP.

Proof. Let $\alpha > 0$ be such that (x^*, y^*) is a local solution of RSP in $(x^*, y^*) + \alpha(\delta + 1)B \subset X \times Y$. For any ε, $0 \leq \epsilon^{\frac{1}{2}} < \alpha$, let $(x, y) \in (x^*, y^*) + \alpha B$ be such that $r^2(x, y) = \epsilon, c(x, y) \leq 0$. Since $S(x)$ is closed, one can choose $y'(x) \in S(x)$ such that $\|y - y'(x)\| = d_{S(x)}(y) \leq \delta r(x, y) = \delta\epsilon^{\frac{1}{2}}$. Since $(x, y'(x))$ is feasible for RSP and

$$\begin{aligned} \|(x, y'(x)) - (x^*, y^*)\| &\leq \|(x, y'(x)) - (x, y)\| + \|(x, y) - (x^*, y^*)\| \\ &\leq \delta\epsilon^{\frac{1}{2}} + \alpha < \alpha(\delta + 1), \end{aligned}$$

we have

$$f(x, y'(x)) \geq f(x^*, y^*).$$

Therefore

$$\begin{aligned}
&f(x,y) - f(x^*, y^*)\\
&\geq f(x,y) - f(x,y'(x)) \qquad \text{by optimality of } (x^*, y^*),\\
&\geq -L\|y - y'(x)\|^2 \qquad \text{by upper Hölder continuity of } f,\\
&= -L(d_{S(x)}(y))^2,\\
&\geq -L\delta^2 r^2(x,y) \qquad \text{since } r(x,y) \text{ is a uniform parametric error bound},\\
&= -L\delta^2\epsilon;
\end{aligned}$$

i.e., RSP is partially calm at (x^*, y^*) with modulus $\delta^2 L$. The rest of the proof is similar to the converse part of Proposition 2.2. □

3. Some uniform parametric error bounds.

In this section we discuss some useful uniform parametric error bounds. We start with two definitions.

DEFINITION 3.1. *Let $\Omega \subset R^n$. A mapping $F(x,y) : R^n \times R^m \to R^m$ is called* strongly monotone with respect to y uniformly *in $x \in \Omega$ with modulus $\mu > 0$ provided*

$$\langle F(x,y) - F(x,z), y - z\rangle \geq \mu\|y - z\|^2 \quad \forall y, z \in U(x), x \in \Omega.$$

DEFINITION 3.2. *Let $\Omega \subset R^n$. The mapping $F(x,y) : R^n \times R^m \to R^m$ is called* pseudostrongly monotone with respect to y uniformly in $x \in \Omega$ *with modulus $\mu > 0$ provided*

$$\langle F(x,y), z - y\rangle \geq 0 \quad \textit{implies } \langle F(x,z), z - y\rangle \geq \mu\|z - y\|^2 \ \forall y, z \in U(x), x \in \Omega.$$

3.1. Uniformly weak sharp minima for the lower level optimization problem.

DEFINITION 3.3 (see [30]). *A family of parametric mathematical programming problems $\{(P_x) : x \in X\}$ as defined in* (3) *is said to have uniformly weak sharp minima in $\Omega \subset GrU$ if there exists an $\delta > 0$ such that*

$$d_{\Sigma(x)}(y) \leq \delta(g(x,y) - V(x)) \quad \forall (x,y) \in \Omega, \tag{17}$$

where $\Sigma(x)$ is the solution set of the lower level optimization problem P_x. *The constant δ is called the modulus of the uniformly weak sharp minima.*

By virtue of (9), $g(x,y) - V(x)$ is a merit function. When $\Sigma(x) = S(x)$ (e.g., when $U(x)$ is convex, $g(x,y)$ is pseudoconvex and differentiable in y), $g(x,y) - V(x)$ is obviously a uniform parametric error bound.

The next result follows easily from a result about regular points due to Ioffe (Theorem 1 and Corollary 1.1 of [8]).

PROPOSITION 3.4. *Let (x^*, y^*) be an optimal solution of the CBLP. Suppose that $g(x,y)$ is Lipschitz continuous in y uniformly in $x \in X$ with constant $L_g > 0$. Assume that there exist $\sigma > 0$ such that for any $(x,y) \in \mathrm{Gr}U$ satisfying $y \notin S(x)$ and any $\xi \in \partial_y g(x,y)$, $\eta \in (L_g + 1)\partial d_{S(x)}(y)$ (or $\eta \in N_{S(x)}(y)$),*

$$\|\xi + \eta\| \geq \sigma.$$

Then

$$d_{S(x)}(y) \leq (1/\sigma)(g(x,y) - V(x)) \ \forall (x,y) \in \mathrm{Gr}U.$$

Consider the bilevel programming problem where the lower level problem is the following parametric quadratic programming problem:

$$\begin{aligned} \mathrm{QP}_x \qquad & \min g(x,y) := \langle y, Px\rangle + \frac{1}{2}\langle y, Qy\rangle + p^t x + q^t y \\ & \text{s.t. } y \in \Omega_x := \{y \in Y : Ax + By - b \leq 0\}. \end{aligned}$$

Here $Q \in R^{m\times m}$ is a symmetric and positive semidefinite matrix, $p \in R^n$, $q \in R^m$, $P \in R^{m\times n}$; A and B are $d \times n$ and $d \times m$ matrices, respectively, and $b \in R^d$.

The next proposition gives a sufficient condition for the family of parametric quadratic programming problems $\{\mathrm{QP}_x : x \in R^n\}$ to have uniformly weak sharp minima.

PROPOSITION 3.5. *Assume that there exists a constant $M > 0$ such that for all $(x, y) \in \mathrm{Gr}S$, every element z of $(N(y, \Omega_x) + \mathrm{span}(\nabla_y g(x, \bar{y}))) \cap B$ can be expressed as*

$$z = \eta \nabla_y g(x, \bar{y}) + \xi,$$

where $|\eta| \leq M$ and $\xi \in N(y, \Omega_x)$. Assume

$$\ker(\nabla_y^2 g(x,\bar{y}))^\perp \subset \mathrm{span}(\nabla_y g(x,\bar{y})) + N(y,\Omega_x) \ \ \forall\ (x,y) \in \mathrm{Gr}S \tag{18}$$

or, equivalently,

$$(\nabla_y g(x,\bar{y}))^\perp \cap T(y,\Omega_x) \subset \ker(\nabla_y^2 g(x,\bar{y})) \ \ \forall\ (x,y) \in \mathrm{Gr}S,$$

where $\bar{y}$ is any element in $S(x)$, $A^\perp := \{y \in R^m : \langle y, x\rangle = 0 \ \forall x \in A\}$ denotes the subspace perpendicular to A, span(d) *represents the subspace generated by the vector d, $T(y, C)$ is the tangent cone to the set C at y, and* ker(A) *is the nullspace of the matrix A. Then $\{QP_x : x \in X\}$ has uniformly weak sharp minima.*

Before proving the above result we first state the following description of the solution set of a convex program given in Mangasarian [21].

LEMMA 3.6. *Let S be the set of solutions to the problem* $\min\{g(y) : y \in \Omega\}$ *where $g : R^n \to R$ is a twice continuously differentiable convex function and Ω is a convex subset of R^n. Let $\bar{y} \in S$. Then*

$$S = \{y \in \Omega : \nabla g(y) = \nabla g(\bar{y}), \langle \nabla g(\bar{y}), y - \bar{y}\rangle = 0\}.$$

It follows that for QP_x, the solution set $S(x)$ is

$$S(x) = \Omega_x \cap \{y : \langle \nabla_y g(x,\bar{y}), y - \bar{y}\rangle = 0\} \cap \{y : \nabla_y^2 g(x,\bar{y})(y - \bar{y}) = 0\}.$$

Since Ω_x is a polyhedral one has

$$T(y, S(x)) = T(y,\Omega_x) \cap (\nabla_y g(x,\bar{y}))^\perp \cap \ker(\nabla_y^2 g(x,\bar{y})) \tag{19}$$

by virtue of Corollaries 16.4.2 and 23.8.1 of Rockafellar [28].

Proof of Proposition 3.5. By virtue of Theorem 2.6 of Burke and Ferris [4], it suffices to show that for all $x \in X, y \in S(x)$, there exists an $\alpha > 0$ such that

$$g_2'(x, y; d) \geq \alpha \|d\| \quad \forall d \in T(y, \Omega_x) \cap N(y, S(x)),$$

where $g_2'(x, y; d)$ is the directional derivative of g with respect to y in the direction d. Note that (19) and (18) imply that

$$\begin{aligned} N(y, S(x)) &= N(y, \Omega_x) + \operatorname{span}(\nabla_y g(x, \bar{y})) + \ker(\nabla_y^2 g(x, \bar{y}))^{\perp} \\ &= N(y, \Omega_x) + \operatorname{span}(\nabla_y g(x, \bar{y})). \end{aligned}$$

Since $d \in T(y, \Omega_x) \cap N(y, S(x))$, one has

$$\begin{aligned} \|d\| &= \sup\{\langle z(x), d\rangle : z(x) \in B \cap N(y, S(x))\} \\ &\leq \sup\{\langle \eta \nabla_y g(x, \bar{y}) + \xi, d\rangle : |\eta| < M, \xi \in N(y, \Omega_x)\} \\ &\leq M\langle \nabla_y g(x, \bar{y}), d\rangle = M\langle \nabla_y g(x, y), d\rangle = M g_2'(x, y; d). \end{aligned}$$

The first inequality follows from the assumption, and the second equality follows from Lemma 3.6. Setting $\alpha = 1/M$ completes the proof. □

The following bilinear programming problem with parameter x is a special case of QP_x.

$$\begin{aligned} \mathrm{BLP}_x \qquad & \min \langle y, Px\rangle + p^t x + q^t y \\ & \text{s.t. } Ax + By - b \leq 0, \\ & \quad y \in R^m. \end{aligned}$$

Proposition 3.5 has the following simple consequence.

COROLLARY 3.7. *The bilinear programming problem* BLP_x *has a uniformly weak sharp minima if there exists a constant* $M > 0$ *such that for all* $(x, y) \in \mathrm{Gr}S$, *every element* z *of* $(N(y, \Omega_x) + \operatorname{span}(Px + q)) \cap B$ *can be expressed as*

$$z = \eta(Px + q) + \xi$$

where $|\eta| \leq M$ *and* $\xi \in N(y, \Omega_x)$.

The following example shows that the assumption in Corollary 3.7 cannot be omitted.

Example 3.8. Consider the problem

$$\begin{aligned} & \min x + y \\ & \text{s.t. } 0 \leq x \leq 1, y \in \arg\min\{-xy : x + y - 1 \leq 0, y \geq 0\}. \end{aligned}$$

The solution set of the lower level problem is

$$S(x) = \begin{cases} [0, 1] & \text{if } x = 0, \\ 1 - x & \text{if } 0 < x \leq 1. \end{cases}$$

The value function of the lower problem is

$$V(x) = \begin{cases} 0 & \text{if } x = 0, \\ -x(1 - x) & \text{if } 0 < x \leq 1. \end{cases}$$

It is easy to check that the assumption in Corollary 3.7 is not satisfied and there is no uniformly weak sharp minimum. In fact, if we replace the constraint $0 \leq x \leq 1$ by $0 < \epsilon \leq x \leq 1$, then the assumption in Corollary 3.7 is satisfied, and uniformly weak sharp minima exist.

3.2. A standard gap bound. Consider a parametric variational inequality with *nonseparable and linear constraints*, i.e.,

$$U(x) = \{y \in R^m | c(x, y) = Ax + By - b \leq 0\}, \tag{20}$$

where A and B are $d \times n$ and $d \times m$ matrices, respectively, and $b \in R^d$. In this case, $\forall x_0 \in X$, $y_0 \in U(x_0)$ solve the variational inequality with parameter x_0 (see (2)) if and only if there exists $\lambda_0 \in R^d$ such that (x_0, y_0, λ_0) satisfies the following *complementarity system*:

$$\begin{aligned} F(x_0, y_0) + B^T \lambda_0 = 0, \\ (Ax_0 + By_0 - b)^T \lambda_0 = 0, \\ Ax_0 + By_0 - b \leq 0, \lambda_0 \geq 0. \end{aligned}$$

If the *gradients of the binding constraints* in the variational inequality (2) at (x_0, y_0), i.e., those $\nabla_y c_j(x_0, y_0)$ such that $c_j(x_0, y_0) = 0, j \in \{1, 2, \dots, d\}$, are *linearly independent*, and the *strict complementarity condition*

$$\lambda_{0i} > 0 \iff c_i(x_0, y_0) = 0, \qquad \forall i \in \{1, 2, \dots, d\} \tag{21}$$

holds, then the variational inequality (2) with parameter x has a unique solution $y(x)$ for all x in a neighborhood of x_0, and the above complementarity system has a unique solution $(y(x), \lambda(x))$ for all x in a neighborhood of x_0. Furthermore, the functions $y(x)$ and $\lambda(x)$ are Lipschitz continuous, and the strict complementarity condition (21) is satisfied in a neighborhood of x_0 (see, e.g., Friesz et al. [10]).

The following result due to Marcotte and Zhu [25] shows that the gap function defined by (8) can serve as a uniform parametric error bound under certain conditions.

PROPOSITION 3.9. *Assume that X is a compact, convex subset of R^n and $U(x)$ defined as in* (20) *is compact. Let the mapping F be strongly monotone with respect to y uniformly in $x \in X$, and let $\nabla_y F$ be Lipschitz continuous in y uniformly in x. Suppose $x_0 \in X$. If the linear independence and strict complementarity conditions hold at $y_0 = y(x_0)$, then there exists a constant $\delta > 0$ and a neighborhood of (x_0, y_0) such that*

$$d_{S(x)}(y) \leq \delta G_0(x, y) \qquad \forall (x, y) \in \mathrm{Gr} U \cap N(x_0, y_0).$$

Now we consider a parametric variational inequality with separable and linear constraints; i.e., $U(x) = \{y \in R^m | By \leq b\}$ is a convex polyhedron. In this case we can weaken the assumptions of Proposition 3.9.

We need the following definition due to Dussault and Marcotte [7].

DEFINITION 3.10. *Let F be a continuous, monotone mapping from a convex polyhedron $X \subset R^n$ into R^n and denote by* VIP*(X, F) the variational inequality problem associated with X and F; i.e., find x^* in X such that*

$$\mathrm{VIP}(F, X) \qquad\qquad \langle F(x^*), x^* - x \rangle \leq 0 \qquad \textit{for all } x \textit{ in } X.$$

We say that VIP*(F, X) is* geometrically stable *if, for any solution x^* of the variational inequality, $\langle F(x^*), x^* - x \rangle = 0$ implies that x lies on the optimal face, i.e., the minimal face of X containing the (convex) solution set to* VIP*(F, X).*

The following result due to Marcotte and Zhu [25] gives a useful error bound.

PROPOSITION 3.11. *Assume that X is a convex polyhedron, $U(x) = \{y : By - b \leq 0\}$ is compact, and the mapping F is strongly monotone with respect to y uniformly in $x \in X$. Let $x_0 \in X$ and assume that there exists a neighborhood of x_0 such that $\mathrm{VIP}(F(x,\cdot), Y)$ is geometrically stable inside that neighborhood. Then there exist some neighborhood $N(x_0)$ of x_0 and a positive number $\delta > 0$ such that*

$$d_{S(x)}(y) \leq \delta G_0(x,y) \qquad \forall y \in U(x), x \in N(x_0).$$

3.3. A square root standard gap bound. The following result gives a uniform parametric error bound in terms of the square root of the gap function G_0.

PROPOSITION 3.12. *Assume that the mapping F is pseudostrongly monotone with respect to y uniformly in $x \in N(x_0)$ with modulus μ. Then one has*

$$d_{S(x)}(y) \leq \frac{\sqrt{\mu}}{\mu}\sqrt{G_0(x,y)} \qquad \forall y \in U(x), x \in N(x_0).$$

Proof. Let $y(x) \in S(x)$. Then, by the definition of $S(x)$, one has

$$\langle F(x, y(x)), y - y(x)\rangle \geq 0 \qquad \forall y \in U(x).$$

Since $y(x) \in U(x)$, it follows from the pseudostrong monotonity of F and the definition of G_0 that, for all $x \in N(x_0)$ and $y \in U(x)$, one has

$$\mu\|y(x) - y\|^2 \leq \langle F(x,y), y - y(x)\rangle \leq G_0(x,y),$$

from which the result follows readily. □

3.4. A square root differentiable gap bound. Recently, Fukushima [11] gave an optimization formulation of a variational inequality based on the *differentiable gap function* defined as

$$G_\alpha(x,y) = \max_{z \in U(x)} \left\{ \langle F(x,y), y - z\rangle - \frac{1}{2\alpha}\|y - z\|_M^2 \right\}, \tag{22}$$

where $\alpha > 0$ is a given constant, $\|\cdot\|_M$ denotes the elliptic norm in R^m defined by $\|z\|_M = \langle z, Mz\rangle^{\frac{1}{2}}$, and M is a symmetric positive definite matrix. It is easy to see that the differentiable gap function G_α satisfies condition (15). The following result gives a uniform parametric error bound based on $\sqrt{G_\alpha}$.

PROPOSITION 3.13. *Suppose $U(x)$ is convex and $x_0 \in X$. Let the mapping F be pseudostrongly monotone with respect to y uniformly in $x \in N(x_0)$. Then there exists $\delta > 0$ such that*

$$d_{S(x)}(y) \leq \delta\sqrt{G_\alpha(x,y)} \qquad \forall y \in U(x), x \in N(x_0).$$

Proof. Let $y(x) \in S(x)$. Then, by the definition of $S(x)$, one has

$$\langle F(x, y(x)), y - y(x)\rangle \geq 0 \qquad \forall y \in U(x).$$

Since $y(x) \in U(x)$, it follows from the pseudostrong monotonity of F that, for every $x \in N(x_0)$ and $y \in U(x)$,

$$\langle F(x,y), y - y(x)\rangle \geq \mu\|y - y(x)\|^2.$$

Let $y_t = y + t(y(x) - y)$ for $t \in [0,1]$. By the convexity of $U(x)$, $y_t \in U(x)$ for any $y \in U(x)$. It follows from the definition of $G_\alpha(x,y)$ (see (22)) that

$$\begin{aligned} G_\alpha(x,y) &\geq \langle F(x,y), y - y_t\rangle - \frac{1}{2\alpha}\|y - y_t\|_M^2 \\ &= t\langle F(x,y), y - y(x)\rangle - \frac{t^2}{2\alpha}\|y - y(x)\|_M^2 \\ &\geq \left(t\mu - \frac{t^2\|M\|}{2\alpha}\right)\|y - y(x)\|^2. \end{aligned}$$

Letting $t = \min\{1, \frac{\alpha\mu}{\|M\|}\}$ gives

$$G_\alpha(x,y) \geq \sigma\|y - y(x)\|^2,$$

where

$$\sigma = \begin{cases} (\mu - \frac{\|M\|}{2\alpha}) & \text{if } \mu \geq \frac{\|M\|}{\alpha}, \\ \frac{\alpha\mu^2}{2\|M\|} & \text{if } \mu \leq \frac{\|M\|}{\alpha}. \end{cases}$$

This proves the result. □

3.5. A projection bound. The following projection characterization of $y \in S(x)$ is well known (see, e.g., [15]).

LEMMA 3.14. *An arbitrary vector $y \in Y$ is a solution of the variational inequality with parameter x if and only if it satisfies*

$$h(x,y) = y - \text{proj}_{U(x)}(y - F(x,y)) = 0$$

where $\text{proj}_{U(x)}(z)$ is the orthogonal projection of a vector z onto the set $U(x)$.

It follows from the above lemma that any vector norm of $h(x,y)$ satisfies condition (15). The following result is a parametric version of [27, Thm. 3.1]. The proof is omitted since it is essentially the same as that of [27, Thm. 3.1].

PROPOSITION 3.15. *Let $x_0 \in X$. Assume that the mapping F is strongly monotone with respect to y uniformly in $N(x_0)$ with modulus μ, and F is Lipschitz continuous in y with constant $L_F > 0$ uniformly in $x \in N(x_0)$. Then we have*

$$d_{S(x)}(y) \leq ((L_F + 1)/\mu)\|h(x,y)\| \qquad \forall y \in U(x), x \in N(x_0). \tag{23}$$

Remark 3.16. An important special case of GBLP is one where $F(x,y) = Qx + My + q$ and $U(x) = R_+^m$, the nonnegative orthant in R^m. In this case, finding a solution $y \in R^m$ to the parametric variational inequality (1) reduces to the parametric linear complementarity problem of finding a $y \in R^m$ satisfying

$$y \geq 0, Qx + My + q \geq 0, \langle y, Qx + My + q\rangle = 0.$$

The uniform projection error bound holds when M is a P-matrix (see Mathias and Pang [24]) and when M is an R_0-matrix. (See Mangasarian and Ren [23] and Luo and Tseng [18].)

4. Kuhn–Tucker-type necessary optimality conditions. In this section we derive Kuhn–Tucker-type necessary optimality conditions for GBLP.

Without loss of generality, we assume in this section that all solutions of the mathematical programming problems lie in the interior of their abstract constraint sets.

First we give a concise review of the material on nonsmooth analysis. Our reference is Clarke [5].

Consider the following mathematical programming problem:

$$\begin{array}{lll} \text{P} & \text{minimize} & \phi(x,y) \\ & \text{subject to} & c(x,y) \le 0, \\ & & x \in X, y \in R^m. \end{array}$$

The corresponding perturbed problem is

$$\begin{array}{lll} \text{P}(\alpha) & \text{minimize} & \phi(x,y) \\ & \text{subject to} & c(x,y) + \alpha \le 0, \\ & & x \in X, y \in R^m, \end{array}$$

where $\phi(x,y) : R^{n+m} \to R$ and $c(x,y) : R^{n+m} \to R^d$ are locally Lipschitz near the points of interest.

DEFINITION 4.1 (calmness). *Let (x^*,y^*) solve* P. *Problem* P *is* calm *at (x^*,y^*) provided that there exist $\delta > 0$ and $\mu > 0$ such that for all $\alpha \in \delta B$, for all $(x,y) \in (x^*,y^*) + \delta B$ which are feasible for P(α), one has*

$$\phi(x,y) - \phi(x^*,y^*) + \mu\|\alpha\| \ge 0.$$

DEFINITION 4.2 (abnormal and normal multipliers). *Let (x,y) be feasible for* P. *Define $M^0(x,y)$, the set of* abnormal multipliers *corresponding to (x,y), as the set*

$$M^0(x,y) := \{s \in R^d : 0 \in \partial c(x,y)^\top s, s \ge 0, \langle s, c(x,y)\rangle = 0\}.$$

Define $M^1(x,y)$, the set of normal multipliers *corresponding to (x,y), as the set*

$$M^1(x,y) := \{s \in R^d : 0 \in \partial\phi(x,y) + \partial c(x,y)^\top s, s \ge 0, \langle s, c(x,y)\rangle = 0\}.$$

Remark 4.3. A sufficient condition for P to be calm at (x^*,y^*) is $M^0(x^*,y^*) = \{0\}$. $M^0(x^*,y^*) = \{0\}$ if and only if the Mangasarian–Fromowitz conditions are satisfied [30].

PROPOSITION 4.4 (Kuhn–Tucker Lagrange multiplier rule). *Let (x^*,y^*) solve P. Suppose ϕ, c are locally Lipschitz near (x^*,y^*) and problem P is calm at (x^*,y^*). Then there exists $s \ge 0$ such that*

$$0 \in \partial\phi(x^*,y^*) + \partial c(x^*,y^*)^\top s$$

and

$$0 = \langle s, c(x^*,y^*)\rangle.$$

The following theorem gives a necessary condition for optimality when an error bound $r(x,y)$ is explicitly known.

THEOREM 4.5. *Let (x^*,y^*) be a solution of problem* GBLP. *Let $r(x,y)$ be a uniform parametric error bound in a neighborhood of (x^*,y^*) and* RP_μ *be the associated penalized problem of* RP, *where $\mu > 0$. Assume that f and r are locally Lipschitz near (x^*,y^*) and the associated penalized problem* RP_μ *is calm at (x^*,y^*). Then there exists a nonzero vector $s \ge 0$ such that*

$$0 \in \partial f(x^*,y^*) + \mu\partial r(x^*,y^*) + \partial c(x^*,y^*)^\top s$$

and

$$0 = \langle s, c(x^*, y^*)\rangle.$$

Proof. By Theorem 2.6, (x^*, y^*) is also a solution of the associated penalized problem RP_μ. The result follows from Proposition 4.4. □

However, in many cases, uniform parametric error bounds are implicit functions of the original problem data. The useful uniform parametric error bounds derived in section 4 involve the class of marginal functions or value functions. In order to derive necessary conditions in these cases, one must first study the generalized differentiability of marginal functions.

Consider the following parametric mathematical programming problem:

$$\begin{array}{lll} \mathrm{P}_\alpha & \text{minimize} & \phi(\alpha, y) \\ & \text{subject to} & c(\alpha, y) \le 0, \\ & & y \in R^m. \end{array}$$

We assume that for problem P_α the functions ϕ and c are locally Lipschitz near the point of interest $y_0 \in R^m$. Let y be feasible for P_α. Define

$$\begin{aligned} M^0_\alpha(y) &:= \{\pi \in R^d : 0 \in \partial_y c(\alpha, y)^\top \pi, \langle \pi, c(\alpha, y)\rangle = 0, \pi \ge 0\}, \\ M^1_\alpha(y) &:= \{\pi \in R^d : 0 \in \partial_y \phi(\alpha, y) + \partial_y c(\alpha, y)^\top \pi, \langle \pi, c(\alpha, y)\rangle = 0, \pi \ge 0\}. \end{aligned}$$

Let $W(\alpha) = \inf\{\phi(\alpha, y) : c(\alpha, y) \le 0, y \in Y\}$. The following result is an easy consequence of Corollary 1 of Theorem 6.5.2 of Clarke [5].

Proposition 4.6 (generalized differentiability of marginal functions). *Let Σ_{α_0} be the solution set to problem P_{α_0} and suppose it is nonempty. Suppose $M^0_{\alpha_0}(\Sigma_{\alpha_0}) = \{0\}$. Then $W(\alpha)$ is Lipschitz near α_0, and one has*

$$\partial W(\alpha_0) \subset \mathrm{clco}\{\partial_\alpha \phi(\alpha_0, y) + \partial_\alpha c(\alpha_0, y)^\top \pi : y \in \Sigma_{\alpha_0}, \pi \in M^1_{\alpha_0}(y)\},$$

where clco$A$ *denotes the closed convex hull of the set A.*

Set $G_0(x, y) = -\min\{\langle F(x, y), z - y\rangle : c(x, z) \le 0, z \in R^m\}$. The parameter here is $\alpha = (x, y)$. Let $\Sigma_{(x,y)}$ denote the set of vectors at which $G_0(x, y)$ attains the maximum. By Proposition 4.6, one has the following result.

Proposition 4.7. *Suppose $M^0_{(x^*,y^*)}(\Sigma_{(x^*,y^*)}) = \{0\}$. Assume that f, F, and c are locally Lipschitz near (x^*, y^*) and that $\partial F(x^*, y^*) \subset \partial_x F(x^*, y^*) \times \partial_y F(x^*, y^*)$. Then $G_0(x, y)$ is Lipschitz near (x^*, y^*) and one has*

$$\begin{aligned} &\partial G_0(x^*, y^*) \\ &\quad \subset \mathrm{co}\{(\partial_x F(x^*, y^*)^\top (y^* - y) - \partial_x c(x^*, y)^\top \pi, \partial_y F(x^*, y^*)^\top (y^* - y) + F(x^*, y^*)) : \\ &\qquad\qquad\qquad\qquad\qquad\qquad y \in \Sigma_{(x^*,y^*)}, \pi \in M^1_{(x^*,y^*)}(y)\}, \end{aligned}$$

where

$$\begin{aligned} M^0_{(x^*,y^*)}(y) &= \{\pi \in R^d : 0 \in \partial_y c(x^*, y)^\top \pi, \pi \ge 0, \langle \pi, c(x^*, y)\rangle = 0\}, \\ M^1_{(x^*,y^*)}(y) &= \{\pi \in R^d : 0 \in F(x^*, y^*) + \partial_y c(x^*, y)^\top \pi, \pi \ge 0, \langle \pi, c(x^*, y)\rangle = 0\}. \end{aligned}$$

Combining Proposition 4.7, Remark 4.3, and Theorems 2.6, 2.9, and 4.5, one has the following result.

Theorem 4.8. *Suppose f, F, and c are C^1. Let (x^*, y^*) be a solution of* GBLP. *Assume either of the following assumptions is satisfied:*

- *$G_0(x, y)$ is a uniform parametric error bound in a neighborhood of (x^*, y^*).*
- *$\sqrt{G_0(x, y)}$ is a uniform parametric error bound in a neighborhood of (x^*, y^*) and f is upper Hölder continuous with exponent 2 near every $y \in S(x)$ uniformly in x in a neighborhood of x^*.*

Suppose $M^0_{(x^*,y^*)}(\Sigma_{(x^*,y^*)}) = \{0\}$. Then there exist $\mu > 0$, $s \in R^d$, positive integers I, J, $\lambda_{ij} \geq 0, \sum_{i=1}^I \sum_{j=1}^J \lambda_{ij} = 1, y_i \in \Sigma_{(x^*,y^*)}$, and $\pi_{ij} \in R^d$ such that

$$0 = \nabla_x f(x^*, y^*) + \nabla_x c(x^*, y^*)^\top s + \mu \sum_{ij} \lambda_{ij}\{\nabla_x F(x^*, y^*)^\top (y^* - y_i) - \nabla_x c(x^*, y_i)^\top \pi_{ij}\},$$

$$0 = \nabla_y f(x^*, y^*) + \nabla_y c(x^*, y^*)^\top s + \mu \sum_{ij} \lambda_{ij}\{\nabla_y F(x^*, y^*)^\top (y^* - y_i) + F(x^*, y^*)\},$$

$$0 = \langle s, c(x^*, y^*)\rangle, s \geq 0,$$
$$0 = F(x^*, y^*) + \nabla_y c(x^*, y_i)^\top \pi_{ij},$$
$$0 = \langle \pi_{ij}, c(x^*, y_i)\rangle, \pi_{ij} \geq 0.$$

For $G_\alpha(x, y)$, the differentiable gap function defined in (22), since y is the unique solution in the right-hand side of (22), we have $\Sigma_{(x,y)} = \{y\}$. By Proposition 4.6, one has the following result.

PROPOSITION 4.9. *Suppose f, F, and c are locally Lipschitz near (x^*, y^*). Assume that $M^0_{(x^*,y^*)}(y^*) = \{0\}$. Then $G_\alpha(x, y)$ is Lipschitz near (x^*, y^*) and one has*

$$\partial G_\alpha(x^*, y^*) \subset \{(-\partial_x c(x^*, y^*)^\top \pi, F(x^*, y^*)) : \ \pi \in M^1_{(x^*,y^*)}(y^*)\},$$

where

$$M^1_{(x^*,y^*)}(y^*) = \{\pi \in R^d : 0 \in F(x^*, y^*) + \partial_y c(x^*, y^*)^\top \pi, \pi \geq 0, \langle \pi, c(x^*, y^*)\rangle = 0\}.$$

Furthermore, if c is a C^1 function and $M^1_{(x,y)}(y) = \{\pi\}$ is a singleton, then $G_\alpha(x, y)$ is C^1 and one has

$$\nabla G_\alpha(x, y) = (-\nabla_x c(x, y)^\top \pi, F(x, y)).$$

Combining Proposition 4.9, Remark 4.3, and Theorems 2.6, 2.9, and 4.5, one has the following result.

THEOREM 4.10. *Let (x^*, y^*) be a solution of* GBLP. *Suppose F is locally Lipschitz near (x^*, y^*) and f and c are C^1 functions. Assume that either of the following assumptions is satisfied:*

- *$G_\alpha(x, y)$ is a uniform parametric error bound in a neighborhood of (x^*, y^*).*
- *$\sqrt{G_\alpha(x, y)}$ is a uniform parametric error bound in a neighborhood of (x^*, y^*) and f is upper Hölder continuous with exponent 2 near every $y \in S(x)$ uniformly in x in a neighborhood of x^*.*

Suppose $M^0_{(x^*,y^*)}(y^*) = \{0\}$. Then there exist $\mu > 0$, $s \in R^d$, and $\pi \in R^d$ such that

$$0 = \nabla_x f(x^*, y^*) + \nabla_x c(x^*, y^*)^\top s - \mu \nabla_x c(x^*, y^*)^\top \pi,$$
$$0 = \nabla_y f(x^*, y^*) + \nabla_y c(x^*, y^*)^\top s + \mu F(x^*, y^*),$$
$$0 = \langle s, c(x^*, y^*)\rangle = 0, s \geq 0,$$
$$0 = F(x^*, y^*) + \nabla_y c(x^*, y^*)^\top \pi,$$
$$0 = \langle \pi, c(x^*, y^*)\rangle, \pi \geq 0.$$

Remark 4.11. To shorten the exposition, we have assumed in Theorems 4.8 and 4.10 that f, F, g, and c are C^1 functions. However, these theorems can also be stated without difficulty when f, F, g, and c are merely Lipschitz continuous.

5. Relationships between various uniform parametric error bounds. In this section, we study the relationships between various uniform parametric error bounds. Through illustrative examples we show that various equivalent single level optimization formulations with uniform parametric error bounds and their corresponding necessary optimality conditions complement each other.

The following result is easy to prove.

PROPOSITION 5.1. *Suppose that r_S and r_B are two merit functions that satisfy the following inequality:*

$$r_S(x,y) \le \delta r_B(x,y) \qquad \forall (x,y) \in \mathrm{Gr}U,$$

for a constant $\delta > 0$. If $r_S(x,y)$ is a uniform parametric error bound, then so is $r_B(x,y)$.

Motivated by the above result we now establish certain inequalities and equalities among various uniform parametric error bounds.

PROPOSITION 5.2.

(1) *If the objective function $g(x,y)$ of the lower level optimization problem* (3) *is convex and C^1 (continuously differentiable) in y, then*

$$g(x,y) - V(x) \le G_0(x,y). \tag{24}$$

Furthermore, if the lower level problem is linear, then

$$g(x,y) - V(x) = G_0(x,y).$$

(2) *For* GBLP*, we have*

$$\sqrt{G_\alpha(x,y)} \le \sqrt{G_0(x,y)}. \tag{25}$$

(3) *For (x,y) in a neighborhood of the solution (x^*,y^*) of* GBLP*,*

$$G_0(x,y) \le \sqrt{G_0(x,y)}. \tag{26}$$

(4) *For* GBLP*, we have*

$$\|h(x,y)\| \le \sqrt{2G_0(x,y)}. \tag{27}$$

Proof. (1) Let $y(x) \in \arg\min_{y \in U(x)} g(x,y)$. By the convexity of $g(x,\cdot)$ and the definition of G_0, we have

$$\begin{aligned} G_0(x,y) &\ge \langle \nabla_y g(x,y), y - y(x) \rangle \\ &\ge g(x,y) - g(x,y(x)) \\ &= g(x,y) - V(x). \end{aligned}$$

The second assertion follows from the definitions of $V(x)$ and $G_0(x,y)$.

(2) This follows directly from the definitions of G_α and G_0.

(3) Since G_0 is continuous in (x,y) and $G_0(x^*,y^*) = 0$, $G_0(x,y) < 1$ in a neighborhood of the solution (x^*,y^*) of GBLP. This implies the result.

(4) Taking $\alpha = 1$ and $M = I$ the identity matrix in the definition of G_α, we have

$$G_1(x,y) = \langle F(x,y), y - p(x,y) \rangle - \frac{1}{2}\|y - p(x,y)\|^2 \ge 0,$$

where $p(x,y) = \mathrm{Proj}_{U(x)}(y - F(x,y))$. Thus

$$\begin{aligned} G_0(x,y) &\geq \langle F(x,y), y - p(x,y)\rangle \\ &\geq \frac{1}{2}\|y - p(x,y)\|^2 = \frac{1}{2}\|h(x,y)\|^2. \end{aligned}$$

The proof is completed. □

As shown in section 4, one of the major applications of the exact penalty formulation with uniform parametric error bounds is to derive Kuhn–Tucker-type necessary optimality conditions. For this purpose parametric error bounds must be Lipschitz continuous (see Theorem 4.5). Among the aforementioned error bounds, G_0, h, and $g - V$ are Lipschitz continuous under appropriate constraint qualifications on $U(x)$. The rest are generally not Lipschitz. By virtue of Proposition 5.1, if we have an exact penalty formulation with a given uniform parametric error bound then a similar exact penalty formulation is also valid, with that error bound replaced by a larger one. Smaller error bounds generally require stronger conditions. Hence, on one hand, error bounds G_0, h, and $\phi - V$ can be Lipschitz continuous but require stronger conditions. On the other hand, larger bounds such as $\sqrt{G_0}$ may not be Lipschitz continuous but require weaker conditions. In the case when uniform parametric error bounds are not Lipschitz continuous, Theorems 4.8 and 4.10 show that stronger assumptions, such as upper Hölder continuity on the upper level objective functions, may be required. Therefore, various error bounds and their equivalent exact penalty representations complement each other. The following are some illustrative examples.

Example 5.3. Consider the following classical bilevel programming problem:

$$\begin{aligned} \text{(P1)} \quad & \min x^2 - 2y \\ & \text{s.t. } x \in [0,2] \text{ and } y \in \arg\min\{y^2 - 2xy : y \in [0,2x]\}. \end{aligned}$$

It is easy to verify that $(1,1)$ is the unique solution of (P1) and assumption (18) does not hold. Therefore, Proposition 3.5 does not apply and one may suspect that (P1) does not have a uniformly weak sharp minimum. Indeed, direct calculation shows that the value function for the lower level problem is $V(x) = x^2$. Using the value function approach, problem (P1) is equivalent to the following problem:

$$\begin{aligned} & \min x^2 - 2y \\ & \text{s.t. } (y - x)^2 = 0, \\ & \quad\quad y \in [0,2x], x \in [0,2]. \end{aligned}$$

Here $(y - x)^2$ is not an exact penalty term for the above problem, since for any $\mu > 0$ $(1,y)$ where $y \in (1, \frac{2+\mu}{\mu})$ assigns a lower value to the objective function than $(1,1)$ in the penalized problem

$$\begin{aligned} & \min x^2 - 2y + \mu(y - x)^2 \\ & \text{s.t. } y \in [0,2x], x \in [0,2]. \end{aligned}$$

It is clear that the function $F(x,y) = \nabla_y g(x,y) = y - x$ is strongly monotone in y uniformly for $x \in R$. The standard gap function takes the form

$$\begin{aligned} G_0(x,y) &= \max_{z \in [0,2x]} \langle y - x, y - z\rangle \\ &= y^2 - xy + \max_{z \in [0,2x]} \langle y - x, -z\rangle \\ &= y^2 - xy - x[(y - x) - |y - x|] \\ &= (y - x)^2 + x|y - x|. \end{aligned}$$

The linear independence and the strict complementarity conditions can easily be verified at $(1,1)$. Hence, by Proposition 3.9, the gap function $G_0(x,y)$ is a uniform parametric error bound in a neighborhood of $(1,1)$. Indeed, it is easy to see that $(1,1)$ is also the unique solution of the penalized problem

$$\text{s.t. } y \in [0,2x], x \in [0,2]$$

for any $\mu > 0$.

We now slightly modify the above example to show that the strict complementarity conditions cannot be omitted from Proposition 3.9.

Example 5.4. Consider the same problem in Example 5.3 with constraints $y, z \in [0,2x]$ replaced by $y, z \in [0,x]$ and with $x \in [0,2]$ replaced by $x \in [0,\infty)$.

Again, one can check that $(1,1)$ is the only solution to the problem. However, the gap function is different. In fact, in this example,

$$\begin{aligned} G_0(x,y) &= \max_{z\in[0,x]} \langle y-x, y-z\rangle \\ &= y^2 - xy + \max_{z\in[0,x]} \langle y-x, -z\rangle \\ &= y^2 - xy + x^2 - xy \\ &= (y-x)^2. \end{aligned}$$

Thus the equivalent single level problem involving the standard gap function is

$$\begin{array}{ll} \text{minimize} & x^2 - 2y \\ \text{subject to} & (y-x)^2 = 0, \\ & y \in [0,x], x \in [0,\infty). \end{array}$$

Again, $(y-x)^2$ is not an exact penalty term. This is due to the fact that the strict complementarity condition does not hold at (1,1).

$F(x,y) = y - x$ is strongly monotone; therefore, it is pseudostrongly monotone with respect to y uniformly for all $x \in R^n$. Using Propositions 3.12, 3.13, and 3.15, the problem has the square root standard gap bound, the square root differentiable gap bound, and the projection bound. The differentiable gap function associated with $\alpha = 1$ and $M = I$ takes the form

$$\begin{aligned} G_1(x,y) &= \max_{z\in[0,x]} \left\{ \langle y-x, y-z\rangle - \frac{1}{2}(y-z)^2 \right\} \\ &= \frac{1}{2}(y-x)^2. \end{aligned}$$

The projection bound takes the form $|h(x,y)| = |y-x|$. Indeed, the original problem is equivalent to the following penalized problem:

$$\text{s.t. } y \in [0,x], x \in [0,\infty),$$

for all $\mu > 0$.

Note that the uniform parametric error bounds for Example 5.4 are all Lipschitz continuous. We now give an example which has a square root standard gap bound that is not Lipschitz continuous.

Example 5.5. Consider the following classical bilevel programming problem:

$$\begin{aligned} &\min (x-1)^2 + x^2(y+1)^2 \\ &\text{s.t. } x \in [-1,1] \text{ and } y \in \arg\min \left\{ \left(\sin \frac{\pi}{2} x\right) y : y \in [-1,1] \right\}. \end{aligned}$$

Here $(1,-1)$ is the optimal solution of the problem, and the solution set of the lower level problem is

$$S(x)=\begin{cases}\{1\} & \text{if } -1\le x<0,\\ \text{[-1,1]} & \text{if } x=0,\\ \{-1\} & \text{if } 0<x\le 1.\end{cases}$$

The standard gap function for the problem is

$$\begin{aligned}G_0(x,y)&=\max\left\{\sin\frac{\pi}{2}x\cdot(y-z): z\in[-1,1]\right\}\\ &=\begin{cases}\sin\frac{\pi}{2}x\cdot(y-1) & -1\le x<0,\\ 0 & x=0,\\ \sin\frac{\pi}{2}x\cdot(y+1) & 0<x\le 1.\end{cases}\end{aligned}$$

Since $F(x,y)=\sin\frac{\pi}{2}x$ is independent of y, F is pseudostrongly monotone with respect to y uniformly for all x in a neighborhood of 1. By Proposition 3.12, $\sqrt{G_0(x,y)}$ is an error bound in the neighborhood of $(1,-1)$. However, $\sqrt{G_0(x,y)}$ is not Lipschitz continuous near $(x^*,y^*)=(1,-1)$. Theorem 4.5 cannot be used.

We now verify that the assumptions of Theorem 4.8 are satisfied. The objective function $f(x,y)=(x-1)^2+x^2(y+1)^2$ is upper Hölder continuous near every $y\in S(x)$ uniformly for x in a neighborhood of 1. Since the constraint set $-1\le x\le 1, -1\le y\le 1$ has an interior point, the Slater condition is satisfied. Theorem 4.8 implies that at $(x^*,y^*)=(1,-1)$, there must exist $\mu>0$, $(s_1,s_2,s_3)\ge(0,0,0)$, an integer J, $\lambda_j\ge 0$, $\sum_{j=1}^J\lambda_j=1$, and $\pi_j=(\pi_j^1,\pi_j^2,\pi_j^3)\in R^3$ such that

$$\begin{aligned}&0=2(x^*-1)+2x^*(y^*+1)^2+s_3-\mu\sum_j\lambda_j\pi_j^3,\\ &0=2{x^*}^2(y^*+1)+\mu\sin\left(\frac{\pi}{2}x^*\right)+s_1-s_2,\\ &0=s_1(y^*-1),\\ &0=s_2(-1-y^*),\\ &0=s_3(x^*-1),\\ &0=\sin\left(\frac{\pi}{2}x^*\right)+\pi_j^1-\pi_j^2,\\ &0=\pi_j^1(y^*-1),\\ &0=\pi_j^2(-1-y^*),\\ &0=\pi_j^3(x^*-1).\end{aligned}$$

Indeed, the above condition holds for $J=1$, $\lambda_1=1$, $\mu=s_2=\pi_1^2=1$, and $s_1=s_3=\pi_1^1=\pi_1^3=0$.

6. Exact penalty functions for the KKT formulation. In this section, we assume that $c(x,y)$ is convex and differentiable in y and that one of the usual constraint qualifications holds for the inequality system $c(x,y)\le 0$ in terms of variable y. Under these assumptions, besides formulating GBLP as the single level equivalent problem GS or VS, one can also formulate GBLP as the equivalent single level problem KS. We will show that some of the uniform parametric error bounds such as $G_0(x,y)$, $\sqrt{G_0(x,y)}$, and $g(x,y)-V(x)$ can not only serve as exact penalty terms themselves, but can also play an important role in deriving equivalent exact penalty formulations for KS.

The following results establish the relationships among the KKT, the standard gap, and the value function formulations of GBLP.

PROPOSITION 6.1. *Suppose $c(x, y)$ is convex and differentiable in the y variable. Then*

$$G_0(x, y) \leq -\langle u, c(x, y)\rangle \text{ for all } (x, y, u) \in X \times R^m \times R^d \text{such that} \\ u \geq 0, c(x, y) \leq 0, F(x, y) + \nabla_y c(x, y)^t u = 0.$$

Proof. From mathematical programming weak duality (see, e.g., [20]), one has

$$\begin{aligned} G_0(x, y) &:= \sup\{\langle F(x, y), y - z\rangle : \forall z \in R^m \text{ s.t. } c(x, z) \leq 0\} \\ &= -\inf\{\langle F(x, y), z - y\rangle : \forall z \in R^m \text{ s.t. } c(x, z) \leq 0\} \\ &\leq -\sup\{\langle F(x, y), z - y\rangle + \langle u, c(x, y)\rangle : \forall (z, u) \in R^m \times R^d \text{ s.t.} \\ &\qquad\qquad u \geq 0, c(x, z) \leq 0, \\ &\qquad\qquad F(x, z) + \nabla_y c(x, z)^t u = 0\} \\ &\leq -\sup\{\langle u, c(x, y)\rangle : \forall (x, y, u) \in X \times R^m \times R^d \text{ s.t.} \\ &\qquad\qquad u \geq 0, c(x, y) \leq 0, \\ &\qquad\qquad F(x, y) + \nabla_y c(x, y)^t u = 0\}. \qquad \square \end{aligned}$$

Combining Proposition 6.1 and (1) from Proposition 5.2, we get the following result.

COROLLARY 6.2. *Assume that the objective function $g(x, y)$ for the lower level optimization problem* (3) *and $c(x, y)$ are convex and C^1 in y. Then*

$$g(x, y) - V(x) \leq -\langle u, c(x, y)\rangle \forall (x, y, u) \in X \times R^m \times R^d \text{ such that} \\ u \geq 0, c(x, y) \leq 0, \nabla_y g(x, y) + \nabla c(x, y)^t u = 0.$$

Remark 6.3. Propositions 5.1 and 6.1 and Corollary 6.2 show that any condition ensuring that the standard gap function or $g(x, y) - V(x)$ provides exact penalty terms for the equivalent single level problems GS and VS, respectively, ensure that $-\langle u, c(x, y)\rangle$ is an exact penalty function for the equivalent single level problem KS. The converse is not necessarily true.

Under assumptions involving continuous subanalytic functions, Luo et al. proved in [19] that there exists a constant $N > 0$ such that $(-\langle u, c(x, y)\rangle)^{1/N}$ is an exact penalty term for KS. Moreover, for the case where the mapping $F(x, y)$ is affine and the feasible region is compact, N can be taken as 1 or 2 depending on whether or not the strict complementarity condition is satisfied. To compare our results with those in [19], we summarize the related results in [19].

THEOREM 6.4 (see Theorems 4 and 6 of [19]). *Consider GBLP where the lower level problem is QP_x. Assume that $f(x, y)$ is Lipschitz continuous in both variables and that the set*

$$\{(x, y) \in X \times R^m : Ax + By - b \leq 0\}$$

is a compact polyhedron. Suppose GBLP has a solution. Then there exist positive scalars μ^* and β such that for all scalars $\mu \geq \mu^*$, any vector (x^*, y^*) solves GBLP *if and only if for some $u^* \in R^d$, the triple (x^*, y^*, u^*) solves the following penalized problem in the variables (x, y, u):*

$$\begin{aligned}&\min f(x,y) + \mu\sqrt{-\langle u, Ax+By-b\rangle}\\ &s.t.\ Px+Qy+q+B^Tu=0,\\ &\quad u \geq 0, \|u\| \leq \beta,\\ &\quad Ax+By-b \leq 0,\\ &\quad x \in X, y \in R^m.\end{aligned}$$

Furthermore, if the strict complementarity condition is satisfied for all (x, y, u) in the feasible region of KS, then we can remove the square root.

In the following result we relax most of the assumptions of Theorem 6.4 but we require stronger conditions on $F(x, y)$.

THEOREM 6.5. *Consider GBLP where $f(x, y)$ is Lipschitz continuous in y uniformly in $x \in R^n$ with constant L, and $c(x, y)$ is convex and differentiable in y. Suppose that there exists a solution to GBLP.*

If $F(x, y)$ is pseudostrongly monotone with respect to y uniformly in $x \in X$ with modulus δ, then any vector (x^*, y^*) is a global solution to GBLP if and only if for some $u^* \in R^d$, the triple (x^*, y^*, u^*) is a global solution to the following penalized problem in the variables (x, y, u):

$$\begin{aligned}&\min f(x,y) + \mu\sqrt{-\langle u, c(x,y)\rangle}\\ &s.t.\ F(x,y)+\nabla_y c(x,y)^T u=0,\\ &\quad u \geq 0, c(x,y) \leq 0,\\ &\quad x \in X, y \in R^m,\end{aligned}$$

for all $\mu \geq \frac{\sqrt{\delta}}{\delta}L$.

Under the assumptions of Propositions 3.4 and 3.5, any vector (x^*, y^*) is a global solution to GBLP if and only if for some $u^* \in R^d$, the triple (x^*, y^*, u^*) is a global solution to the following penalized problem in the variables (x, y, u):

$$\begin{aligned}&\min f(x,y) - \delta\mu\langle u, c(x,y)\rangle\\ &s.t.\ F(x,y)+\nabla_y c(x,y)^T u=0,\\ &\quad u \geq 0, c(x,y) \leq 0,\\ &\quad x \in X, y \in R^m,\end{aligned}$$

for all $\mu \geq L$, where δ is the modulus of the uniformly weak sharp minimum.

Proof. We only prove the first assertion, since the proof of the second is similar. Assume (x^*, y^*, u^*) is a global solution of CS. Then (x^*, y^*) is a global solution of GBLP. By Proposition 3.12, $\sqrt{G_0(x, y)}$ is a uniform parametric error bound with modulus $\frac{\sqrt{\delta}}{\delta}$. Therefore, by Theorem 2.6, (x^*, y^*) is a global solution of RP$_{\frac{\sqrt{\delta}}{\delta}\mu}$ with $r(x, y) = \sqrt{G_0(x, y)}$ for all $\mu \geq L$. Therefore,

$$\begin{aligned}f(x^*,y^*) &\leq f(x,y) + \frac{\sqrt{\delta}}{\delta}\mu\sqrt{G_0(x,y)} \qquad \forall x,y \text{ s.t. } c(x,y) \leq 0,\\ &\leq f(x,y) + \frac{\sqrt{\delta}}{\delta}\mu\sqrt{-\langle u, c(x,y)\rangle} \qquad \forall (x,y,u) \text{ s.t.}\\ &\qquad u \geq 0, c(x,y) \leq 0, F(x,y)+\nabla_y c(x,y)^t u = 0,\end{aligned}$$

where the last inequality follows from Proposition 6.1.

The proof of the converse is similar to the converse part in the proof of Proposition 2.2. □

Even when $c(x, y)$ is convex and C^1 in y, the ranges of applications of Theorems 6.4 and 6.5 are different. Indeed, the following example, taken from [19], is a situation where Theorem 6.5 is applicable but Theorem 6.4 is not.

Example 6.6. Consider the problem:

$$\begin{aligned} \text{(P2)} \quad & \min x - y \\ & \text{s.t. } x \geq 0, \text{ and } y \in \arg\min \left\{ \frac{1}{2} y^2 : x + y \geq 0, y \geq 0 \right\}. \end{aligned}$$

In [19], by direct arguments, (P2) is shown to be equivalent to the penalized problem

$$\begin{aligned} \text{(P3)} \quad & \min x - y + \mu \sqrt{u(x+y)} \\ & \text{s.t. } y - u = 0, x + y \geq 0, \\ & \quad (x, y, u) \geq 0, \end{aligned}$$

for any $\mu > 0$. Indeed, it is easy to see that $(0, 0)$ is the unique solution to the problem (P2) and $(0, 0, 0)$ is the unique solution of the penalized problem (P3) for any $\mu > 0$. It is also observed in [19] that Theorem 6.4 is not applicable because the feasible region is not compact. On the other hand, since $F(x, y) = y$ is strongly monotone with respect to y for all $x \in R^n$, this example does satisfy all the conditions of Theorem 6.5. By Theorem 6.5 both $\sqrt{G_0(x, y)}$ and the square root of the complementarity term are exact penalty terms. The standard gap function in this case is $G_0(x, y) = y^2$ for all $x \geq 0$. Therefore, (P2) is equivalent to both (P3) and the following problem:

$$\begin{aligned} \text{(P4)} \quad & \min x - y + \mu |y| \\ & \text{s.t. } x + y \geq 0, x \geq 0, y \geq 0. \end{aligned}$$

Indeed, it is easy to see that $(0, 0)$ is the unique solution of (P4).

Now we discuss an example to which both the KKT and the non-KKT approaches apply, but yield different equivalent single level problems.

Example 6.7. Consider the problem

$$\begin{aligned} \text{(P5)} \quad & \min x + y_1 + y_2 \\ & \text{s.t. } a \leq x \leq b, \\ & \quad (y_1, y_2) \in \arg\min_{y_1, y_2} \left\{ \frac{1}{2} y_1^2 + x y_1 + y_2, : \frac{1}{2} y_1 + x \geq 0, y_1 \geq 0, y_2 \geq 0 \right\}, \end{aligned}$$

where a and b are positive constants. It is obvious that $S(x) = \{(0, 0)\}$ and $V(x) = 0$ for all $x \geq 0$. $F(x, y) = (y_1 + x, 1)$ is not pseudostrongly monotone. Therefore, the assumptions of Propositions 3.9, 3.11, and 3.13 are not satisfied. However, one can verify that the assumptions of Proposition 3.5 are satisfied. Therefore, for any $\mu > 0$, (x^*, y^*) is a solution of the original problem (P5) if and only if it is the solution of the following problem (by the value function approach):

$$\begin{aligned} \text{(P6)} \quad & \min x + y_1 + y_2 + \mu \left(\frac{1}{2} y_1^2 + x y_1 + y_2 \right) \\ & \text{s.t. } \frac{1}{2} y_1 + x \geq 0, a \leq x \leq b, y_1 \geq 0, y_2 \geq 0. \end{aligned}$$

By Theorem 6.5, there exists $u^* \in R^3$ such that (x^*, y^*, u^*) is a solution of the following problem:

$$\begin{aligned}
\text{(P7)} \quad & \min x + y_1 + y_2 + \mu \left(u_1 \left(\frac{1}{2} y_1 + x \right) + u_2 y_1 + u_3 y_2 \right) \\
& \text{s.t. } 0 = y_1 + x - \frac{1}{2} u_1 - u_2, \\
& \qquad 0 = 1 - u_3, \\
& \qquad a \le x \le b, \frac{1}{2} y_1 + x \ge 0, y_1 \ge 0, y_2 \ge 0, u_1 \ge 0, u_2 \ge 0
\end{aligned}$$

for any $\mu > 0$. Clearly, (P5) and (P6) have a unique solution $(a, 0, 0)$, and (P7) has a unique solution $(a, 0, 0, 0, 0, 1)$. Note that the compactness of the feasible region and the strict complementarity assumptions of Theorem 6.4 fail for this example.

Examples 6.6 and 6.7 illustrate that both the KKT and the non-KKT approaches have their advantages and disadvantages. On one hand, by the KKT approach, the exact penalty term is an explicit function of the problem data, but the number of variables in the single level problem increases. On the other hand, by the non-KKT approach, although the number of variables stays the same in the equivalent single level problem, the exact penalty function needs to be computed.

Acknowledgment. We thank J. S. Treiman for his comments on an early version of this paper, which helped to improve the exposition.

REFERENCES

[1] G. Anandalingam and T. L. Friesz, eds., *Hierarchical optimization*, Ann. Oper. Res., 34 (1992).

[2] J. V. Burke, *Calmness and exact penalization*, SIAM J. Control Optim., 29 (1991), pp. 493–497.

[3] J. V. Burke, *An exact penalization viewpoint of constraint optimization*, SIAM J. Control Optim., 29 (1991), pp. 968–998.

[4] J. V. Burke and M. C. Ferris, *Weak sharp minima in mathematical programming*, SIAM J. Control Optim., 31 (1993), pp. 1340–1359.

[5] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Wiley-Interscience, New York, 1983.

[6] S. Dempe, *A necessary and sufficient optimality condition for bilevel programming problems*, Optimization, 25 (1992), pp. 341–354.

[7] J.-P. Dussault and P. Marcotte, *Conditions derégularité géométrique pour les inéquations variationnelles*, RAIRO Rech. Opér., 23 (1988), pp. 1–16.

[8] A. D. Ioffe, *Regular points of Lipschitz functions*, Trans. Amer. Math. Soc., 251 (1979), pp. 61–69.

[9] M. C. Ferris and O. L. Mangasarian, *Minimum principle sufficiency*, Math. Programming, 57 (1992), pp. 1–14.

[10] T. L. Friesz, R. T. Tobin, H.-J. Cho, and N. J. Mehta, *Sensitivity analysis based heuristic algorithms for mathematical programs with variational inequality constraints*, Math. Programming, 48 (1990), pp. 265–284.

[11] M. Fukushima, *Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems*, Math. Programming, 53 (1992), pp. 99–110.

[12] P. T. Harker and J. S. Pang, *On the existence of optimal solutions to mathematical programs with equilibrium constraints*, Oper. Res. Lett., 7 (1988), pp. 61–64.

[13] P. T. Harker and J. S. Pang, *Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications*, Math. Programming, 48 (1990), pp. 161–220.

[14] D. W. Hearn, *The gap function of a convex program*, Oper. Res. Lett., 1 (1982), pp. 67–71.

[15] D. Kinderlehrer and G. Stampacchia, *An Introduction to Variational Inequalities and Their Applications*, Academic Press, New York, 1980.

[16] M. Labbé, P. Marcotte, and G. Savard, *A bilevel model of taxation and its application to optimal highway pricing*, preprint.

[17] P. Loridan and J. Morgan, *A theoretical approximation scheme for Stackelberg problems*, J. Optim. Theory Appl., 11 (1989), pp. 95–110.

[18] X. D. Luo and P. Tseng, *Conditions for a projection-type error bound for the linear complementarity problem to be global*, preprint.

[19] Z. Q. Luo, J. S. Pang, D. Ralph, and S.-Q. Wu, *Exact penalization and stationarity conditions of mathematical programs with equilibrium constraints*, preprint.

[20] O. L. Mangasarian, *Nonlinear Programming*, McGraw-Hill, New York, 1969.

[21] O. L. Mangasarian, *A simple characterization of solution sets of convex programs*, Oper. Res. Lett., 7 (1988), pp. 21–26.

[22] O. L. Mangasarian and R. R. Meyer, *Nonlinear perturbation of linear programs*, SIAM J. Control Optim., 17 (1979), pp. 745–752.

[23] O. L. Mangasarian and J. Ren, *New improved error bounds for the linear complementarity problem*, Math. Programming, 66 (1994), pp. 241–255.

[24] R. Mathias and J. S. Pang, *Error bounds for the linear complementarity problem with a $P-$matrix*, Linear Algebra Appl., 132 (1990), pp. 123–136.

[25] P. Marcotte and D. L. Zhu, *Exact and Inexact Penalty Methods for the Generalized Bilevel Programming Problem*, Publication of Centre de recherche sur les transports, Université de Montréal, Canada, CRT-920, 1992.

[26] J. V. Outrata, *Necessary optimality conditions for Stackelberg problems*, J. Optim. Theory Appl., 76 (1993), pp. 305–320.

[27] J. S. Pang, *A posteriori error bounds for the linearly-constrained variational inequality problem*, Math. Oper. Res., 12 (1987), pp. 474–484.

[28] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[29] H. Von Stackelberg, *The Theory of the Market Economy*, Oxford University Press, Oxford, England, 1952.

[30] J. J. Ye and D. L. Zhu, *Optimality conditions for bilevel programming problems*, Optimization, 33 (1995), pp. 9–27.

[31] R. Zhang, *Problems of hierarchical optimization: Nonsmoothness and analysis of solutions*, Ph.D. thesis, University of Washington, Seattle, 1990.

[32] R. Zhang, *Problems of hierachical optimization in finite dimensions*, SIAM J. Optim., 4 (1994), pp. 521–536.

SIAM J. Optim.
Vol. 7, No. 2, pp. 508–518, May 1997

ON UNIQUENESS OF LAGRANGE MULTIPLIERS IN OPTIMIZATION PROBLEMS SUBJECT TO CONE CONSTRAINTS*

ALEXANDER SHAPIRO†

Abstract. In this paper we study uniqueness of Lagrange multipliers in optimization problems subject to cone constraints. The main tool in our investigation of this question will be a calculus of dual (polar) cones. We give sufficient and in some cases necessary conditions for uniqueness of Lagrange multipliers in general Banach spaces. General results are then applied to two particular examples of the semidefinite and semi-infinite programming problems, respectively.

Key words. Lagrange multipliers, cone constraints, first-order optimality conditions, semidefinite programming, semi-infinite programming

AMS subject classifications. 90C30, 90C34, 90C48

PII. S1052623495279785

1. Introduction.

Consider the following optimization problem:

$$\min_{x \in X} f(x) \quad \text{subject to} \quad g(x) \in K. \tag{1.1}$$

Here X and Y are (real) Banach spaces, $f : X \to \mathbb{R}$ and $g : X \to Y$ are continuously differentiable functions, $K \subset Y$ is a convex closed cone, and

$$L(x, \lambda) = f(x) + \langle \lambda, g(x) \rangle$$

is the Lagrangian function. The first-order necessary conditions for a feasible point x_0 to be a locally optimal solution of the above problem can be written as follows (see [6, 9, 10]). Under a constraint qualification there exists $\lambda \in K^-$ such that

$$D_x L(x_0, \lambda) = 0, \tag{1.2}$$

$$\langle \lambda, g(x_0) \rangle = 0. \tag{1.3}$$

In this paper we discuss uniqueness of Lagrange multipliers satisfying the first-order necessary conditions. The question of uniqueness of Lagrange multipliers arises naturally, for example, in sensitivity analysis of optimization problems (see, e.g., [7, 13]) and in convergence analysis of Newton type optimization algorithms (cf. [2]). In case the space Y is finite dimensional and the cone K is polyhedral, there are reasonably simple necessary and sufficient conditions ensuring uniqueness of Lagrange multipliers [5]. The situation is considerably more subtle in the general case of cone constraints.

The main tool in our investigation of this question will be a calculus of dual cones. For the reader's convenience and in order to make the paper self contained we describe in the remainder of this section a few required facts from the theory of dual cones. We view the Banach space Y and its dual Y^* as paired spaces. By $\langle \alpha, y \rangle$ we denote the value $\alpha(y)$ of a continuous linear functional $\alpha \in Y^*$. We consider $\langle \cdot, \cdot \rangle$ as a bilinear form on $Y^* \times Y$ and equip Y and Y^* with a pair of compatible topologies. That is, for

* Received by the editors January 11, 1995; accepted for publication (in revised form) September 26, 1995.
http://www.siam.org/journals/siopt/7-2/27978.html

† School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205 (ashapiro@isye.gatech.edu).

every $\alpha \in Y^*$ the linear functional $\langle \alpha, \cdot \rangle$ is continuous in the considered topology of Y, and all continuous linear functionals on Y can be represented in such form. Similarly, all linear, continuous in the considered topology of Y^* functionals can be represented in the form $\langle \cdot, v \rangle$ for some $v \in Y$. The pair of compatible topologies that we use in this paper will be the norm topology of Y and the weak star topology (w^*-topology) of Y^*.

For a cone $C \subset Y$, its polar (negative dual) cone C^- is defined as follows:

$$C^- = \{\alpha \in Y^* : \ \langle \alpha, y \rangle \leq 0 \ \text{ for all } y \in C\}.$$

Similarly, for a cone $\Sigma \subset Y^*$, its polar cone is given by

$$\Sigma^- = \{y \in Y : \ \langle \alpha, y \rangle \leq 0 \ \text{ for all } \alpha \in \Sigma\}.$$

Note that the polar cones C^- and Σ^- are always convex and closed in the considered compatible topologies; i.e., C^- is closed in the w^*-topology of Y^* and Σ^- is closed in the norm topology of Y. If C is a linear space, then C^- coincides with the orthogonal complement $C^\perp$ of C. In particular, if $\lambda \in Y^*$, then $[\lambda]^- = [\lambda]^\perp = \text{Ker}\lambda$, where $\text{Ker}\lambda$ is the null space of λ and $[\lambda]$ denotes the (one-dimensional) space generated by λ.

It follows from the Hahn–Banach theorem that if the cone $C \subset Y$ is convex, then $(C^-)^- = \text{cl}\{C\}$, where $\text{cl}\{C\}$ denotes the topological closure, in the norm topology of Y, of the cone C (e.g., [1, Chapter 1, section 5]). Similarly, if the cone $\Sigma \subset Y^*$ is convex, then $(\Sigma^-)^- = \text{cl}^*\{\Sigma\}$, where $\text{cl}^*\{\Sigma\}$ denotes the topological closure of Σ in the w^*-topology of Y^*. Note that if the space Y is reflexive and Σ is convex, then $\text{cl}^*\{\Sigma\} = \text{cl}\{\Sigma\}$.

It is straightforward to verify (cf. [1]) that if C_1 and C_2 are two cones in Y or in Y^*, then

$$(C_1 + C_2)^- = C_1^- \cap C_2^-. \tag{1.4}$$

It follows from (1.4) that the polar of the cone $C_1^- \cap C_2^-$ coincides with the polar of $(C_1 + C_2)^-$. Consequently, if the cones C_1 and C_2 are convex, then the polar cone of $C_1^- \cap C_2^-$ is given by the topological closure of the cone $C_1 + C_2$. Denote $K_1 = C_1^-$ and $K_2 = C_2^-$. It follows that $(K_1 \cap K_2)^-$ coincides with the topological closure of the cone $K_1^- + K_2^-$. Since any convex closed cone can be represented as the polar cone, we obtain that if K_1 and K_2 are two convex cones in Y or Y^*, closed in the respective compatible topology, then

$$(K_1 \cap K_2)^- = \begin{cases} \text{cl}^*\{K_1^- + K_2^-\} & \text{if } K_1, K_2 \subset Y, \\ \text{cl}\{K_1^- + K_2^-\} & \text{if } K_1, K_2 \subset Y^*. \end{cases} \tag{1.5}$$

(See, e.g., [3] for details.)

Now let S be a convex set in Y or Y^* and $v \in S$. We denote by $\mathcal{R}(S, v)$ the *radial* cone of S at v. That is, $\mathcal{R}(S, v)$ is the cone generated by the set $S - v$ or (equivalently) is the set formed by such vectors u that $v + tu \in S$ for some $t > 0$. If S is a convex cone, then $\mathcal{R}(S, v) = S + [v]$, where $[v]$ denotes the one-dimensional linear space generated by vector v. The topological closure in the norm topology of $\mathcal{R}(S, v)$ is called the *tangent* cone to S at v and denoted $T(S, v)$. When $S \subset Y^*$, we also consider $T^*(S, v) = \text{cl}^*\{\mathcal{R}(S, v)\}$. Since the radial cone of a convex set is convex, we have that if Y is reflexive, then $T^*(S, v) = T(S, v)$. If S is a convex cone closed in the respective compatible topology, we have that

$$T^*(S, v)^- = T(S, v)^- = \mathcal{R}(S, v)^- = (S + [v])^- = S^- \cap [v]^\perp.$$

Let $A : X \to Y$ be a continuous linear operator. Its adjoint operator $A^* : Y^* \to X^*$ is defined by the relation

$$\langle A^*\lambda, x\rangle = \langle \lambda Ax\rangle \text{ for all } x \in X \text{ and } \lambda \in Y^*.$$

Note that it follows from the above definition that $A^*\lambda = 0$ iff $\langle \lambda, Ax\rangle = 0$ for all $x \in X$. Therefore $\mathrm{Ker}A^* = (AX)^\perp$.

For a convex set $S \subset Y$ we denote by $\mathrm{int}(S)$, $\mathrm{lin}(S)$, and $\mathrm{ri}(S)$ its interior, the linear space generated by S, and its relative interior, respectively. That is, $\mathrm{lin}(S)$ is the intersection of all linear subspaces which contain S and $\mathrm{ri}(S)$ is the interior of S relative to $\mathrm{lin}(S)$; i.e., $y \in \mathrm{ri}(S)$ iff $y \in S$ and there is a neighborhood N (in the norm topology of Y) of y such that $N \cap \mathrm{lin}(S) \subset S$.

Proposition 1.1. *Let Y be a normed space, $C \subset Y$ be a convex cone with a nonempty interior, and L be a linear subspace of Y. Then* $\mathrm{cl}\{L + C\} = Y$ *if and only if* $L \cap \mathrm{int}(C) \neq \emptyset$.

Proof. Suppose that $L \cap \mathrm{int}(C) \neq \emptyset$. This means that there exist $y \in L$ and a ball $B \subset Y$ of radius $r > 0$ and centered at zero such that $y + B \subset C$. We have then that $B = (-y) + y + B \subset L + C$, and since $L + C$ is a cone, it follows that $tB \subset L + C$ for any $t \geq 0$. This implies that $L + C = Y$.

Conversely, suppose that $L \cap \mathrm{int}(C) = \emptyset$. Then by a separation theorem (e.g., [4, p. 163]) there exists $\alpha \in Y^*$, $\alpha \neq 0$ such that $\langle \alpha, y\rangle = 0$ for any $y \in L$ and $\langle \alpha, y\rangle \leq 0$ for any $y \in C$. It follows that $\alpha \in (L + C)^-$ and hence $\mathrm{cl}\{L + C\} \subset \{y : \langle \alpha, y\rangle \leq 0\} \neq Y$. □

2. Basic results. Let $\lambda_0 \in K^-$ be a Lagrange multiplier satisfying optimality conditions (1.2) and (1.3). In this section we discuss general conditions for uniqueness of this Lagrange multiplier. Consider the set

$$C = \{\lambda \in K^- : \langle \lambda, g(x_0)\rangle = 0\}.$$

Note that C is a convex cone, closed in the w^*-topology of Y^*, and that $\lambda_0 \in C$. Moreover, by (1.5), $C^- = \mathrm{cl}\{K + [g(x_0)]\}$ and hence $C^- = T(K, g(x_0))$.

Proposition 2.1. *The Lagrange multiplier λ_0 is unique if and only if*

$$\mathcal{R}(C, \lambda_0) \cap [Dg(x_0)X]^\perp = \{0\}. \tag{2.1}$$

Proof. Consider a vector $\lambda \in K^-$ and let $\mu = \lambda - \lambda_0$. We have that λ satisfies (1.2) iff $[Dg(x_0)]^*\mu = 0$, and λ satisfies (1.3) iff $\lambda_0 + \mu \in C$. Therefore λ can be a Lagrange multiplier different from λ_0 iff there exists a nonzero vector $\mu \in Y^*$ such that $\mu \in [Dg(x_0)X]^\perp$ and $\mu \in \mathcal{R}(C, \lambda_0)$. □

In the following theorem we give sufficient, and in some cases necessary, conditions for uniqueness of λ_0 which can be viewed as dual to (2.1).

Theorem 2.2. *The following condition is sufficient for uniqueness of λ_0:*

$$\mathrm{cl}\{Dg(x_0)X + T(K, g(x_0)) \cap \mathrm{Ker}\lambda_0\} = Y. \tag{2.2}$$

If $\mathcal{R}(C, \lambda_0) = T^*(C, \lambda_0)$, then condition (2.2) *is also necessary.*

Proof. Consider the cone

$$Q = Dg(x_0)X + T(K, g(x_0)) \cap \mathrm{Ker}\lambda_0.$$

Its polar cone is given by

$$Q^- = [Dg(x_0)X]^- \cap [T(K, g(x_0)) \cap \mathrm{Ker}\lambda_0]^-.$$

Moreover, we have that $[Dg(x_0)X]^- = [Dg(x_0)X]^\perp$ and, by (1.5),

$$[T(K, g(x_0)) \cap \mathrm{Ker}\lambda_0]^- = \mathrm{cl}^*\{[T(K, g(x_0))]^- + [\lambda_0]\} = \mathrm{cl}^*\{C + [\lambda_0]\} = T^*(C, \lambda_0).$$

Therefore,

$$Q^- = [Dg(x_0)X]^\perp \cap T^*(C, \lambda_0). \tag{2.3}$$

Suppose now that condition (2.2) holds. Then $Q^- = \{0\}$ and since $\mathcal{R}(C, \lambda_0) \subset T^*(C, \lambda_0)$, condition (2.1) follows from (2.3). Moreover, if $\mathcal{R}(C, \lambda_0) = T^*(C, \lambda_0)$, then by (2.3) condition (2.1) is equivalent to $Q^- = \{0\}$, which in turn is equivalent to (2.2). □

Consider now the cone $K_0 = K \cap \mathrm{Ker}\lambda_0$. We have that $T(K_0, g(x_0)) \subset T(K, g(x_0)) \cap \mathrm{Ker}\lambda_0$ and hence it follows from Theorem 2.2 that the condition

$$Dg(x_0)X + T(K_0, g(x_0)) = Y \tag{2.4}$$

is sufficient for uniqueness of λ_0. Condition (2.4) is equivalent to a constraint qualification, with respect to the cone K_0, in the sense of Robinson [11]. Its sufficiency for uniqueness of λ_0 was discussed in [12]. If the space Y is finite dimensional and the cone K is polyhedral, the condition (2.4) is also necessary (cf. [5]).

Let us remark that since $[Dg(x_0)X]^\perp = \mathrm{Ker}[Dg(x_0)]^*$, condition (2.1) is equivalent to

$$\{\mu \in \mathcal{R}(C, \lambda_0) : [Dg(x_0)]^*\mu = 0\} = \{0\}. \tag{2.5}$$

Similarly and because of (2.3), condition (2.2) is equivalent to

$$\{\mu \in T^*(C, \lambda_0) : [Dg(x_0)]^*\mu = 0\} = \{0\}. \tag{2.6}$$

In some applications it will be convenient to formulate the sufficient condition (2.2) of Theorem 2.2 in the following form.

PROPOSITION 2.3. *Let $\mathcal{L}$ be a linear space generated by the cone $\mathcal{T} = T(K, g(x_0)) \cap \mathrm{Ker}\lambda_0$ and suppose that $\mathcal{T}$ has a nonempty relative interior (relative to $\mathcal{L}$). Then condition* (2.2) *holds if the following two conditions are satisfied:*

(i) $\mathrm{cl}\{Dg(x_0)X + \mathcal{L}\} = Y$*, and*

(ii) *there exists a vector $h \in X$ such that $Dg(x_0)h \in \mathrm{ri}(\mathcal{T})$.*

Conversely, if condition (2.2) *holds and $\mathcal{L} \subset Dg(x_0)X + \mathcal{T}$, then conditions* (i) *and* (ii) *follow.*

Proof. Suppose that the above conditions (i) and (ii) are satisfied. By Proposition 1.1 it follows from condition (ii) that $\mathcal{L} \subset Dg(x_0)X + \mathcal{T}$. Together with condition (i) this implies that $\mathrm{cl}\{Dg(x_0)X + \mathcal{T}\} = Y$, meaning that condition (2.2) holds.

Conversely, let us suppose that condition (2.2) holds. Since $\mathrm{cl}\{Dg(x_0)X + \mathcal{T}\} \subset \mathrm{cl}\{Dg(x_0)X + \mathcal{L}\}$, condition (i) then follows. Also, we have that $\mathcal{L} \subset Dg(x_0)X + \mathcal{T}$ and, since $\mathcal{T} \subset \mathcal{L}$, we obtain that $\mathcal{L} = \mathcal{M} + \mathcal{T}$, where $\mathcal{M} = \mathcal{L} \cap Dg(x_0)X$. By Proposition 1.1, condition (ii) then follows. □

By Theorem 2.2 we obtain then that conditions (i) and (ii) of Proposition 2.3 are sufficient for uniqueness of λ_0. Note that if $Dg(x_0)X + \mathcal{T}$ is closed, then the condition $\mathcal{L} \subset Dg(x_0)X + \mathcal{T}$ follows from condition (2.2). In that case conditions (i) and (ii) are equivalent to condition (2.2).

3. Examples and applications. In this section we discuss two examples of semidefinite and semi-infinite programming. Let us start with the example of semidefinite programming. Let $X = \mathbb{R}^m$ and $Y = \mathcal{S}_n$, where $\mathcal{S}_n$ denotes the space of an $n \times n$ symmetric matrix. We equip $\mathbb{R}^m$ with the standard scalar product $x \cdot y = \sum_{i=1}^m x_i y_i$ and $\mathcal{S}_n$ with the scalar product $A \bullet B = \mathrm{tr} AB$ for any $A, B \in \mathcal{S}_n$. The spaces X and Y can be then identified with their duals X^* and Y^*, respectively. In the space $\mathcal{S}_n$ we consider the cone K of positive semidefinite matrices, i.e., $K = \{A \in \mathcal{S}_n : A \succeq 0\}$. The cone K is convex and closed and its polar cone K^- is formed by negative semidefinite matrices, i.e., $K^- = \{\Omega \in \mathcal{S}_n : \Omega \preceq 0\}$. In what follows we denote by E^T the transpose of a matrix E.

Let $f : \mathbb{R}^m \to \mathbb{R}$ and $G : \mathbb{R}^m \to \mathcal{S}_n$ be continuously differentiable functions, $L(x, \Lambda) = f(x) + \Lambda \bullet G(x)$, and let $x_0 \in \mathbb{R}^m$ be a point satisfying the corresponding first-order optimality conditions. That is, $G(x_0) \in K$ and there exists a matrix $\Lambda_0 \in K^-$ such that

$$D_x L(x_0, \Lambda_0) = 0, \tag{3.1}$$

$$\Lambda_0[G(x_0)] = 0. \tag{3.2}$$

Note that since $G(x_0) \succeq 0$ and $\Lambda_0 \preceq 0$, condition (3.2) is equivalent to the complementarity condition $\Lambda_0 \bullet G(x_0) = 0$.

Let $r = \mathrm{rank} G(x_0)$ and let E be an $n \times (n-r)$ matrix of full column rank $n - r$ such that $G(x_0)E = 0$. Then it is not difficult to show (cf. [15]) that the tangent cone to K at $G(x_0)$ can be written in the form

$$T(K, G(x_0)) = \{Z \in \mathcal{S}_n : E^T Z E \succeq 0\}. \tag{3.3}$$

We also have that the cone $C = \{\Lambda \in K^- : \Lambda \bullet G(x_0) = 0\}$ is given by

$$C = \{E \Theta E^T : \Theta \in \mathcal{S}_{n-r},\ \Theta \preceq 0\}. \tag{3.4}$$

We say that the *strict complementarity* condition holds if

$$\mathrm{rank}\, \Lambda_0 + \mathrm{rank} G(x_0) = n. \tag{3.5}$$

The Lagrange multipliers matrix Λ_0 belongs to the cone C and hence can be represented in the form $\Lambda_0 = E \Theta_0 E^T$ for some $(n-r) \times (n-r)$ symmetric, negative semidefinite matrix Θ_0. The strict complementarity condition (3.5) means that the matrix Θ_0 is nonsingular and hence is negative definite.

Under the strict complementarity condition the radial cone $\mathcal{R}(C, \Lambda_0)$ coincides with the tangent cone $T(C, \Lambda_0)$ and is given by the linear space $\{\Omega \in \mathcal{S}_n : \Omega = E \Theta E^T : \Theta \in \mathcal{S}_{n-r}\}$. Furthermore,

$$\mathcal{R}(C, \Lambda_0)^- = \mathcal{R}(C, \Lambda_0)^\perp = T(K, G(x_0)) \cap \mathrm{Ker} \Lambda_0 = LT(K, G(x_0)),$$

where

$$LT(K, G(x_0)) = \{Z \in \mathcal{S}_n : E^T Z E = 0\}$$

is the lineality space of the cone $T(K, G(x_0))$. Therefore we obtain from Theorem 2.2 that, under the strict complementarity condition, the Lagrange multipliers matrix Λ_0 is unique iff

$$DG(x_0)\mathbb{R}^m + LT(K, G(x_0)) = \mathcal{S}_n. \tag{3.6}$$

Equation (3.6) represents a necessary and sufficient condition for a transversality relation between the mapping G and the manifold of symmetric $n \times n$ matrices of rank r (cf. [15]). It can be written in an equivalent form as follows. The adjoint $[DG(x_0)]^* : \mathcal{S}_n \to \mathbb{R}^m$ of $DG(x_0)$ is given by

$$[DG(x_0)]^*\Omega = (\Omega\bullet G_1(x_0), \dots, \Omega\bullet G_m(x_0)), \ \ \Omega \in \mathcal{S}_n,$$

where $G_i(x_0) = \partial G(x_0)/\partial x_i$ are the $n \times n$ partial derivatives matrices of $G(x)$ at $x = x_0$. Therefore, by using (2.6), we have that (3.6) is equivalent to the condition that the m-dimensional vectors $v_{ij} = (e_i^T G_1(x_0)e_j, \dots, e_i^T G_m(x_0)e_j)$, $1 \le i \le j \le n-r$, are linearly independent. Here $e_1, \dots, e_{n-r}$ are the column vectors of the matrix E.

Suppose now that $\operatorname{rank} \Theta_0 = q < n - r$. Let $\Theta_0 = V\Phi_0 V^T$ be the spectral decomposition of Θ_0; i.e., V is an $(n-r) \times q$ matrix such that $V^TV = I_q$ and Φ_0 is a $q \times q$ negative definite (diagonal) matrix. Let U be an orthogonal complement of V, i.e., U is an $(n-r) \times (n-r-q)$ matrix such that $U^TV = 0$ and $U^TU = I_{n-r-q}$, and consider the matrices $E_1 = EV$ and $E_2 = EU$ and the cone $\mathcal{T} = T(K, G(x_0)) \cap \mathrm{Ker}\Lambda_0$. We have then that

$$\mathcal{T} = \{Z \in \mathcal{S}_n : E^TZE \succeq 0, \ E_1^TZE_1 = 0\}.$$

Note that the column space generated by the $n \times (n-r)$ matrix $[E_1, E_2]$ is the same as the column space generated by the matrix E. Therefore we can write the cone $\mathcal{T}$ in the form

$$\mathcal{T} = \{Z \in \mathcal{S}_n : E_1^TZE_1 = 0, \ E_1^TZE_2 = 0, \ E_2^TZE_2 \succeq 0\}.$$

The linear space $\mathcal{L}$, generated by the cone $\mathcal{T}$, is then given by

$$\mathcal{L} = \{Z \in \mathcal{S}_n : E_1^TZE_1 = 0, \ E_1^TZE_2 = 0\}$$

and the relative interior of $\mathcal{T}$ is

$$\mathrm{ri}(\mathcal{T}) = \{Z \in \mathcal{S}_n : E_1^TZE_1 = 0, \ E_1^TZE_2 = 0, \ E_2^TZE_2 \succ 0\}.$$

We now can employ conditions (i) and (ii) of Proposition 2.3 in order to derive sufficient conditions for uniqueness of the Lagrange multipliers matrix Λ_0. Let $\bar{e}_1, \dots, \bar{e}_{n-r}$ be the column vectors of the matrix $[E_1, E_2]$; i.e., $\bar{e}_1, \dots, \bar{e}_q$ are the column vectors of E_1 and $\bar{e}_{q+1}, \dots, \bar{e}_{n-r}$ are the column vectors of E_2, and consider the m-dimensional vectors $\bar{v}_{ij} = (\bar{e}_i^T G_1(x_0)\bar{e}_j, \dots, \bar{e}_i^T G_m(x_0)\bar{e}_j)$, $i, j = 1, \dots, n-r$. Then, in the present situation, conditions (i) and (ii) are equivalent to the following conditions and hence, by Theorem 2.2, are sufficient for uniqueness of Λ_0.

PROPOSITION 3.1. *The following two conditions are sufficient for uniqueness of the Lagrange multipliers matrix* Λ_0.

(i$'$) *Vectors* $\bar{v}_{ij}$, $(i,j) \in \mathcal{I}$, *where*

$$\mathcal{I} = \{(i,j) : i,j = 1, \dots, q, \ \ i \le j\} \cup \{(i,j) : i = 1, \dots, q, \ j = q+1, \dots, n-r\},$$

are linearly independent.

(ii$'$) *There exists a vector* $h \in \mathbb{R}^m$ *such that* $h \cdot \bar{v}_{ij} = 0$, $(i,j) \in \mathcal{I}$, *and*

$$\sum_{k=1}^{m} h_k E_2^T G_k(x_0) E_2 \succ 0. \tag{3.7}$$

In a sense conditions (i′) and (ii′) can be viewed as an analog of the strong Mangasarian–Fromovitz constraint qualification used in [5] for nonlinear programming problems.

Let us discuss now the example of semi-infinite programming. Consider the following optimization problem:

$$\min_{x\in\mathbb{R}^m} f(x) \quad \text{subject to } \ h(x,t)\le 0,\ t\in T, \tag{3.8}$$

where $f:\mathbb{R}^m\to\mathbb{R}$, $h:\mathbb{R}^m\times T\to\mathbb{R}$ and T is a compact metric space. We assume that $f(\cdot)$ and $h(\cdot,t)$ for all $t\in T$ are continuously differentiable and that $h(x,t)$ and $\nabla h(x,t)$ are continuous on $\mathbb{R}^m\times T$. (The gradient $\nabla h(x,t)$ is taken with respect to x.)

In order to formulate the inequality constraints of the semi-infinite program (3.8) in a form of cone constraints, we proceed as follows. Consider the space $C(T)$ of continuous functions $y:T\to\mathbb{R}$, equipped with the sup-norm $\|y\|=\sup_{t\in T}|y(t)|$, and the cone

$$K=\{y\in C(T): y(t)\le 0,\ t\in T\}$$

formed by nonpositive valued continuous functions. Consider also the mapping $g:\mathbb{R}^m\to C(T)$ taking a point $x\in\mathbb{R}^m$ into the function $y=g(x)$, $y(\cdot)=h(x,\cdot)$. Then the feasible set of the program (3.8) can be defined by the cone constraint $g(x)\in K$. Note that under the above assumptions the mapping g is continuously differentiable and $[Dg(x)v](\cdot)=v\cdot\nabla h(x,\cdot)$.

The dual space Y^* of the Banach space $Y=C(T)$ is the space of finite signed measures on $(T,\mathcal{B})$, where $\mathcal{B}$ is the Borel σ-algebra of T, with the norm given by the total variation of the corresponding measure, and $\langle\lambda,y\rangle=\int_T y(t)\lambda(dt)$, $\lambda\in Y^*$, $y\in Y$. The polar cone K^- of the cone K is formed by the set of (nonnegative) Borel measures on T. For a feasible point x (satisfying $g(x)\in K$), denote by $\Delta(x)$ the set

$$\Delta(x)=\{t\in T: h(x,t)=0\}$$

of active-at-x constraints. Then the tangent cone to K at $g(x)$ can be written in the form (e.g., [14])

$$T(K,g(x))=\{y\in C(T): y(t)\le 0 \ \text{ for all } t\in\Delta(x)\}. \tag{3.9}$$

Let x_0 be a locally optimal solution of (3.8). Suppose that there exists a vector $v\in\mathbb{R}^m$ such that

$$v\cdot\nabla h(x_0,t)<0 \ \text{ for all } \ t\in\Delta(x_0). \tag{3.10}$$

In case the set T is finite, this is the Mangasarian–Fromovitz constraint qualification [8]. In the case of semi-infinite programming this condition is equivalent (e.g., [14]) to regularity of x_0 (with respect to the mapping g and the cone K) in the sense of Robinson [10].

Under the constraint qualification (3.10), x_0 corresponds with a Lagrange multiplier $\mu\in K^-$, satisfying the first-order optimality conditions, and the set of such Lagrange multipliers is bounded in the norm topology of Y^* (e.g., [9]). In the present case of semi-infinite programming, $\mu\in K^-$ is a measure and the first-order optimality conditions (1.2) and (1.3) take the form

$$\nabla f(x_0)+\int_T\nabla h(x_0,t)\mu(dt)=0, \tag{3.11}$$

and the support of the measure μ is contained in the set $\Delta(x_0)$. Moreover, the measure μ can be chosen to be a discrete measure. That is, there are points $t_i \in \Delta(x_0)$ and numbers $\lambda_i > 0$, $i = 1, \dots, n$ such that $\mu = \sum_{i=1}^n \lambda_i \delta(t_i)$, where $\delta(t)$ denotes the measure of mass one at the point t. The optimality condition (3.11) then takes the form

$$\nabla f(x_0) + \sum_{i=1}^{n} \lambda_i \nabla h(x_0, t_i) = 0. \tag{3.12}$$

It is not difficult to show that if a measure μ is not discrete, then it cannot be an extreme point of the set of Lagrange multipliers measures and hence cannot be unique (e.g., [14, p. 750]). Therefore, we assume subsequently that $\mu = \sum_{i=1}^n \lambda_i \delta(t_i)$ is a discrete measure satisfying the first-order optimality conditions.

The cone $\mathcal{T} = T(K, g(x_0)) \cap \operatorname{Ker} \mu$ can be written here in the form

$$\mathcal{T} = \{y \in C(T) : y(t) \leq 0 \ \text{ for all } t \in \Delta(x_0),\ y(t_i) = 0,\ i = 1, \dots, n\}. \tag{3.13}$$

The linear space $\mathcal{L}$ generated by the cone $\mathcal{T}$ is given then by

$$\mathcal{L} = \{y \in C(T) : y(t_i) = 0,\ i = 1, \dots, n\}.$$

Let us observe that it is possible that the relative interior of the cone $\mathcal{T}$ (relative to the space $\mathcal{L}$) is empty. This can happen if the points $t_1, \dots, t_n$ are not isolated points of the set $\Delta(x_0)$. Consider, for example, $T = [0, 1]$ and let $h(x_0, t) = 0$ for all $t \in [0, 1]$; i.e., $\Delta(x_0) = [0, 1]$, and let $t_1 = 1/2$, $n = 1$. Then it is not difficult to see that for any function $y(\cdot)$ in $\mathcal{T}$, one can find a function $\bar{y}(\cdot)$ in $\mathcal{L}$, arbitrarily close to $y(\cdot)$ in the sup-norm topology and such that $\bar{y}(t) > 0$ for some t sufficiently close to $1/2$.

This shows that in general we cannot apply here the sufficient conditions of Proposition 2.3. Therefore we work directly with condition (2.2) of Theorem 2.2.

Proposition 3.2. *The following two conditions are necessary and sufficient for uniqueness of the Lagrange multipliers measure* $\mu = \sum_{i=1}^n \lambda_i \delta(t_i)$.

(i$''$) *The gradient vectors* $\nabla h(x_0, t_i)$, $i = 1, \dots, n$ *are linearly independent.*

(ii$''$) *For any neighborhood* N *of the set* $\{t_1, \dots, t_n\}$ *there exists* $v \in \mathbb{R}^m$ *such that*

$$v \cdot \nabla h(x_0, t_i) = 0, \quad i = 1, \dots, n, \tag{3.14}$$

$$v \cdot \nabla h(x_0, t) < 0, \quad t \in \Delta(x_0) \setminus N. \tag{3.15}$$

Proof. Let us first show that if the set of Lagrange multipliers measures is not a singleton, then it contains at least two different *discrete* measures. We argue as follows. Consider the set Γ of measures $\gamma \in K^-$, whose support is contained in the set $\Delta(x_0)$ and such that $\|\gamma\|^* \leq 1$ and

$$c \nabla f(x_0) + \int \nabla h(x_0, t) \gamma(dt) = 0 \tag{3.16}$$

for some $c \geq 0$. Here $\|\cdot\|^*$ denotes the total variation norm on the space Y^*. For a nonnegative measure $\gamma \in K^-$, we have that $\|\gamma\|^* = \gamma(T)$. Clearly, if $\gamma \in \Gamma$ and the corresponding coefficient c in (3.16) is not zero, then $c^{-1}\gamma$ is a Lagrange multipliers measure. Conversely, if λ is a nonzero Lagrange multipliers measure, then $\lambda / \|\lambda\|^* \in \Gamma$. It is not difficult to see that Γ is convex, bounded and closed in the w^*-topology subset of Y^*, and hence is w^*-compact. By the Krein–Millman theorem it follows then that Γ

coincides with the closure (in the w^*-topology) of the convex hull of its extreme points. In order to complete the arguments it will be sufficient to show now that if a measure γ is an extreme point of Γ, then it is discrete. Consider a nondiscrete, nonzero measure $\gamma \in \Gamma$. Then $\gamma = \gamma_1 + \cdots + \gamma_{m+2}$, where γ_i, $i = 1, \dots, m+2$ are positive measures with disjoint supports. Consider vectors $b_i = \int \nabla h(x_0, t)\gamma_i(dt)$, $i = 1, \dots, m+2$. By dimensionality arguments there exist numbers a_i, $i = 1, \dots, m+2$, not all of them zeros, such that $|a_i| < 1$, $\sum_{i=1}^{m+2} a_i b_i = 0$ and $\sum_{i=1}^{m+2} a_i \gamma_i(T) = 0$. Consider the measures $\gamma' = \sum_{i=1}^{m+2}(1 - a_i)\gamma_i$ and $\gamma'' = \sum_{i=1}^{m+2}(1 + a_i)\gamma_i$. Clearly $\gamma', \gamma'' \in \Gamma$ and $\gamma = (\gamma' + \gamma'')/2$. Therefore γ cannot be an extreme point of Γ.

Suppose that conditions (i$''$) and (ii$''$) hold. Because of the above arguments, in order to verify uniqueness of μ it will be sufficient to show that if $\alpha \in [Dg(x_0)\mathbb{R}^m + \mathcal{T}]^-$ and α is discrete, then $\alpha = 0$. Let $\alpha \in [Dg(x_0)\mathbb{R}^m + \mathcal{T}]^-$ be a discrete measure and let S be a *finite* subset of T containing the support of α and the set $\{t_1, \dots, t_n\}$. We can write then $\alpha = \sum_{t \in S} \alpha(t)\delta(t)$, where $\alpha(t)$ is a nonnegative valued function on the set S. Consider a function $z \in C(T)$. Because of the condition (i$''$), there exists a vector $u \in \mathbb{R}^m$ such that $u \cdot \nabla h(x_0, t_i) = z(t_i)$, $i = 1, \dots, n$. Choose a neighborhood N of the set $\{t_1, \dots, t_n\}$ which does not contain other points of the set S. Then, because of the assumption (ii$''$), there exists a vector v satisfying condition (3.14) and such that $v \cdot \nabla h(x_0, t) < -c$ for all $t \in S \setminus \{t_1, \dots, t_n\}$ and some $c > 0$. Let τ be a positive number and consider the function $a(t) = (u - \tau v) \cdot \nabla h(x_0, t)$. Note that $a \in Dg(x_0)\mathbb{R}^m$, and it follows from (3.14) that $a(t_i) = z(t_i)$, $i = 1, \dots, n$. Moreover, we can choose τ large enough such that $a(t) \geq z(t)$ for all $t \in S \setminus \{t_1, \dots, t_n\}$. It follows then from the representation (3.13) of the cone $\mathcal{T}$ that there exists $y \in \mathcal{T}$ such that $a(t) + y(t) = z(t)$ for all $t \in S$. Since $\int_T z(t)\alpha(dt) = \sum_{t \in S} \alpha(t)z(t)$ and $z(t)$ is an arbitrary function, it follows that $\alpha(t) = 0$ for all $t \in S$ and hence $\alpha = 0$.

Now let us show that in the present situation the condition (2.2) is necessary, as well as sufficient, for uniqueness of the Lagrange multipliers measure μ. In order to show that condition (2.2) is necessary we have to verify that $\mathcal{R}(C, \mu) = T^*(C, \mu)$. For a set $A \in \mathcal{B}$, denote by $\mathcal{Z}(A)$ the set of (nonnegative) Borel measures whose support is contained in the set A. We have that $C = \mathcal{Z}(\Delta(x_0))$ and

$$(3.17)\ \mathcal{R}(C, \mu) = \{\alpha \in Y^* : \alpha = \alpha_1 - \alpha_2,\ \alpha_1 \in \mathcal{Z}(\Delta(x_0)),\ \alpha_2 \in \mathcal{Z}(\{t_1, \dots, t_n\})\}.$$

Consider a signed measure $\beta \in Y^* \setminus \mathcal{R}(C, \mu)$. Let $\beta = \beta^+ - \beta^-$ be the Jordan decomposition of β; i.e., β^+ and β^- are (nonnegative) Borel measures with disjoint supports T_1 and T_2, respectively. Since $\beta \notin \mathcal{R}(C, \mu)$, we have that $T_2 \not\subset \{t_1, \dots, t_n\}$. Consequently there is a nonzero function $y \in K$ whose support has empty intersection with the set $\{t_1, \dots, t_n\}$ and such that $\int_T y(t)\beta(dt) < 0$. It follows from the representation of $\mathcal{R}(C, \mu)$ given in (3.17) that for any $\alpha \in \mathcal{R}(C, \mu)$, $\int_T y(t)\alpha(dt) \geq 0$ and hence we can separate β from $\mathcal{R}(C, \mu)$ by the linear functional $\langle \cdot, y \rangle$. This shows that $\mathcal{R}(C, \mu)$ is closed in the w^*-topology of Y^* and hence $\mathcal{R}(C, \mu) = T^*(C, \mu)$.

Suppose now that condition (2.2) holds. Since $\mathcal{T} \subset \mathcal{L}$, condition (2.2) implies that $Dg(x_0)\mathbb{R}^m + \mathcal{L}$ is dense in $C(T)$. Therefore $\mathcal{L}^\perp \cap \operatorname{Ker}[Dg(x_0)]^* = \{0\}$ and hence condition (i$''$) follows. Furthermore, consider a function $z \in C(T)$ such that $z(t_i) = 0$, $i = 1, \dots, n$, and $z(t) > 0$ for all $t \in T \setminus \{t_1, \dots, t_n\}$. Let N be an open neighborhood of the set $\{t_1, \dots, t_n\}$. Then the set $\Delta(x_0) \setminus N$ is compact and hence there exists $\varepsilon > 0$ such that $z(t) \geq \varepsilon$ for all $t \in \Delta(x_0) \setminus N$. It follows then from condition (2.2) that there exists a function $a(t) = w \cdot \nabla h(x_0, t)$ such that $a(t) \geq \varepsilon/2$ for all $t \in \Delta(x_0) \setminus N$ and $a(t_i)$, $i = 1, \dots, n$, are arbitrarily close to zero. Because of the condition (i$''$), we can find a vector $u \in \mathbb{R}^m$ such that $u \cdot \nabla h(x_0, t_i) = a(t_i)$, $i = 1, \dots, n$. We obtain then that $(w-u)\cdot\nabla h(x_0,t_i)=0$, $i=1,\ldots,n$. Moreover, for $a(t_i)$, $i=1,\ldots,n$ sufficiently close to zero, we can choose such u that $(w-u)\cdot\nabla h(x_0,t)\geq\varepsilon/3$ for all $t\in\Delta(x_0)\setminus N$. Vector $v=u-w$ then satisfies (3.14) and (3.15) and hence condition (ii$''$) follows. □

Note that the Mangasarian–Fromovitz constraint qualification (3.10) is not assumed in Proposition 3.2. We only assume existence of a discrete Lagrange multipliers measure μ.

Vector v in the condition (ii$''$) of Proposition 3.2 generally depends on the neighborhood N. It is natural then to ask whether condition (ii$''$) can be replaced by the following stronger condition.

(ii$'''$) There exists $v\in\mathbb{R}^m$ such that

$$v\cdot\nabla h(x_0,t_i)=0,\quad i=1,\ldots,n, \tag{3.18}$$

$$v\cdot\nabla h(x_0,t)<0,\quad t\in\Delta(x_0)\setminus\{t_1,\ldots,t_n\}. \tag{3.19}$$

It is not difficult to see that if the set of active constraints $\Delta(x_0)$ is *finite*, then conditions (ii$''$) and (ii$'''$) are equivalent. As the following example shows, however, in general condition (ii$'''$) is not necessary for uniqueness of the Lagrange multipliers measure μ.

Example 3.1. Let $T=[0,4]$ and consider $h:\mathbb{R}^3\times[0,4]\to\mathbb{R}$ of the form $h(x,t)=x_1a_1(t)+x_2a_2(t)+x_3a_3(t)$, with the functions $a_i(t)$ defined as follows:

$$a_1(t)=\begin{cases} t^2, & t\in[0,1],\\ 1.5-0.5t, & t\in[1,3],\\ 0, & t\in[3,4],\end{cases}$$

$$a_2(t)=\begin{cases} -t, & t\in[0,1],\\ t-2, & t\in[1,4],\end{cases}$$

and $a_3(t)=1$ for $t\in[0,4]$. Also let $f(x)$ be a linear function with $\nabla f(x)=(0,0,-1)$. We have then that at $x_0=0$, $\nabla f(x_0)+\nabla h(x_0,0)=0$ and $\Delta(x_0)=[0,4]$. Therefore the first-order optimality conditions hold at $x_0=0$, with the Lagrange multipliers measure $\mu=\delta(t_1)$, $t_1=0$, and hence, since the considered program is convex, $x_0=0$ is the optimal solution of the considered program. We also have that for $v=(0,0,-1)$ and all $t\in[0,4]$, $v\cdot\nabla h(x_0,t)=-1$ and hence condition (3.10) is satisfied.

Let us observe now that condition (ii$'''$) does not hold here. Indeed, suppose there is a vector $v=(v_1,v_2,v_3)$ satisfying (3.18) and (3.19). It follows then from (3.18) that $v_3=0$ and from (3.19) that $v_2<0$. We obtain that $v\cdot\nabla h(x_0,0)=0$ and $\partial[v\cdot\nabla h(x_0,0)]/\partial t>0$. Therefore $v\cdot\nabla h(x_0,t)$ is positive for sufficiently small $t>0$, which of course contradicts (3.19).

On the other hand, it is not difficult to verify that conditions (i$''$) and (ii$''$) of Proposition 3.2 are satisfied here and hence μ is unique. This demonstrates that conditions (ii$''$) and (ii$'''$) are not equivalent and condition (ii$'''$) is not necessary for uniqueness of μ.

REFERENCES

[1] J. P. Aubin and I. Ekeland, *Applied Nonlinear Analysis*, Wiley, New York, 1984.

[2] J. F. Bonnans, *Local analysis of Newton-type methods for variational inequalities and nonlinear programming*, Appl. Math. Optim., 29 (1994), pp. 161–186.

[3] R. B. Holmes, *Geometric Functional Analysis and Its Applications*, Springer-Verlag, Berlin, New York, 1975.

[4] A. D. Ioffe and V. M. Tihomirov, *Theory of Extremal Problems*, North-Holland, Amsterdam, 1979.

[5] J. Kyparisis, *On uniqueness of Kuhn-Tucker multipliers in non-linear programming*, Math. Programming, 32 (1985), pp. 242–246.

[6] S. Kurcyusz, *On the existence and nonexistence of Lagrange multipliers in Banach spaces*, J. Optim. Theory Appl., 20 (1976), pp. 81–110.

[7] F. Lempio and H. Maurer, *Differential stability in infinite-dimensional nonlinear programming*, Appl. Math. Optim., 6 (1980), pp. 139–152.

[8] O. L. Mangasarian and S. Fromovitz, *The Fritz John necessary optimality conditions in the presence of equality and inequality constraints*, J. Math. Anal. Appl., 7 (1967), pp. 37–47.

[9] H. Maurer and J. Zowe, *First and second-order necessary and sufficient optimality conditions for infinite-dimensional programming problems*, Math. Programming, 16 (1979), pp. 98–110.

[10] S. M. Robinson, *First order conditions for general nonlinear optimization*, SIAM J. Appl. Math., 30 (1976), pp. 597–607.

[11] S. M. Robinson, *Stability theory for systems of inequalities, Part* II: *Differentiable nonlinear systems*, SIAM J. Numer. Anal., 13 (1976), pp. 497–513.

[12] A. Shapiro, *Perturbation analysis of optimization problems in Banach spaces*, Numer. Funct. Anal. Optim., 13 (1992), pp. 97–116.

[13] A. Shapiro, *Sensitivity analysis of parametrized programs via generalized equations*, SIAM J. Control Optim., 32 (1994), pp. 553–571.

[14] A. Shapiro, *On Lipschitzian stability of optimal solutions of parametrized semi-infinite programs*, Math. Oper. Res., 19 (1994), pp. 743–752.

[15] A. Shapiro, *First and second order analysis of nonlinear semidefinite programs*, Math. Programming Series B, to appear.

SIAM J. OPTIM.
Vol. 7, No. 2, pp. 519–526, May 1997

HADAMARD AND STRONG WELL-POSEDNESS FOR CONVEX PROGRAMS *

JULIAN P. REVALSKI†

Abstract. It is proved that Hadamard well-posedness of a constrained convex optimization problem with respect to Attouch–Wets convergence on the data implies, in general, its strong (and hence Tykhonov) well-posedness. An example is given showing that the opposite implication fails without assuming an appropriate constrained qualification condition.

Key words. convex optimization problems, well-posed optimization problem, Attouch–Wets convergence

AMS subject classifications. 49J45, 90C25

PII. S1052623495286776

1. Introduction.

Let $(X, \|\cdot\|)$ be a real Banach space and $\Gamma(X)$ denote the family of all convex lower semicontinuous extended real-valued functions in X which are proper. Recall that a function $f : X \to \mathbf{R} \cup \{+\infty\}$ is called proper if its *domain* $\operatorname{dom} f := \{x \in X : f(x) < +\infty\}$ is nonempty. An equivalent way to say that the function f is in $\Gamma(X)$ is that its *epigraph* $\operatorname{epi} f := \{(x,t) \in X \times \mathbf{R} : f(x) \leq t\}$ is a nonempty convex and closed subset of $X \times \mathbf{R}$ considered with the usual product topology.

Further, let $\operatorname{Conv}(X)$ designate the family of all nonempty convex and closed subsets of X. Each couple (A, f) from the Cartesian product $\operatorname{Conv}(X) \times \Gamma(X)$ determines in a natural way the following constrained convex minimization problem:

$$\text{find } x_0 \in A \text{ so that } f(x_0) = \inf\{f(x) : x \in A\} =: \inf(A, f).$$

Such a problem (which often is also termed a convex program) will be identified in the sequel with the couple (A, f), and its (possible empty) set of solutions will be denoted by $\operatorname{argmin}(A, f)$.

A sequence $\{x_n\}_{n=1}^{\infty} \subset A$ with $f(x_n) \to \inf(A, f)$ is called *minimizing* for the problem (A, f). Such sequences are also called sequences of approximate solutions for (A, f). The minimization problem $(A, f) \in \operatorname{Conv}(X) \times \Gamma(X)$ is called *Tykhonov well-posed* [Ty, DZ] if it has a unique solution $x_0 \in A$ and, moreover, every minimizing sequence for (A, f) converges to x_0. For convex functions in finite dimensions the uniqueness of the solution is enough to guarantee its Tykhonov well-posedness (even a stronger notion for well-posedness; see below). This is no longer valid in infinite dimensions (even in Hilbert spaces) as the following well-known example shows.

Example 1.1. Consider the Hilbert space $\ell_2 = \{\{x_i\}_{i=1}^{\infty} : x_i \in \mathbf{R}, \sum_{i=1}^{\infty} x_i^2 < \infty\}$ with the usual ℓ_2-norm. Let $f : \ell_2 \to \mathbf{R}$ be defined as follows:

$$f(x) = \sum_{n=1}^{\infty} \frac{\langle x, e_n \rangle^2}{n^2},$$

* Received by the editors May 26, 1995; accepted for publication (in revised form) January 8, 1996. This study was partially supported by Bulgarian National Fund for Scientific Research contract MM-408/94.

http://www.siam.org/journals/siopt/7-2/28677.html

† Institute of Mathematics, Bulgarian Academy of Sciences, Acad. G. Bonchev Street, Block 8, 1113 Sofia, Bulgaria (revalski@banmatpc.math.acad.bg).

where $\langle \cdot, \cdot \rangle$ is the usual scalar product and $e_n = (0, 0, \dots, 1, 0, \dots)$, 1 at the nth place; $n = 1, 2, \dots$, is the standard basis. The function f is convex and continuous, the problem (X, f) has unique solution at $x_0 = 0$, but $\{e_n\}_{n=1}^{\infty}$ is a minimizing sequence which does not converge to x_0.

The idea of Tykhonov well-posedness is to control the sequences of approximate solutions, and this is motivated by the numerical point of view—every numerical method solving (A, f) usually produces a minimizing-for-(A, f) sequence. This notion, however, takes into account the minimizing sequences only inside A. But when we search for a minimum of f on a proper subset A of X the above idea deserves to be broadened by taking care also for sequences of approximate solutions that can be outside A, not only those in A. Various reasons could lead to such kinds of sequences like, for example, approximations of the data A and f, possible "errors" in these data, the use of methods for solving (A, f) which allow a minimizing sequence for (A, f) to be outside A (e.g., penalty methods), etc.

We will consider two generalizations of the notion of minimizing sequence. The first one was introduced and studied by Levitin and Polyak [LePo]: A sequence $\{x_n\}_{n=1}^{\infty} \subset X$ is called a *Levitin–Polyak minimizing sequence* for the minimization problem (A, f) if, in addition to $f(x_n) \to \inf(A, f)$, one has also $d(x_n, A) \to 0$ where $d(x, A) := \inf\{\|x - y\| : y \in A\}$, $x \in X$, is the distance function generated by the set A. In other words, a sequence $\{x_n\}_{n=1}^{\infty}$ is a Levitin–Polyak minimizing sequence for (A, f) if not only $\{f(x_n)\}_{n=1}^{\infty}$ approaches the infimum of f over A but also the sequence $\{x_n\}_{n=1}^{\infty}$ tends (with respect to the norm) to A.

A second (and further) generalization of the usual notion of minimizing sequence is the following: a sequence $\{x_n\}_{n=1}^{\infty} \subset X$ is said to be a *generalized minimizing sequence* for the minimization problem (A, f) [BL2, BL3] if both $d(x_n, A) \to 0$ and $\limsup f(x_n) \leq \inf(A, f)$ are fulfilled. Let us mention a fact that will be used often in the sequel: every subsequence of a usual, Levitin–Polyak, or generalized minimizing sequence is again a minimizing sequence from the corresponding type.

Now, following the scheme of the definition of Tykhonov well-posedness, we have the next two strengthened versions of the well-posedness: the minimization problem $(A, f) \in \mathrm{Conv}(X) \times \Gamma(X)$ is said to be *Levitin–Polyak well-posed* [LePo, DZ] (respectively, *strongly well-posed* [BL2, BL3]) if it has unique solution $x_0 \in A$ and, moreover, every Levitin–Polyak minimizing sequence (respectively, every generalized minimizing sequence) for (A, f) converges to x_0. The strong well-posedness was previously considered in an equivalent geometric form by this author in [R1, R2].

The three notions above, each of which is based on the behavior of a certain prescribed set of minimizing sequences, have been intensively studied (not only for convex programs) in many papers—see, e.g., [BL1, BL2, BL3, L, LPa1, LPa2, R1, R2, RZh, Sh] and especially the monograph [DZ] and the collection of overviews devoted to the subject [LR]. We will mention here several facts related to them only for convex problems from $\mathrm{Conv}(X) \times \Gamma(X)$.

In the following facts we always have in mind a convex minimization problem $(A, f) \in \mathrm{Conv}(X) \times \Gamma(X)$. Obviously, in general, strong well-posedness of the problem (A, f) implies Levitin–Polyak one, which in its turn implies Tykhonov well-posedness. In finite dimensions the uniqueness of the solution to (A, f) implies its strong well-posedness ([BL2], Theorem 2.4; here the convexity assumptions play a crucial role), a fact which is no longer valid in infinite dimensions as Example 1.1 shows. Levitin–Polyak and strong well-posedness for (A, f) coincide in reflexive Banach spaces ([BL2], Theorem 2.2), but, in general, in this setting (even in the setting of Hilbert spaces), Tykhonov well-posedness is a strictly weaker notion than Levitin–Polyak well-posedness (a counterexample is given in [BL2]).

Our concern here, however, will be to compare the above types of well-posedness with another one, which arises from the original idea of Hadamard for continuous dependence of the solution of a problem on the data. To introduce it in the setting we consider, we need a suitable topology on the data space $\mathrm{Conv}(X) \times \Gamma(X)$. Here we will consider the already well-known Attouch–Wets topology (known also as bounded Hausdorff topology) in the hyperspace of all closed and convex subsets of a Banach space (see [AW, Mo] for the origins and the recent monograph of Beer [B] for a detailed study). This topology has turned out to be very useful in quantitative analysis in convex optimization when we deal with convex subsets that are not necessarily bounded. Below we give a short description of this topology following mostly [B].

Given the Banach space X and two nonempty subsets $A, B \subset X$ we put, as usual, $e(A,B) := \sup\{d(a,B) : a \in A\}$ to denote the *excess* of A to B. Further, for $\rho > 0$ let $B_\rho(X)$ be the closed ball in X centered at the origin θ with radius ρ. Then, the so-called ρ-Hausdorff distance between A and B is the following number:

$$\mathrm{haus}_\rho(A,B) := \max\{e(A \cap B_\rho(X), B), e(B \cap B_\rho(X), A)\}.$$

The usual convention here is $e(\emptyset, C) = 0$ for each nonempty $C \subset X$.

The sequence $\{A_n\}_{n=1}^\infty \subset \mathrm{Conv}(X)$ is called *Attouch–Wets convergent* to $A \in \mathrm{Conv}(X)$ if there exists $\rho_0 > 0$ so that $\lim_{n\to\infty} \mathrm{haus}_\rho(A_n, A) = 0$ for every $\rho \geq \rho_0$. This convergence is (completely) metrizable (see, e.g., [B]), but we will use here the above definition which is more convenient in our setting. The resulting topology is denoted usually by τ_{aw} and is known as Attouch–Wets topology or bounded Hausdorff topology.

Further, to introduce the same kind of topology in $\Gamma(X)$ we identify, as usual, each function from $\Gamma(X)$ with its epigraph $\mathrm{epi}\, f$ which is an element of $\mathrm{Conv}(X \times \mathbf{R})$, $X \times \mathbf{R}$ being considered with the product topology. Take on $X \times \mathbf{R}$ the box norm (which generates the product topology)

$$\|(x,\alpha)\| := \max\{\|x\|, |\alpha|\},\ (x,\alpha) \in X \times \mathbf{R}.$$

Thinking that $\Gamma(X)$ is the family $\{\mathrm{epi}\, f : f \in \Gamma(X)\}$, we consider on the latter the topology inherited from the Attouch–Wets topology in $\mathrm{Conv}(X \times \mathbf{R})$. This inherited topology is again known as Attouch–Wets topology in $\Gamma(X)$ or also as epi-distance topology. For this topology we will again use the symbol τ_{aw}. Let us mention that the uniform convergence on bounded subsets of X in $\Gamma(X)$ is a stronger convergence than that generated by the Attouch–Wets topology.

Having the Attouch–Wets topology both in $\mathrm{Conv}(X)$ and $\Gamma(X)$ we consider on the Cartesian product $\mathrm{Conv}(X) \times \Gamma(X)$ the (metrizable) product topology (again denoted by τ_{aw}) generated by the Attouch–Wets topologies in $\mathrm{Conv}(X)$ and $\Gamma(X)$.

A minimization problem $(A, f) \in \mathrm{Conv}(X) \times \Gamma(X)$ is said to be *Hadamard well-posed* with respect to τ_{aw} (or τ_{aw}-Hadamard well-posed) if it has unique solution $x_0 \in A$ and, moreover, if $\{(A_n, f_n)\}_{n=1}^\infty$ is a sequence of problems from $\mathrm{Conv}(X) \times \Gamma(X)$ which τ_{aw}-converges to (A, f) and if $\{x_n\}_{n=1}^\infty$ is a sequence form X so that $x_n \in \mathrm{argmin}(A_n, f_n)$ for every $n = 1, 2, \ldots$, then $x_n \to x_0$.

The above notion reflects the idea of continuous dependence of the solution on the data. Let us stress the fact that it depends both on the data space (here $\mathrm{Conv}(X) \times \Gamma(X)$) and on the topology considered in this data space (in this case the Attouch–Wets topology). Observe also that in the definition of τ_{aw}-Hadamard well-posedness for (A, f) we are interested only in those τ_{aw}-convergent to (A, f) sequences $\{(A_n, f_n)\}_{n=1}^{\infty}$ for which the solution sets $\operatorname{argmin}(A_n, f_n)$, $n = 1, 2, \ldots$, are nonempty. The existence of nontrivial sequences from this type is guaranteed by the Ekeland variational principle (see the next section).

Having this notion of well-posedness naturally arises the question of its comparison with the above introduced notions of well-posedness based on the behavior of minimizing sequences. With different convergences on $\operatorname{Conv}(X)$ or $\Gamma(X)$ partial results in this direction have been already done. These partial results are related to settings when either f is fixed and only A may vary, or vice-versa—f varies while A is fixed and can be found in [BL1, L, LPa2].

Our aim, however, is to see what are the relationships in the general setting we consider, namely, when both A and f can vary with respect to the Attouch–Wets convergence in $\operatorname{Conv}(X)$ and $\Gamma(X)$, respectively. In this case, Beer and Lucchetti [BL3] proved that under suitable constrained qualification conditions for $(A, f) \in \operatorname{Conv}(X) \times \Gamma(X)$, Tykhonov well-posedness of (A, f) implies its τ_{aw}-Hadamard well-posedness (see the precise formulation in the next section). It has not been known so far what could be said about the opposite implication. Here we fill this gap by showing that, in general, τ_{aw}-Hadamard well-posedness of $(A, f) \in \operatorname{Conv}(X) \times \Gamma(X)$ implies even the strong well-posedness of (A, f) without any additional conditions on (A, f). An example is given showing that, in general, (even in finite dimensions), τ_{aw}-Hadamard well-posedness for a convex program is a strictly stronger notion than the strong well-posedness of this program. A similar result in the case when the constrained set is given by inequalities is obtained in [KoR]. Due to the different way of introducing the notions of well-posedness in the setting with inequalities, neither of the results is derivable from the other.

Finally, based on the recommendation of one of the referees we discuss the relationship between the well-posedness of a convex program and its so-called value well-posedness (convergence of infimal values).

2. Main result. We start this section with a known result giving a sufficient condition under which Tykhonov well-posedness of a problem $(A, f) \in \operatorname{Conv}(X) \times \Gamma(X)$ implies its τ_{aw}-Hadamard well-posedness. In the next theorem $\operatorname{Int}A$ means, as usual, the interior of the set $A \subset X$.

THEOREM 2.1 (see [BL3], Theorem 4.1). *Let $(A, f) \in \operatorname{Conv}(X) \times \Gamma(X)$ be Tykhonov well-posed and suppose that either f is continuous and finite at some point of A or $\operatorname{Int}A \cap \operatorname{dom} f \neq \emptyset$. Then, (A, f) is τ_{aw}-Hadamard well-posed. Moreover, $\inf(A_n, f_n) \to \inf(A, f)$., whenever $\{(A_n, f_n)\}_{n=1}^{\infty} \tau_{aw}$-converges to (A, f).*

The last conclusion (i.e., when $\inf(A_n, f_n) \to \inf(A, f)$, provided $(A_n, f_n) \to (A, f)$) is known as *value well-posedness* of the problem (A, f) (see, e.g., [DZ]). Of course this notion again depends on the convergence on the data space.

In general, Tykhonov well-posedness (even strong well-posedness) of (A, f) does not imply its τ_{aw}-Hadamard well-posedness—see Example 2.3 below. However, we will show that the opposite implication is always true. In other words, τ_{aw}-Hadamard well-posedness for a minimization problem $(A, f) \in \operatorname{Conv}(X) \times \Gamma(X)$ is a stronger notion than Tykhonov well-posedness (even than strong well-posedness). Namely, we will prove the following result.

THEOREM 2.2. *Let $(A, f) \in \operatorname{Conv}(X) \times \Gamma(X)$ be τ_{aw}-Hadamard well-posed. Then, (A, f) is strongly well-posed. In particular, (A, f) is Levitin–Polyak and Tykhonov well-posed.*

Before giving the proof of this theorem we formulate one of the versions of the famous Ekeland variational principle that we will need later.

Ekeland Variational Principle. *Let (Z, d) be a complete metric space and $f : Z \to \mathbf{R}\bigcup\{+\infty\}$ be a proper lower semicontinuous and bounded-from-below function. Then for every $\varepsilon > 0$ and $z_0 \in Z$ with $f(z_0) < \inf(Z, f) + \varepsilon$ there exists a point $z_1 \in Z$ so that*

(i) $\|z_1 - z_0\| \leq \sqrt{\varepsilon}$;

(ii) $f(z) + \sqrt{\varepsilon}\|z - z_1\| > f(z_1)$ *for every* $z \in Z$ *and* $z \neq z_1$.

Proof of Theorem 2.2. Let $(A, f) \in \operatorname{Conv}(X) \times \Gamma(X)$ be τ_{aw}-Hadamard well-posed with unique solution $x_0 \in A$ and assume that it is not strongly well-posed. Then there is a generalized minimizing sequence $\{x_n\}_{n=1}^{\infty} \subset X$ for (A, f) which does not converge to the unique solution x_0. In other words, $d(x_n, A) \to 0$, $\limsup f(x_n) \leq \inf(A, f)$, but $x_n \not\to x_0$ in the norm. Without loss of generality we may assume that $x_0 = \theta$, θ being the origin in X, and also that $f(\theta) = \inf(A, f) = 0$. Since $\{x_n\}_{n=1}^{\infty}$ does not converge to θ there is some real number $r > 0$ so that $\|x_n\| \geq r$ for infinitely many n. Again, without loss of generality, we may think that the last inequality is fulfilled for every $n = 1, 2, \ldots$ and that $r \leq 1$. Consider the line segments $[\theta, x_n]$ and the points $z_n := (r/\|x_n\|)x_n$, $n = 1, 2, \ldots$ on them. Observe that $\|z_n\| = r$ for every $n = 1, 2, \ldots$.

Since A is convex, we have $d(z_n, A) \leq d(x_n, A)$ for every $n = 1, 2, \ldots$. Therefore, $d(z_n, A) \to 0$. On the other hand, $f(\theta) = \inf(A, f) = 0$; when using the convexity of f, one easily sees that $\limsup f(z_n) \leq \inf(A, f)$. Therefore, we have shown that the sequence $\{z_n\}_{n=1}^{\infty}$ is again a generalized minimizing sequence for the minimization problem (A, f).

For each $n = 1, 2, \ldots$, consider now the sets $A_n := \operatorname{co}(\{z_n\} \cup (B_n(X) \cap A))$, where co means the convex hull operation and $\{z_n\}$ is the one-point set consisting of z_n. Remember that $B_n(X)$ meant the closed ball with center at the origin and radius n. Since $\theta \in A$, the sets A_n are nonempty for every $n = 1, 2, \ldots$. Hence, $\{A_n\}_{n=1}^{\infty} \subset \operatorname{Conv}(X)$. We will show that this sequence τ_{aw}-converges to (A, f).

Let $\rho_0 > 0$ and fix some $\rho \geq \rho_0$ and $\varepsilon > 0$. Take n so large that $n \geq \rho$. Then

$$e(B_\rho(X) \cap A, A_n) = 0. \tag{1}$$

On the other hand, let n be so large that we also have $d(z_n, A) \leq \varepsilon$. Take an arbitrary $x \in A_n$. Then for some $a \in B_n(X) \cap A$ and $\lambda \in [0, 1]$, we have $x = \lambda z_n + (1 - \lambda)a$, whence (using the convexity of A) we get $d(x, A) \leq \varepsilon$. Since this is true for every $x \in A_n$ we obtain $e(A_n, A) \leq \varepsilon$. This together with (1) imply that $\operatorname{haus}_\rho(A_n, A) \leq \varepsilon$. Hence, $\{A_n\}_{n=1}^{\infty}$ τ_{aw}-converges to A in $\operatorname{Conv}(X)$.

Further, consider the sequence of real numbers $\{\gamma_n\}_{n=1}^{\infty}$ defined by

$$\gamma_n := f(z_n) - \inf(A_n, f),\ n = 1, 2, \ldots. \tag{2}$$

Each γ_n, $n = 1, 2, \ldots$, is well defined since A_n are bounded and hence f is bounded from below on A_n. Moreover, γ_n are nonnegative since $z_n \in A_n$ for every $n = 1, 2 \ldots$. We will show that $\{\gamma_n\}_{n=1}^{\infty}$ converges to 0.

Indeed, fix n, take $c_n \in A_n$ with

$$f(c_n) \leq \inf(A_n, f) + \frac{1}{n^4}, \tag{3}$$

and apply the Ekeland variational principle for the set A_n, the function f, the point c_n, and the number $1/n^4$. We get a point $c_n' \in A_n$ with the following two properties:

(a) $\|c_n - c_n'\| \leq \dfrac{1}{n^2}$;

(b) $h_n(x) := f(x) + \frac{1}{n^2}\|x - c'_n\| > f(c'_n) = h_n(c'_n)$ for each $x \in A_n$, $x \neq c'_n$ (i.e., $c'_n = \operatorname{argmin}(A_n, h_n)$).

Repeating this procedure for every $n = 1, 2, \ldots$, we obtain a sequence of points $\{c_n\}_{n=1}^{\infty} \subset X$, with $c_n \in A_n$ for every n, satisfying (3), a sequence of points $\{c'_n\}_{n=1}^{\infty} \subset X$ so that $c'_n \in A_n$ for every n, and a sequence of functions $\{h_n\}_{n=1}^{\infty} \subset \Gamma(X)$ satisfying (a) and (b). We will show that $\{h_n\}_{n=1}^{\infty}$ τ_{aw}-converges to f.

First, observe that $\operatorname{dom} h_n = \operatorname{dom} f$ for every $n = 1, 2, \ldots$. Further, since $\operatorname{epi} h_n \subset \operatorname{epi} f$ for each $n = 1, 2, \ldots$, the only thing to be seen is that for some ρ_0 we have $e(B_\rho(X \times \mathbf{R}) \cap \operatorname{epi} f, \operatorname{epi} h_n) \to 0$ for each $\rho \geq \rho_0$. Fix some $\rho_0 > 0$ and observe that $B_{\rho_0}(X \times \mathbf{R}) \cap \operatorname{epi} f \neq \emptyset$ since $(\theta, 0) \in \operatorname{epi} f$. Pick arbitrary $\rho \geq \rho_0$ and $\varepsilon > 0$. Let (x, t) be from $B_\rho(X \times \mathbf{R}) \cap \operatorname{epi} f$. Then, $f(x) \leq t$ and, moreover, $\|x\| \leq \rho$ and $|t| \leq \rho$. We will show that for n large enough $d((x,t), \operatorname{epi} h_n) < \varepsilon$ giving

$$e(B_\rho(X \times \mathbf{R}) \cap \operatorname{epi} f, \operatorname{epi} h_n) \leq \varepsilon,$$

thus showing that $\{h_n\}_{n=1}^{\infty}$ is τ_{aw}-convergent to f.

Indeed, let $t = f(x) + t_0$, $t_0 \geq 0$. Consider the point (x, t') where $t' := h_n(x) + t_0$. Obviously, $(x, t') \in \operatorname{epi} h_n$. Let n be so large that $(1/n^2)\rho + 1/n < \varepsilon$. Hence, we have the following (remember that $r \leq 1$, hence $A_n \subset B_n(X)$ giving $\|c'_n\| \leq n$ since $c'_n \in A_n$):

$$h_n(x) - f(x) = \frac{1}{n^2}\|x - c'_n\| \leq \frac{1}{n^2}\|x\| + \frac{1}{n^2}\|c'_n\| \leq \frac{1}{n^2}\rho + \frac{1}{n} \leq \varepsilon.$$

Therefore, for n large enough we get

$$\|(x,t) - (x,t')\| = |h_n(x) - f(x)| \leq \varepsilon,$$

giving $d((x,t), \operatorname{epi} h_n) \leq \varepsilon$.

Hence, we have shown that $\{h_n\}_{n=1}^{\infty}$ τ_{aw}-converges to f. We saw also that $\{A_n\}_{n=1}^{\infty}$ τ_{aw}-converges to A. Thus, by the fact that (A, f) is τ_{aw}-Hadamard well-posed and $\{c'_n\} = \operatorname{argmin}(A_n, h_n)$ for every $n = 1, 2, \ldots$, we get that $c'_n \to x_0$. Therefore (see (a) above), $c_n \to x_0 (= \theta)$. Let $\varepsilon > 0$ be arbitrary. Then, using that f is lower semicontinuous, we have that for large n

$$f(c_n) \geq f(x_0) - \varepsilon = \inf(A, f) - \varepsilon. \tag{4}$$

On the other hand, $\limsup f(z_n) \leq \inf(A, f)$ giving that for large n

$$f(z_n) \leq \inf(A, f) + \varepsilon. \tag{5}$$

Hence, combining (4) and (5) and having in mind (3) we get that for n large enough

$$0 \leq \gamma_n = f(z_n) - \inf(A_n, f) \leq \inf(A, f) + \varepsilon - \inf(A_n, f)$$

$$\leq f(c_n) + 2\varepsilon - \inf(A_n, f) \leq 2\varepsilon + \frac{1}{n^4}.$$

Therefore, $\gamma_n \to 0$.

Let us mention that without loss of generality we may assume that $\gamma_n > 0$ for every $n = 1, 2 \ldots$, since, otherwise, passing to subsequences and with abuse of notation, we would have $z_n \in \operatorname{argmin}(A_n, f)$ which together with τ_{aw}-convergence of $\{A_n\}_{n=1}^{\infty}$

to A would give $z_n \to x_0 = \theta$ in contrast to $\|z_n\| = r > 0$. So, we may think that $\gamma_n > 0$ for every $n = 1, 2 \cdots$.

Apply now for each $n = 1, 2 \cdots$ the Ekeland variational principle, this time for the set A_n, the function f, the point z_n, and the number $2\gamma_n$ (look at the definition in (2)). We get the existence of $z_n' \in A_n$ with

(i) $\|z_n - z_n'\| \leq \sqrt{2\gamma_n}$;

(ii) $f_n(x) := f(x) + \sqrt{2\gamma_n}\|x - z_n'\| > f(z_n') = f_n(z_n')$ for every $x \in A_n$ with $x \neq z_n'$ (i.e., $\{z_n'\} = \operatorname{argmin}(A_n, f_n)$).

Observe that $\operatorname{dom} f_n = \operatorname{dom} f$ and that $\operatorname{epi} f_n \subset \operatorname{epi} f$ for every $n = 1, 2, \ldots$. Moreover, since $\{z_n\}_{n=1}^{\infty}$ is bounded and $\gamma_n \to 0$, the sequence $\{z_n'\}_{n=1}^{\infty}$ is bounded too. Hence, using that $\gamma_n \to 0$, the sequence $\{f_n\}_{n=1}^{\infty}$ converges to f uniformly on the bounded subsets of X. Therefore, $\{f_n\}_{n=1}^{\infty}$ is τ_{aw}-convergent to f. The last together with τ_{aw}-convergence of $\{A_n\}_{n=1}^{\infty}$ to A give that $z_n' \to x_0 (= \theta)$ since (A, f) was τ_{aw}-Hadamard well-posed. By (i) above, $z_n \to x_0 (= \theta)$ as well. The last contradicts to $\|z_n\| = r > 0$. The proof of the theorem is completed. □

Now, let us give an example showing that, in general, τ_{aw}-Hadamard well-posedness of a convex problem $(A, f) \in \operatorname{Conv}(X) \times \Gamma(X)$ is a strictly stronger assumption than strong well-posedness for (A, f) even in finite dimensions. In other words, in the absence of a constrained qualification condition, strong well-posedness does not imply τ_{aw}-Hadamard well-posedness.

Example 2.3. Let $X := \mathbf{R}^2$ with the usual norm $\|\cdot\|$ and $A := \{(x, y) \in \mathbf{R}^2 : y = 0\}$. Let $f(x) := \|x\|$ if $x \in A$ and $f(x) := \infty$ provided x is outside A. Then, (A, f) is Tykhonov well-posed and, moreover, since $\operatorname{dom} f = A$, it is also strong well-posed with unique minimum at $(0, 0)$. On the other hand, let $A_n := \{(x, y) \in \mathbf{R}^2 : y = 1/n(x - 1)\}$. It is seen that $\{A_n\}_{n=1}^{\infty}$ τ_{aw}-converges to A while $(1, 0) = \operatorname{argmin}(A_n, f)$ does not converge to $(0, 0)$.

Finally, we discuss shortly the connection between well-posedness of a convex problem $(A, f) \in \operatorname{Conv}(X) \times \Gamma(X)$ and its τ_{aw}-value well-posedness. This issue was brought to our attention by one of the referees. Unfortunately, it seems little can be done in this direction. First of all, in general, τ_{aw}-Hadamard well-posedness (or Tykhonov well-posedness) does not imply τ_{aw}-value well-posedness as the following example given by the same referee shows.

Example 2.4. Let $X = \mathbf{R}$, $f : X \to \mathbf{R}$ be defined by $f(x) := 0$ if $x = 0$ and $f(x) := \infty$ if $x \neq 0$. Let $A := \{0\}$. Then, it is easily seen that (A, f) is τ_{aw}-Hadamard well-posed (and also Tykhonov well-posed). However, if we consider $A_n := \{1/n\}$ and $f_n(x) := nx$ if $x \in [0, 1/n]$ and $f(x) = \infty$ otherwise, $n = 1, 2, \ldots$, then obviously (A_n, f_n) τ_{aw}-converges to (A, f) while $\inf(A_n, f_n) = 1$ does not converge to $\inf(A, f) = 0$.

On the other hand, as is stated in Theorem 2.1, the constrained qualification conditions from Theorem 2.1 together with Tykhonov well-posedness imply τ_{aw}-value well-posedness. Hence the following corollary is straightforward having in mind Theorem 2.2.

Corollary 2.5. *Let $(A, f) \in \operatorname{Conv}(X) \times \Gamma(X)$ be τ_{aw}-Hadamard well-posed and suppose that either f is continuous and finite at some point of A or $\operatorname{Int} A \cap \operatorname{dom} f \neq \emptyset$. Then, (A, f) is τ_{aw}-value well-posed.*

The question that is put by the referee is whether the converse is true provided one of the constrained qualification conditions above is fulfilled. Unfortunately, this is not true either as the following slight modification of Example 1.1 shows. For a function $g \in \Gamma(X)$ and a set $C \in \operatorname{Conv}(X)$ we consider the following type of restriction:

$g|C(x) = g(x)$ if $x \in C$ and $g|C(x) = \infty$ otherwise. Observe that $g|C \in \Gamma(X)$ and that $\inf(C, g) = \inf(X, g|C)$.

Example 2.6. Let X and f be as in Example 1.1. Let A be the closed unit ball in X. The problem (A, f) satisfies both constrained qualification conditions from Theorem 2.1. Let $\{(A_n, f_n)\}$ τ_{aw}-converge to (A, f). Because of the constrained qualification conditions we have that $\{f_n|A_n\}$ τ_{aw}-converges to $f|A$ (Theorem 3.6 from [BL3]) and since $f|A$ has bounded level sets, we get $\inf(X, f_n|A_n) \to \inf(X, f|A)$ (Theorem 3.7 from [BL1]). Hence, (A, f) is τ_{aw}-value well-posed. But as Example 1.1 shows, (A, f) is not Tykhonov well-posed (and, by Theorem 2.2, it is not τ_{aw}-Hadamard well-posed either; the latter also could be seen directly).

Acknowledgments. The author would like to express his gratitude to two anonymous referees for their valuable remarks and proposals for additional considerations which led to an improvement of the paper.

REFERENCES

[AW] H. Attouch and R. Wets, *Quantitative stability of variational systems*, Trans. Amer. Math. Soc., 328 (1991), pp. 695–730.

[B] G. Beer, *Topologies on closed and closed convex sets*, in Mathematics and its Applications, Vol. 268, Kluwer Academic Publishers, Dordrecht, 1993.

[BL1] G. Beer and R. Lucchetti, *Convex optimization and the epi-distance topology*, Trans. Amer. Math. Soc., 327 (1991), pp. 795–813.

[BL2] G. Beer and R. Lucchetti, *Solvability for constrained problems*, Univ. Degli Studi di Milano, Dipartimento di Mat., Quaderno n.3, 1991, preprint.

[BL3] G. Beer and R. Lucchetti, *The epi-distance topology: Continuity and stability results with application to convex optimization problems*, Math. Oper. Res., 17 (1992), pp. 715–726.

[DZ] A. L. Dontchev and T. Zolezzi, *Well-posed Optimization Problems*, in Lecture Notes in Mathematics, Vol. 1543, Springer-Verlag, Berlin, New York, 1993.

[KoR] A. S. Konsulova and J. P. Revalski, *Constrained convex optimization problems—well-posedness and stability*, Numer. Funct. Anal. Optim., 15 (1994), pp. 889–907.

[LePo] E. S. Levitin and B. T. Polyak, *Convergence of minimizing sequences in conditional extremum problems*, Soviet Math. Dokl., 7 (1966), pp. 764–767.

[L] R. Lucchetti, *Some aspects of the connection between Hadamard and Tykhonov well-posedness of convex problems*, Boll. Un. Mat. Ital. C, 6 (1982), pp. 337–345.

[LPa1] R. Lucchetti and F. Patrone, *A characterization of Tykhonov well-posedness for minimum problems, with applications to variational inequalities*, Numer. Funct. Anal. Optim., 3 (1981), pp. 461–476.

[LPa2] R. Lucchetti and F. Patrone, *Hadamard and Tykhonov well-posedness of a certain class of convex functions*, J. Math. Anal. Appl., 88 (1982), pp. 204–215.

[LR] R. Lucchetti and J. P. Revalski, eds., *Recent developments in well-posed variational problems*, in Mathematics and its Applications, Vol. 331, Kluwer Academic Publishers, Dordrecht, 1995.

[Mo] U. Mosco, *Convergence of convex sets and of solutions of variational inequalities*, Adv. Math., 3 (1969), pp. 510–585.

[R1] J. P. Revalski, *Generic properties concerning well-posed optimization problems*, Compt. Rend. Acad. Bulg. Sci., 38 (1985), pp. 1431–1434.

[R2] J. P. Revalski, *Generic well-posedness in some classes of optimization problems*, Acta Univ. Carolin. Math. Phys., 28 (1987), pp. 117–125.

[RZh] J. P. Revalski and N. V. Zhivkov, *Well-posed constrained optimization problems in metric spaces*, J. Optim. Theory Appl., 76 (1993), pp. 145–163.

[Sh] P. Shunmugaraj, *Well-set and well-posed minimization problems*, Set-valued Anal., 3 (1995), pp. 295–305.

[Ty] A. N. Tykhonov, *On the stability of the functional optimization problem*, U.S.S.R. Comput. Math. and Math. Phys., 6 (4) (1966), pp. 28–33.

SIAM J. Optim.
Vol. 7, No. 2, pp. 527–546, May 1997

A PROJECTION-BASED ALGORITHM FOR CONSISTENT AND INCONSISTENT CONSTRAINTS[*]

TUVIA KOTZER[†], NIR COHEN[‡], AND JOSEPH SHAMIR[‡]

Abstract. Signal synthesis and reconstruction is considered when the signal is to be determined by N constraint sets, C_i. The solution sought is required to minimize a weighted quadratic cost functional $\hat{J}$. Emphasis is on cases in which the intersection of the sets C_i is empty.

Our proposed procedure employs a suitably weighted simultaneous projection iteration method. It is shown that the iterates generated by the algorithm converge weakly to a global minimizer of $\hat{J}$ provided the set of fixed points of the algorithm is nonempty. If the problem is consistent ($C_o := \bigcap C_i \neq \emptyset$), weak convergence is to an element in C_o. However, it is indicated that large classes of inconsistent problems, which could not be treated by existing methods, admit a solution as well.

Key words. asymptotic regularity, fixed points, product space, projections onto convex sets, nonexpansivity, weak convergence, weighted norms

AMS subject classifications. 47H09, 47H10, 52A41, 65D15, 90C25

PII. S1052623494278347

1. Introduction. With the increasing demands and complexity of signal processing systems, renewed interest is emerging in the set theoretic formulations (see, e.g., [11]) applied to optimization problems such as signal synthesis in pattern recognition [35, 33, 22, 26], computerized tomography [20], constrained deconvolution [36, 23], etc. The general problem is to restore (or synthesize) a signal from a finite set of constraints it is known to satisfy.

More specifically, in the set theoretic formulation we are given N convex constraint sets C_i in a signal space $\mathcal{H}$ (typically a Hilbert space). A signal $f \in \mathcal{H}$ is called *feasible* if and only if $f \in C_o := \cap_{i=1}^{N} C_i$. The problem is called *consistent* if C_o is nonempty, i.e., if feasible signals exist. Assuming consistency, the aim is to produce a feasible signal given an initial, nonfeasible estimate.

In theory, a consistent problem may be solved directly by projecting the initial estimate orthogonally onto C_o. This, however, requires a precise analytic description of the intersection $C_o = \cap_{i=1}^{N} C_i$, which may be a highly nontrivial task in practice, especially when the sets C_i contain uncertainties. In reality, a feasible solution can be approached iteratively, via algorithms which use exclusively the N individual orthogonal projections onto the individual sets C_i.

The algorithms available for solving consistent problems are of two major types: sequential (serial) or simultaneous (parallel). A third important class, consisting of the so-called block iterative algorithms, will not be considered here.

In a sequential algorithm, each step involves a single set C_i, properly chosen. The simplest sequential procedure consists of a cyclic selection of the sets C_i in equal cycles of length N. In a simultaneous algorithm, each step involves all the sets C_i, where typically the weighting of the different sets is iteration independent. The literature

[*] Received by the editors December 5, 1994; accepted for publication (in revised form) December 5, 1995. This work was performed within the Technion Advanced Opto-Electronics Center and supported by the American Technion Society (ATS)-New York.

http://www.siam.org/journals/siopt/7-2/27834.html

[†] Sarin Technologies, Rehov Habonim 7, Ramat Gan 52462, Israel (tuvia@techunix.technion.ac.il).

[‡] Department of Electrical Engineering, Technion–Israel Institute of Technology, Haifa 32000, Israel (nirc@ee.technion.ac.il, jsh@ee.technion.ac.il).

on the application of these two types of algorithms for solving consistent problems is quite extensive; see, e.g., [1, 14, 15, 16] and the reference therein. Typically, general conditions guarantee the weak convergence of the iterates to a feasible solution.

In practice, many problems turn out to be marginally consistent or fully inconsistent (see, e.g., [29]). This is often the result of overly optimistic design: underestimating noise statistics (in image restoration, see, e.g., [23]) or imposing too narrow design margins (see, e.g., [34, 32] for a nondiffracting beam design). In other problems, consistency cannot be easily confirmed a priori. For further discussion, see also [12].

When consistency is not guaranteed, feasible solutions may not exist, and the design objectives must be redefined. The most natural alternative is optimization: the introduction of a natural cost functional which describes the overall distance from complete feasibility and the search for a solution which minimizes this functional. Experience suggests a quadratic functional, such as $\hat{J}(h) = \sum \beta_i d^2(h, C_i)$, where β_i are positive constants left for interactive tuning and d is the Euclidean distance function, obtained via orthogonal projection of h onto C_i. A nonquadratic alternative was suggested in [25] for nonconvex problems, i.e., the same functional $\hat{J}(h)$ which, in the nonconvex case, may not be quadratic.

For various recent applications, this choice of cost function is too restrictive. In particular, one would like to be able to apply different measures of distance to the different sets C_i, i.e., minimize a functional of the more general form

$$\hat{J}(h) = \sum_{i=1}^{N} \beta_i d_i^2(h, P_i(h)), \tag{1}$$

where d_i are distance functions, chosen differently for different sets C_i, and $P_i(h)$ is the projection onto C_i with respect to d_i. We shall call such a problem *multidistance*, as opposed to the special case $d_i(h, g) = \|h - g\|$, which will be termed *unidistance*.

Strictly speaking, the multidistance problem is no longer purely "set theoretic": now the sets C_i are not the only data required, and one should also specify a priori the distance functions d_i. The goal of a multidistance algorithm is to approximate the optimal solution (minimize $\hat{J}(h)$ of (1)) using only the individual projections $P_i = P_{C_i}^{d_i}$ which project the current signal onto the element d_i-closest to it in C_i. In other words, the algorithm should be composed exclusively from the N possibly nonlinear, possibly nonorthogonal, projections P_i.

The interest in inconsistent and, in particular, multidistance problems is relatively new. We shall comment on this point in a while.

Whereas sequential algorithms fare well in a consistent environment, they are not suitable for inconsistent problems, since the iterates are confined to the sets C_i whereas the optimal solution may be found elsewhere. In some cases, this difficulty can be overcome by the use of special relaxation policies, though these lead to very slow convergence to the optimal solution [20, 6].

Moreover, weak convergence results for sequential algorithms are essentially restricted to the unidistance case. For *multidistance* projections, weak convergence is inconclusive, as stated in [34], and divergence is common, as a simple example in [22, Fig. 1] suggests. Further evidence for divergence was accumulated based on the papers [35, 21]. Some, severely restrictive, sufficient conditions for weak convergence (in the sequential algorithms) were studied in [22].

Based on these reservations, focus is restricted here to *parallel* algorithms involving the nonorthogonal projections P_i. It turns out that for an inconsistent problem which requires the minimization of $\hat{J}$ in (1) with weights β_i and the various distance functions d_i, the appropriate parallel algorithm should weight the different projections P_i exactly by the same weights β_i. The algorithm is given in detail in section 2.

Following this choice, it turns out that $\hat{J}$ is always a Lyapunov functional in the weak sense; i.e., its values are nonincreasing along iterates of the algorithm. This behavior is instrumental in, but not sufficient for, guaranteeing the weak convergence of the iterates, and some further assumptions must be made.

In studying the multidistance case, we were motivated by the results [22] of Kotzer, Cohen, and Shamir and the work of [7] by Censor and Elfving. Below we shall refer to a special case of the algorithm proposed in [7], to mitigate the difficulties in [22], as Algorithm I.

Compared with [12], the goals of [7] were rather limited: (i) only the consistent case was discussed; (ii) the signal space was real Euclidean space, as opposed to real or complex (infinite dimensional) Hilbert space (this is a severe limitation in many signal and image applications); (iii) relaxation was not incorporated in the algorithm. However, [12] does not allow the wide latitude of generalized distance functions treated in [7].

Some clarification may be necessary here concerning point (iii) above. Relaxation is a commonly used procedure for improving the convergence profile of many projection-based algorithms. Admittedly, the role of relaxation in the convergence pattern of projection-based algorithms has never been thoroughly analyzed. For serial algorithms, it has been demonstrated experimentally in [27, 8]. Recently, it was demonstrated in [14, 12] for Algorithm I, but assuming unidistance. See also [10] for an analytical study of relaxation.

Reference [7] admits only zero relaxation. Reference [12] admits time-varying relaxation, λ_i with $|\lambda_i| \leq 1 - \epsilon$, though the analysis pertains to the unidistance case only. We shall consider in our analysis the *multidistance* case, allowing a fixed relaxation value λ, with $|\lambda| < 1$.

Thus, our approach in the present paper combines the multidistance aspects of Censor and Elfving with the wide signal-oriented treatment of Combettes in a nontrivial way. One feature of [7] which we were forced to abandon was their latitude in choosing the distance functions d_i. The class they considered was the so-called class of Bregman distance functions, [3], which is so large that its elements are not even confined to the usual axioms of a metric, such as symmetry ($d(x, y) = d(y, x)$). The tremendous flexibility offered by this class is offset by one major disadvantage: due to its generality, it leaves implicit the mathematical expression for the iterated map representing Algorithm I (which is denoted by $P_{\beta,\lambda}$ in the present paper).

In variance with [7], we consider here only a relatively small subclass of Bregman distance functions in $L^2(\mathbb{R})$: the set of weighted L^2 norms in the "frequency domain" (i.e., after the application of the Fourier transform, which is denoted here by $\hat{}$):

$$d^2(f, g) = \int_{-\infty}^{+\infty} W(u)|\hat{f}(u) - \hat{g}(u)|^2 du. \tag{2}$$

This restraint allowed us to go beyond the analysis of Censor and Elfving and obtain an explicit expression for the map $P_{\beta,\lambda}$ representing Algorithm I (as well as obtaining weak convergence results even when C_o is empty). Consequently, we were able to characterize its set of fixed points. As mentioned earlier, this point is of crucial significance, since weak convergence to a fixed point can be guaranteed whenever the set of fixed points is not empty (even if C_o is empty).

It should be noted that the smaller set of distance functions used here is still sufficient for the analysis of many important multidistance problems. In particular, we were motivated by several problems in wave scattering analysis [21], design of filters for pattern recognition [22, 35, 26], restoration [36], and image restoration [23], to mention a few. In these problems, some of the sets C_i are known only implicitly:

$$C_i = \{f \in \mathcal{H} : \quad K_i(f) \in \hat{C}_i\}, \qquad \hat{C}_i \text{ is convex,}$$

where $\hat{C}_i$ is given explicitly; see [22]. In principle, K_i may be any affine operator but we restrict the discussion to the important class where K_i is a convolution operator. These problems are known as *constrained deconvolution* problems. A detailed discussion and analysis in [22] shows that problems of this type, which are handled relatively inefficiently by unidistance projection algorithms, can be solved efficiently by recasting the problem as multidistance. In particular, (various) weighted L^2 norms in the frequency domain (a different weighted norm for each convolution operator K_i). This can easily be justified by means of the Fourier transform; see Appendix A.

In analyzing Algorithm I, we use the so-called product space construction of Pierra, as do the papers [7, 12]. Our construction, however, contains a nontrivial topological extension, which is necessary to adopt this construction to a multidistance environment.

The structure of the paper is as follows: Algorithm I is defined and studied. Then the product space formalism is introduced and used to show that Algorithm I is nonexpansive in an appropriate sense. Monotone nonincrease with respect to $\hat{J}$ is established. Then we use functional analysis to characterize the set of fixed points of Algorithm I and to show that the algorithm converges weakly whenever this set is not empty. Sufficient conditions for the existence of fixed points are given.

Notationwise, we follow [7]. We shall use "$^\wedge$" and "$^\vee$" to denote the Fourier transform on $L^2(\mathbb{R})$ and its inverse, respectively. Indefinite integrals will invariably denote integration over the real line.

2. Algorithm I and its stability. In this section we describe our main algorithm and summarize its main stability and convergence properties.

Data. Given are N convex sets C_i, N respective norms d_i, N weights $\beta_i > 0$ with $\sum_{i=1}^N \beta_i = 1$, and one relaxation parameter, $\lambda \in [-1, 1]$. We denote by P_i the mapping of projecting onto C_i with respect to d_i. To include relaxation, we define $P_{i,\ \lambda}(h) = P^{d_i}_{C_i}(h) + \lambda(h - P^{d_i}_{C_i}(h))$. Relaxation is used extensively in the literature, for both theoretical and practical reasons, to enhance the convergence; see, e.g., [37, 27, 10]. In general, if T is a mapping, its relaxed version T_λ with relaxation parameter $\lambda \in \mathbb{R}$ is defined as $T_\lambda := T + \lambda(I - T)$, i.e., $T_\lambda(h) := T(h) + \lambda(h - T(h))$. Following Baillon, Bruck, and Reich, [1], whenever $\lambda \in (0, 1)$ we term T_λ an *averaged mapping.* For more details, see also [19, 10, 37].

We assume that the distance functions d_i are all of the *weighted* L^2 type described in (2). Let $W_i(t)$ be the corresponding weight functions. The sets C_i are assumed convex and closed with respect to d_i. β_i are tuning parameters, which are added only for practical design convenience. Mathematically, they may be absorbed in the weight functions W_i.

Algorithm I has the following simple formulation.

Initialization. An arbitrary initial function $h^o \in L^2(\mathbb{R})$.

The main step. Given the function $h^k \in L^2(\mathbb{R})$, calculate $h^{k+1} \in L^2(\mathbb{R})$ via

$$v_i^{k+1}(x) = P_{i,\ \lambda}(h^k), \qquad i = 1,\ 2, \ldots N, \tag{3a}$$

$$h^{k+1}(x) = \left\{ \left(\sum_{i=1}^{N} \beta_i W_i(u) \right)^{-1} \sum_{i=1}^{N} \beta_i W_i(u) \hat{v}_i^{k+1} \right\}^{\vee} . \tag{3b}$$

Recall that "$^\wedge$" and "$^\vee$" denote the Fourier transform and its inverse, respectively.

Algorithm I with multiple metrics d_i but with zero relaxation has already been successfully implemented in various signal synthesis and restoration tasks involving nonconvex constraints [26, 25] and inconsistent constraints [23]. In all cases, some of the original constraint sets were given implicitly.

The special unidistance case of Algorithm I with orthogonal projections onto the sets C_i has been studied before for consistent as well as inconsistent problems. See, e.g., [15, 16], [11, Section III.E], and [12].

2.1. Stability and convergence. To analyze the stability properties of Algorithm I, we use a Lyapunov theoretic approach. We shall call a "Lyapunov functional" any functional on $\mathcal{H}$ whose values are nonincreasing along iterates of Algorithm I. According to classical theory, the existence of such a functional is instrumental, but not sufficient, for weak convergence. In problems involving only convex sets, it makes sense to look for a convex functional, for which local minima must be global. The additional assumption of coercivity of the functional guarantees weak convergence to a global minimum. Strict convexity implies that the global minimum is unique.

Specifically for Algorithm I, it turns out (section 8) that the functional (1)

$$\hat{J}(h) = \sum_{i=1}^{N} \beta_i d_i^2(h, P_i(h))$$

is a convex Lyapunov functional which decreases strictly along nonstationary iterates (for other choices, see [28]). If the problem is consistent, Algorithm I converges weakly to an element in C_o (for which $\hat{J}$ is zero).

For an inconsistent problem, weak convergence of the iterates cannot be guaranteed in general. However, we find that the iterates generated by Algorithm I converge weakly to the set of global minimizers of $\hat{J}$, whenever this set is nonempty. In accordance with this finding, we find some general conditions guaranteeing weak convergence of the iterates even in the inconsistent formulation (Theorem 9.6 (b), (c), combined with Theorems 7.6 and 8.3). In most routine applications, it can be shown that $\hat{J}$ is also coercive and strictly convex. Coercivity is guaranteed if, e.g., any one of the sets C_i are d_i-bounded. For a typical illustration of an inconsistent situation, see Figure 1. Strict convexity is guaranteed if, e.g., all of the sets C_i are strictly convex.

To prove weak convergence of Algorithm I, one needs to go beyond stability. In this paper we analyze the properties of the mapping $P_{\beta,\lambda}$ which represents Algorithm I. It is shown that for all $\lambda \in (-1, 1)$, this mapping is nonexpansive and asymptotically regular. Moreover, simple sufficient conditions (Theorem 9.6) are provided which guarantee that this mapping has fixed points. According to classical theory, these three elements (nonexpansivity, asymptotic regularity, and the existence of fixed points) guarantee weak convergence.

2.2. A theoretical perspective. The problem is called *nonconvex* if at least one of the sets C_i is not convex. For nonconvex problems in pattern recognition and image restoration we refer the reader to [35, 26, 25] and [27]. Although a projection onto a nonconvex set is not well defined mathematically, it can usually be implemented without difficulty [27, 25].

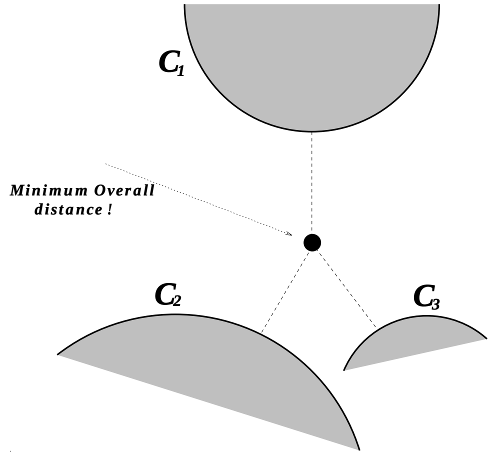


FIG. 1. *A typical global minimizer of $\hat{J}$ when C_o is empty.*

When the problem is inconsistent, or nonconvex, it may be impossible to establish weak convergence under general conditions. Instead, and as a first step towards convergence results, one would like to establish the weaker property of monotonicity with respect to a particular Lyapunov functional.

In both the inconsistent and nonconvex contexts for serial algorithms, a general Lyapunov monotonicity result is available only when $N = 2$, i.e., when only two constraints are involved. See [27, 25] for nonconvex problems and [9, 18] for inconsistent problems. These results are, moreover, restricted to unidistance problems.

In contrast, for Algorithm I, $\hat{J}$ still acts as a Lyapunov functional; i.e., its values are nonincreasing (and typically decreasing) along iterates, *independent* of the number N of sets involved, and include multidistance. This is established in [25] for the nonconvex case and in the present paper for the inconsistent case.

The distinction between $N = 2$ and $N > 2$ for serial algorithms is of theoretical significance since there is a special construction, called the *product space construction*, which can be used to transform any parallel multidistance algorithm, with N arbitrary, into an equivalent serial algorithm, with only two constraints, of a necessarily unidistance character. Thus, the distinction between $N = 2$ and $N > 2$ does not exist for parallel algorithms, in particular, for Algorithm I.

Formalization through the product space, introduced by Pierra [31], considered only finite-dimensional and unidistance problems to begin with. It was generalized to finite-dimensional multidistance problems in [7]. The product space is simply a Cartesian product of N copies of the original signal space. We develop an infinite-dimensional product space formalism, which requires some topological finesse: the product space is not a cartesian product of identical copies of the original signal space, since each copy is governed by a different distance function d_i, inducing a *possibly different* Hilbert space $\mathcal{H}_i$.

3. The product space. In this section we introduce the product space formulation to be used in subsequent sections. For concreteness, and for the purpose of one-dimensional continuous-time signal applications, we shall work with the space $\mathcal{H} = L^2(\mathbb{R})$.

Our data consists of N convex subsets C_i in $\mathcal{H}$, N essentially positive weight functions $W_i(x)$, and N positive constants β_i. We assume that C_i are d_i closed, where d_i are the weighted L^2 distance functions associated with W_i via (1). We also assume the normalization $\sum_{i=1}^{N} \beta_i = 1$.

Let $\mathcal{H}_i$ denote the closure of $\mathcal{H}$ with respect to the norm d_i; namely,

$$\mathcal{H}_i := \left\{ h \mid \int W_i(u)|\hat{h}(u)|^2 du < \infty \right\}.$$

Let $\langle \cdot, \cdot \rangle_i$ denote the natural (W_i-weighted) inner product on $\mathcal{H}_i$, i.e., $\langle f, g \rangle_i = \int W_i(u)\hat{f}(u)\hat{g}(u)du$. Recall that W_i is real, essentially bounded, and essentially positive. Therefore, $\mathcal{H}$ is a dense subspace in $\mathcal{H}_i$ (if, in addition, W_i is essentially bounded away from zero, the metric d_i is equivalent to the norm in $\mathcal{H}$, and $\mathcal{H}_i = \mathcal{H}$). The space

$$\mathcal{H}_o := \bigcap_{i=1}^{N} \mathcal{H}_i$$

may be described as the closure of $\mathcal{H}$ with respect to the overall weight function W_o and distance function d_o given by

$$W_o := \sum_{i=1}^{N} \beta_i W_i, \qquad d_o^2 = \sum_{i=1}^{N} \beta_i d_i^2.$$

The associated inner product will be denoted by $\langle \cdot, \cdot \rangle_0$. Again, $\mathcal{H}$ is a dense subspace of $\mathcal{H}_o$, and the two spaces coincide if and only if W_o is essentially bounded away from zero. Note that this coincidence, whenever it occurs, is independent of the choice of $\beta_i > 0$. In particular, note that if any one of the distance functions d_i is the (unweighted) Euclidean norm, $\mathcal{H} = \mathcal{H}_o$.

Define the product space of multiplicity N,

$$\mathbf{H} := \prod_{i=1}^{N} \mathcal{H}_i, \tag{4}$$

whose elements will be denoted by

$$\mathbf{h} = (h_1, h_2, \ldots, h_N)^T, \qquad h_i \in \mathcal{H}_i. \tag{5}$$

$\mathbf{H}$ is endowed with the following inner product and norm:

$$\ll \mathbf{h}, \mathbf{h}' \gg = \sum_{i=1}^{N} \beta_i \langle h_i, h_i' \rangle_i, \qquad |||\mathbf{h}|||^2 = \ll \mathbf{h}, \mathbf{h} \gg . \tag{6}$$

Clearly, the product space $\mathbf{H}$ is a Hilbert space with respect to this structure. The associated distance function is

$$D(\mathbf{h}, \mathbf{h}') := |||\mathbf{h} - \mathbf{h}'|||.$$

The space $\mathcal{H}$ and, more generally, the space $\mathcal{H}_o$ will be embedded in $\mathbf{H}$ via the linear "duplication operator"

$$\tau(h) := (\underbrace{h, h, \ldots h}_{N \text{ times}})^T; \qquad h \in \mathcal{H}_o. \tag{7}$$

The linear subspace $\mathbf{\Delta} = \tau(\mathcal{H}_o)$ will be referred to as the *diagonal subspace*. It can be seen that $\mathbf{\Delta}$ is D-closed.

The mapping $\tau : \mathcal{H} \to (\mathbf{H}, D)$ is well defined but not necessarily isometric. Its image is dense in $\mathbf{\Delta}$. However, let us change the metric of the input space. It is easy to check that the mapping $\tau : (\mathcal{H}_o, d_o) \to (\mathbf{\Delta}, D)$ is isometric and, in fact, unitary. We shall later use the isometric properties of the densely restricted operator $\tau : (\mathcal{H}, d_o) \to (\mathbf{\Delta}, D)$.

4. The mapping $P_{\beta,\lambda}$. In this section we define our main objective as a mapping $P_{\beta,\lambda}$ on $\mathcal{H}$ whose iterative action embodies Algorithm I. As a first step, we define the auxiliary mapping $\mathbf{Q} : \mathbf{H} \to \mathcal{H}$ and the auxiliary energy functionals $\psi_{\mathbf{h}}$ on $\mathcal{H}$ for all $\mathbf{h} = (h_1, \ldots, h_N) \in \mathbf{H}$ via

$$\mathbf{Q}(\mathbf{h}) = \left\{ \sum_{i=1}^{N} \beta_i \frac{W_i(x)}{W_o(x)} \hat{h}_i(u) \right\}^{\vee}, \qquad \psi_{\mathbf{h}}(h') := \sum_{i=1}^{N} \beta_i d_i^2(h', h_i). \tag{8}$$

The following lemma gives a nice variational characterization for the mapping $\mathbf{Q}$.

LEMMA 4.1. (i) $g = \mathbf{Q}(\mathbf{h})$ *if and only if* g *is a global minimum of* $\psi_{\mathbf{h}}$.

(ii) $\mathbf{Q}$ *is a well defined mapping on* $\mathbf{H}$.

Proof. (i): It can be seen by direct calculation that ψ_h is strictly convex; hence, it can have at most one global minimum. The global minimizer $g \in \mathcal{H}$, if it exists, is the only point at which the first variation (or *Gâteaux* derivative [17, p. 23]) $\nabla\psi_h$ is zero. In our case of weighted L^2 norms, a routine computation shows that

$$\nabla\psi_{\mathbf{h}}(g) = 2\sum_{i=1}^{N} \beta_i \{W_i(\hat{g} - \hat{h}_i)\}^{\vee}. \tag{9}$$

Now, by equating the first variation to zero, we obtain the equality $g = \mathbf{Q}(\mathbf{h})$, as required.

(ii): We need to show that for any $\mathbf{h} \in \mathbf{H}$ the function $g = \mathbf{Q}(\mathbf{h})$ is both measurable and square integrable. To show measurability, it suffices to note that in (8) both the numerator and denominators of the expression for $\mathbf{Q}(\mathbf{h})$ are measurable, and the denominator is nonzero almost everywhere (a.e.). To show square integrability, define the function $H = \max_{1\le i\le N} |\hat{h}_i| \in L^2(\mathbb{R})$. By Parseval's identity and the triangle inequality we get

$$\begin{aligned} \int |g(x)|^2 dx &= \int |W_o^{-1}(u) \sum \beta_i W_i(u) \hat{h}_i(u)|^2 du \\ &\le \int |W_o^{-1} \sum \beta_i W_i(u) H(u)|^2 du = \int |H(u)|^2 du < \infty\,, \end{aligned}$$

showing that g is square integrable, i.e., $g \in \mathcal{H}$. □

If ψ is also coercive on $\mathcal{H}$, it is a priori clear that its minimizer is in $\mathcal{H}$, in which case (ii) follows automatically from (i).

Define the mapping $P_\beta : \mathcal{H} \to \mathcal{H}$ and the functional φ_h ($h \in \mathcal{H}$) on $\mathcal{H}$ via

$$P_\beta(h) = \mathbf{Q}(P_1(h), \ldots, P_N(h)), \qquad \varphi_h(g) = \sum_{i=1}^{N} \beta_i d_i^2(g, P_i(h)). \tag{10}$$

It can be seen from the definition that P_β represents Algorithm I with zero relaxation. Adding relaxation in Algorithm I amounts to adding the same relaxation to P_β, yielding the relaxed mapping

$$P_{\beta,\lambda}(h) := P_\beta(h) + \lambda(h - P_\beta(h)). \tag{11}$$

We note that indeed the image of $P_{\beta,\lambda}$ is in $\mathcal{H}$ (although $\mathbf{Q}$ is defined on $\mathbf{H}$, which contains $\prod_{i=1}^{N} \mathcal{H}$), due to the fact that $g = \mathbf{Q}(\mathbf{h})$ is square integrable for *any* $\mathbf{h} \in \mathbf{H}$, in particular for $\mathbf{h} = \tau(h)$; $h \in \mathcal{H}$. In any event, the mapping P_β can be defined also via $P_\beta : \mathcal{H}_o \to \mathcal{H}_o$, due to Lemma 4.1(ii). This will be used in various theorems used in section 7.

COROLLARY 4.2. (i) *$g = P_\beta(h)$ if and only if g minimizes φ_h.*

(ii) *P_β is a well defined mapping on $\mathcal{H}$.*

4.1. A theoretical perspective. Although P_β is in general not a projection, it shares with the projections several essential properties. For a projection of the form $P = P_C^d$, the following hold:

(a) Given h, the vector $g = P(h)$ trivially minimizes the associated strictly convex coercive energy functional $d^2(g, P(h))$. This may be regarded as a "(trivial) variational characterization" of P.

(b) The fixed points of P are the elements of C, i.e., the global minimizers of the functional $h \to d^2(h, C)$.

(c) Every sequence of iterates of P converges weakly to C.

(d) For any relaxation $\lambda \in (-1, 1]$, the relaxed projection is nonexpansive and its iterates still converge weakly to the same set of fixed points of the unrelaxed operator (we remark that for a general nonexpansive operator in Hilbert space, only relaxations in $[0, 1)$ are automatically guaranteed to be nonexpansive). We note that attribute (c) above is trivial for a projection, and it was stated only because it holds for P_β too, with C appropriately defined.

For the mapping P_β, Corollary 4.2(i) provides the variational characterization analogous to (a) above. Concerning property (b), it will be shown in section 8 that the fixed points of $P_{\beta,\lambda}$ are indeed the global minimizers of the functional $\hat{J}$ in (1), independent of the relaxation. However, their existence is not a priori guaranteed, as in the case of a projection. Concerning properties (c, d), nonexpansivity of $P_{\beta,\lambda}$ for all $\lambda \in [-1, 1]$ will be established in section 6, and sufficient conditions guaranteeing weak convergence for all $\lambda \in (-1, 1]$ will be given in section 9.

5. The cyclic algorithm on the product space.

We now describe the dynamics of Algorithm I in the product space $\mathbf{H}$ defined in section 3. What emerges is a cyclic unidistance projection algorithm on the product space $\mathbf{H}$, involving two projections $P_{\mathbf{C}}^D$ and $P_{\mathbf{\Delta}}^D$.

The metric D and the diagonal subspace $\mathbf{\Delta}$ were already defined in section 3. The convex set $\mathbf{C}$ is just the direct sum

$$\mathbf{C} = \prod_{i=1}^{N} C_i.$$

It can be checked that $\mathbf{\Delta}$ and $\mathbf{C}$ are closed and convex and are considered subsets of $(\mathbf{H}, D)$. Moreover, recalling that $C_o = \cap_{i=1}^{N} C_i$, we find that $\mathbf{\Delta} \cap \mathbf{C} = \tau(C_o)$. Namely, the sets C_i intersect in $\mathcal{H}$ if and only if $\mathbf{\Delta}$ and $\mathbf{C}$ intersect in $\mathbf{H}$.

Together with the two sets $\mathbf{C}$ and $\mathbf{\Delta}$ we associate the two respective projections $P_{\mathbf{C}}^D$ and $P_{\mathbf{\Delta}}^D$, defined with respect to the distance function D. Considered as mappings on $(\mathbf{H}, D)$, these are two *orthogonal* projections; hence, in particular, the context is unidistance. This point will be of great significance later on. Below we provide a more concrete description of the two projection mappings.

LEMMA 5.1. *Under the above construction, we have*

$$P^D_{\mathbf{C}}(\mathbf{h}) = (P_1(h_1), \quad P_2(h_2), \ldots, P_N(h_N))^T, \tag{12a}$$

$$P^D_{\boldsymbol{\Delta}}(\mathbf{h}) = \tau \circ \mathbf{Q}(\mathbf{h}), \tag{12b}$$

where $\mathbf{h} = (h_1, \ldots, h_N)^T$.

Proof. Equations (12a, 12b) are direct consequences of the definition of the sets $\mathbf{C}, \boldsymbol{\Delta}$ and the metric D (on $\mathbf{H}$). In particular, to derive (12b) we note that, by definition, we have the equality $D(\mathbf{h}, \tau(g)) = \psi_{\mathbf{h}}(g)$. Also by definition, projecting a vector $\mathbf{h} \in \mathbf{H}$ orthogonally onto $\boldsymbol{\Delta}$ amounts to minimizing $D(\mathbf{h}, \tau(g))$ over all $\tau(g)$. By Lemma 4.1, applying $\mathbf{Q}$ amounts to minimizing the same expression over all g. Thus, (12b) must hold. □

Define the composed mapping on $\mathbf{H}$ by

$$P^c(\mathbf{h}) = P^D_{\boldsymbol{\Delta}} \circ P^D_{\mathbf{C}}\,(\mathbf{h}). \tag{13}$$

Using Corollary 4.2 and Lemma 5.1, it is easy to establish the identity

$$P^c \circ \tau = \tau \circ P_\beta. \tag{14}$$

We interpret this identity as follows: without relaxation, Algorithm I, defined on $\mathcal{H}$, manifests itself in the product space $\mathbf{H}$ as a simple *cyclic* algorithm involving two orthogonal projections, again with zero relaxation. We emphasize that even if Algorithm I is multidistance and involves $N > 2$ convex sets, the product space algorithm is unidistance and involves only two convex sets.

Adding relaxation to Algorithm I amounts to adding the same relaxation to P^c or, equivalently, to $P^D_{\boldsymbol{\Delta}}$. This follows from an examination of (14). We therefore conclude that Algorithm I with relaxation manifests itself in the product space as a cyclic algorithm involving two projections, one relaxed and one unrelaxed.

6. Establishing nonexpansivity. It will be shown below that the dynamics of Algorithm I are nonexpansive with respect to the metric d_o on $\mathcal{H}$. To avoid notational inconvenience, the zero subscript in d_o will be omitted throughout.

THEOREM 6.1. (i) ${P^c}_\lambda(\mathbf{h})$ *is D-nonexpansive for all* $\lambda \in [-1, 1]$.

(ii) $P_{\beta,\lambda}(h)$ *is d-nonexpansive for all* $\lambda \in [-1, 1]$.

(iii) *In particular,* P_β *and* P^c *are nonexpansive in the appropriate metrics.*

Note that in general (iii) implies (i, ii) automatically only for $\lambda \in [0, 1]$.

Proof. Using the isometric properties of τ in (14), it is enough to establish (iii) only for P^c and then to establish only (ii).

To prove (iii) for P^c, note that $P^D_{\boldsymbol{\Delta}}$ and $P^D_{\mathbf{C}}$ are orthogonal projections with respect to D; hence, they are nonexpansive with respect to this metric. Therefore, their composition P^c is also D-nonexpansive.

To prove (ii), let $h_1, h_2 \in \mathcal{H}$ be arbitrary but fixed. Define the vectors

$$k_m := P_{m,\lambda}(h_1) - P_{m,\lambda}(h_2), \qquad m = 1, 2, \ldots, N.$$

Then we have

$$d^2(P_{\beta,\lambda}(h_1), P_{\beta,\lambda}(h_2)) = \sum \beta_i d_i^2(P_{\beta,\lambda}(h_1), P_{\beta,\lambda}(h_2))$$

$$
\begin{aligned}
&= \sum_{i=1}^{N} \beta_i \int W_i(u) \left| \frac{\sum_{m=1}^{N} \beta_m W_m(u) \hat{k}_m(u)}{\sum_{m=1}^{N} \beta_m W_m(u)} \right|^2 du \\
&\leq \int \sum_{i=1}^{N} \beta_i W_i(u) \left(\frac{\sum_{m=1}^{N} \beta_m W_m(u) |\hat{k}_m(u)|}{\sum_{m=1}^{N} \beta_m W_m(u)} \right)^2 du \\
&= \int \frac{\left(\sum_{m=1}^{N} \beta_m W_m(u) |\hat{k}_m(u)|\right)^2}{\left(\sum_{m=1}^{N} \beta_m W_m(u)\right)} du \,.
\end{aligned}
$$

Define the functions $Q_i(u) = \sqrt{\beta_i W_i(u)}$ and $S_i(u) = \sqrt{\beta_i W_i(u)} |\hat{k}_m(u)|$. The Cauchy–Schwarz inequality $\left[\sum_{i=1}^{N} Q_i(u) S_i(u)\right]^2 \leq \sum_{i=1}^{N} Q_i^2(u) \sum_{i=1}^{N} S_i^2(u)$ can be written as

$$
\left(\sum_{m=1}^{N} \beta_m W_m(u) |\hat{k}_m| \right)^2 \leq W_o(u) \sum_{m=1}^{N} \beta_m W_m(u) |\hat{k}_m|^2.
$$

Consequently, we have

$$
\begin{aligned}
&\int \frac{\left(\sum_{m=1}^{N} \beta_m W_m(u) |\hat{k}_m(u)|\right)^2}{W_o(u)} du \leq \int \sum_{m=1}^{N} \beta_m W_m(u) |\hat{k}_m(u)|^2 \\
&= \sum_{m=1}^{N} \beta_m d_m^2(\hat{k}_m, 0) = \sum_{m=1}^{N} \beta_m d_m^2(k_m, 0) \leq \sum_{m=1}^{N} \beta_m d_m^2(h_1 - h_2, 0) \,.
\end{aligned}
$$

To justify the last inequality above, note that since P_m is d_m-nonexpansive, its relaxed version $P_{m,\lambda}$ is automatically nonexpansive for all $0 \leq \lambda \leq 1$. In fact, since P_m is also an orthogonal projection on the Hilbert space $\mathcal{H}_m$, $P_{m,\lambda}$ is nonexpansive for all $-1 \leq \lambda \leq 1$.

Finally, combining the above chains of inequalities, we obtain

$$
d(P_{\beta,\lambda}(h_1), P_{\beta,\lambda}(h_2)) \leq d(h_1, h_2) \qquad \forall |\lambda| \leq 1,
$$

implying that $P_{\beta,\lambda}$ is nonexpansive with respect to the metric d. □

7. Establishing weak convergence.

Let $\mathcal{H}_o$ be as defined in section 3.

DEFINITION 7.1. *Denote by F the set of fixed points of P_β in $\mathcal{H}$.*

In the next section we shall bring evidence to the fact that F (the set of fixed points of P_β in $\mathcal{H}$) is not empty in most cases of interest. In this section we shall show that nonemptiness of F implies the weak convergence of Algorithm I. The proof follows from combining two classical results of Opial and Browder concerning asymptotic regularity.

DEFINITION 7.2. *An operator $T : \mathcal{H} \rightarrow \mathcal{H}$ is termed asymptotically regular if* $\lim_{k\to\infty} \parallel T^{k+1}(h) - T^k(h) \parallel \rightarrow 0 \quad \forall h \in \mathcal{H}$.

THEOREM 7.3 (Opial, [30, Theorem 1]). *Let C be a closed convex set in a Hilbert space $\mathcal{H}$ and let $T : C \rightarrow C$ be a nonexpansive asymptotically regular mapping for which the set F_T of fixed points is nonempty. Then, for any h in C, the sequence of successive approximations $\{h^o, h^1 \ldots\}$, where $h^{n+1} := Th^n$, is weakly convergent to an element of F_T.*

Theorem 7.4 (Browder, [4, Theorem 5]). *Let X be a uniformly convex Banach space, and let $T : X \to X$ be a nonexpansive mapping with a nonempty set F_T of fixed points. For a given constant γ, $0 < \gamma < 1$, let $S_\gamma = \gamma I + (1-\gamma)T$, i.e., $S_\gamma(h) = \gamma h + (1-\gamma)T(h)$. Then S_γ is asymptotically regular and has the same set of fixed points as T.*

We may sharpen Theorem 7.4 slightly, as follows.

Corollary 7.5. *If, in the same setup, T and all its relaxations in the interval $[a, 1]$ are nonexpansive and F_T is nonempty, then all the relaxations of T in the open interval $(a, 1)$ are nonexpansive and asymptotically regular and have the same set of fixed points as T.*

Proof. It is easy to see that any relaxed mapping T_λ with $a < \lambda < 1$ can be represented as an averaged mapping based on T_a, i.e., a convex combination of T_a and $T_1 = I$. Indeed, we have

$$T_\lambda = \gamma I + (1-\gamma)T_a; \quad \gamma = \frac{\lambda - a}{1 - a}$$

and $\gamma \in (0, 1)$. Now apply Theorem 7.4 on the operator T_a and its relaxation T_λ. It follows that T_λ is asymptotically regular . ◻

We are now able to state and prove our main result concerning weak convergence of Algorithm I.

Theorem 7.6. *The relaxed mapping $P_{\beta,\lambda}$ defined in section 4 is nonexpansive and asymptotically regular for all $\lambda \in (-1, 1)$ (where nonexpansivity and asymptotic regularity are in $(\mathcal{H}_o, d_o)$). Thus, for any initial point $h^o \in \mathcal{H}$, the sequence of successive approximations $\{h^o,\ h^1, \ldots\}$, where $h^{k+1} := P_{\beta,\lambda}(h^k)$, converges weakly to an element of F, whenever F is not empty.*

Proof. The underlying Banach space in our case is $(\mathcal{H}_o, d_o)$, as defined in section 3. Theorem 6.1 guarantees nonexpansivity of $P_{\beta,\lambda}$. Since $\mathcal{H}_o$ is a Hilbert space, it is uniformly convex, and Corollary 7.5 can be applied to the mapping $T = P_\beta$ with $a = -1$, completing the proof of asymptotic regularity of $P_{\beta,\lambda}$.

Moreover, by Corollary 7.5, the set of fixed points of $P_{\beta,\lambda}$ is the same as the set of fixed points of P_β, i.e., F. Thus, assuming F is nonempty and using Theorem 7.3, iterations of $P_{\beta,\lambda}$ converge weakly to an element of F from any initial function. ◻

Concerning Theorem 7.6, we make the following remarks: (1) The extreme relaxation values $\lambda = \pm 1$ are not included in the Theorem, and $\lambda = -1$ may indeed lead to nonconvergence, e.g., in the trivial case $N = 1$ where P_β is a projection. (2) In principle, we have only established that the weak limit point is in $\mathcal{H}_o$. However, since the image of P_β is in $\mathcal{H}$, the weak convergence is, in fact, to an element in $\mathcal{H}$ [although the underlying Banach space in our case is $(\mathcal{H}_o, d_o)$]. (3) F is independent of relaxation, but may depend on β_i (if C_o is empty). This may be used to advantage: if we wish the weak limit point to be closer to one of the sets C_i, we can increase the relative value of β_i.

An alternative proof of Theorem 7.6 would be to demonstrate weak convergence on the product space mapping P^c, rather than on $P_{\beta,\lambda}$, since the convergence pattern of these two mappings is the same. Indeed, one could split P^c into its two constituents, $P_{\mathbf{\Delta},\lambda}$ and $P_{\mathbf{C}}$, and use classical convergence results for cyclic projection algorithms. For example, the paper of Cheney and Goldstein [9, Theorem 4] guarantees that iterates of the composed mapping P^c converge (weakly) to a fixed point of $P_{\mathbf{\Delta}} P_{\mathbf{C}}$, say $\mathbf{h} \in \mathbf{H}$, if such a point exists. In fact, $\mathbf{h} \in \mathbf{\Delta}$ (see also [9, Theorem 2]); hence, $\mathbf{h} = \tau(\mathbf{h})$ for some $h \in \mathcal{H}$, and h must be a fixed point of P_β. However, we opted our approach, due to its generality, i.e., only requiring that the operator is nonexpansive, asymptotically regular, and has a nonempty set of fixed points, not requiring the operator to be a projection operator. Indeed, the operator $P_{\beta,\lambda}$ is only a *projection-based* operator, not a projection operator.

A closely related approach is given in [12], using several classical results on firmly nonexpansive mappings. This approach is more general in that it admits time-varying relaxation (λ_i with $|\lambda_i| \le 1-\epsilon$). Moreover, [12] presents a special variant of Algorithm I which converges *strongly* without any special assumptions on the sets C_i, provided F is not empty. However, *unidistance* projections are assumed in [12].

Theorem 7.6 provides an alternative proof of convergence of the Censor–Elfving method [7], at least in the case of weighted L^2 distance functions. In fact, their result is generalized here in several directions: (1) the space $\mathcal{H}$ may be infinite dimensional; (2) consistency is not assumed; (3) it may be real or complex; (4) relaxation values ($\lambda \in (-1,1)$) are allowed. Note that the consistency assumption made in [7] implies the existence of a fixed point.

8. Characterization of the fixed points. To complete our analysis, it only remains to check that F, the set of fixed points of P_β in $\mathcal{H}$, is nonempty. While nonemptiness of F is expected in all practical situations, it cannot be easily guaranteed a priori. However, an indirect variational characterization exists for the set F, in terms of the nonnegative convex functional $\hat{J}$ on $\mathcal{H}_o$ defined in equation (1). Namely, it will be shown in Theorem 8.3 below that F coincides with the set G of global minimizers of $\hat{J}$ in $\mathcal{H}_o$ (and in particular it will follow that $G \subset \mathcal{H}$).

This characterization of F is in full agreement with the *consistent* case, where the set $C_o := \cap_{i=1}^{N} C_i$ is nonempty. It is known that in this case the process always converges weakly to an element in C_o. Indeed, $\hat{J}(h) = 0$ exactly when $h \in C_o$, and so we have $F = G = C_o$, and weak convergence to C_o is guaranteed.

In fact, it is intuitively quite clear that G is not empty also in many inconsistent situations. The weak limit point in this case need not belong to any of the sets C_i but is the "closest" point to these sets in an averaged sense, i.e., is a global minimizer of $\hat{J}$; see, e.g., Figure 1.

$\hat{J}$ is a strict Lyapunov functional for the *relaxed* mapping $P_{\beta,\lambda}$ ($\lambda \in (-1,1)$); i.e., it is strictly decreasing along iterates which are not fixed points. The discussion here follows closely with [15, 12].

LEMMA 8.1. *For any $h \in \mathcal{H}$ we have*

$$\hat{J}(P_{\beta,\lambda}(h)) \le \hat{J}(h) - (1-\lambda^2) d^2(h, P_\beta(h)). \tag{15}$$

Proof. Since $P_i\{P_{\beta,\lambda}(h)\}$ is the closest element to $P_{\beta,\lambda}(h)$ in C_i (in the d_i sense), we have

$$\begin{aligned} \|P_i P_{\beta,\lambda}(h) - P_{\beta,\lambda}(h)\|_i^2 &\le \|P_i(h) - P_{\beta,\lambda}(h)\|_i^2 \\ &= \|P_i(h) - h\|_i^2 + \|h - P_{\beta,\lambda}(h)\|_i^2 - 2\mathrm{Re}\,\langle P_i(h) - h,\ P_{\beta,\lambda}(h) - h\rangle_i. \end{aligned} \tag{16}$$

Summing over i, using the definition of P_β, the definition of the ith inner product, and some algebra, we obtain

$$\hat{J}(P_{\beta,\lambda}(h)) = \sum_{i=1}^{N} \beta_i \|P_i\{P_{\beta,\lambda}(h)\} - P_{\beta,\lambda}(h)\|_i^2 \le \sum_{i=1}^{N} \beta_i \|P_i(h) - h\|_i^2$$

$$
\begin{aligned}
&+\sum_{i=1}^{N}\beta_i\|h-P_{\beta,\lambda}(h)\|_i^2-2\sum_{i=1}^{N}\beta_i\mathrm{Re}\,\langle P_i(h)-h,\ P_{\beta,\lambda}(h)-h\rangle_i\\
\le&\sum_{i=1}^{N}\beta_i\|P_i(h)-h\|_i^2+\sum_{i=1}^{N}\beta_i\|h-P_{\beta,\lambda}(h)\|_i^2\\
&-2\sum_{i=1}^{N}\beta_i\mathrm{Re}\,\langle P_\beta(h)-h,\ P_{\beta,\lambda}(h)-h\rangle_i\\
\le&\sum_{i=1}^{N}\beta_i\{\|P_i(h)-h\|_i^2+[(1-\lambda)^2-2(1-\lambda)]\|P_\beta(h)-h\|_i^2\}\,.
\end{aligned}
$$

By rearranging and using the definition of d and $\hat{J}$, we obtain (15). □

For $\lambda = 0$ (no relaxation), Lemma 8.1 guarantees the largest decrease rate bound for $\hat{J}$:

$$\hat{J}(P_\beta(h)) \le \hat{J}(h) - d^2(h, P_\beta(h)). \tag{17}$$

We will also need the following elementary result.

LEMMA 8.2. (i) *$\hat{J}$ is a convex functional on $\mathcal{H}_o$.*

(ii) *If the problem is inconsistent (i.e., $C_o := \bigcap_{i=1}^{N} C_i = \emptyset$) and all sets C_i are strictly convex in $\mathcal{H}_i$, then $\hat{J}$ is strictly convex.*

Proof. We have the easily verifiable identity

$$2d_i^2\left(\frac{g+h}{2},\frac{Pg+Ph}{2}\right)+2d_i^2\left(\frac{g+Pg}{2},\frac{h+Ph}{2}\right)=d_i^2\left(g,Pg\right)+d_i^2(h,Ph)\,.$$

Using this identity, we conclude that

$$d_i^2\left(\frac{g+h}{2},P\left(\frac{g+h}{2}\right)\right)\le d_i^2\left(\frac{g+h}{2},\frac{Pg+Ph}{2}\right)\le\frac{d_i^2(g,Pg)+d_i^2(h,Ph)}{2};$$

hence, by summation, $\hat{J}$ is convex. If g, h are not both in C_i and C_i is strictly convex in $\mathcal{H}_i$, the left inequality (as well as the right inequality) is strict, and $\hat{J}$ shows strict convexity as required. However, if both g, h are inside C_i this argument is not valid.

The inconsistency assumption implies that indeed for some i either g or h does not belong to C_i, completing the argument. □

We are now able to prove the variational characterization proposed earlier.

THEOREM 8.3. *Let G be defined as the (possibly empty) set of global minimizers of $\hat{J}$ in $\mathcal{H}_o$. Then $G \in \mathcal{H}$. Moreover, G coincides with the set F of fixed points of the mapping P_β in $\mathcal{H}$.*

Proof. (i) First we show that $G \subset F$. Take $h \in G$. By definition, $\hat{J}(h) \le \hat{J}(P_\beta(h))$. However, from equation (17) we obtain the opposite inequality, hence, $\hat{J}(h) = \hat{J}(P_\beta(h))$. Again by equation (17), it follows that $d(h, P_\beta(h)) = 0$, i.e., $h = P_\beta(h)$ as required.

(ii) Now we show that $F \subset G$. Take $f \in F$, $h \in \hat{\mathcal{H}}$. Assume by contradiction that $\hat{J}(h) < \hat{J}(f)$. Consider the set

$$A = \{h_1 \in \mathcal{H} \mid \hat{J}(h_1) \le \hat{J}(h)\}.$$

A is a closed convex set due to the continuity and convexity of $\hat{J}$ (see Lemma 8.2). Let f' be the unique closest element to f in A (in the d sense), i.e., $f' = P_A^d(f)$. Define

$$k = P_\beta(f') - P_\beta(f), \qquad m_i = P_i(f') - P_i(f).$$

Using equation (17) we have $\hat{J}(P_\beta(f')) \leq \hat{J}(f')$; hence, $P_\beta(f') \in A$. Since P_β is nonexpansive, we have $d(P_\beta(f'), f) \leq d(f', f)$. However, the opposite inequality holds in the strict sense; i.e., from the definition of f',

$$d(P_\beta(f'), f) \geq d(f', f) \qquad \text{with equality only if} \quad P_\beta(f') = f'.$$

This follows from the optimal choice of f' relative to f. From this it is concluded that $P_\beta(f') = f'$, and since $f \in F$, we get

$$d^2(f, f') = d^2(P_\beta(f), P_\beta(f')) = \sum_{i=1}^{N} \beta_i \|k\|_i^2 = \sum_{i=1}^{N} \beta_i \, \langle k, m_i \rangle_i \ \leq \sum_{i=1}^{N} \beta_i \|k\|_i \|m_i\|_i.$$

The last inequality is due to the Cauchy–Schwarz inequality applied to each d_i. Next, since P_i and P_β are contractive mappings, we get

$$d^2(f, f') \leq \sum_{i=1}^{N} \beta_i \|f' - f\|_i^2 = d^2(f, f'). \tag{18}$$

Hence, by Lemma 4.2 and section 4.1 for all i, the inequality $\|f' - f\|_i = \|P_i(f') - P_i(f)\|_i$ holds if and only if $f' - f = P_i(f') - P_i(f)$. That is, $f - P_i(f) = f' - P_i(f')$ holds for all i, and $\hat{J}(f) = \hat{J}(f') \leq \hat{J}(h)$, a contradiction. Hence, $\hat{J}(f) \leq \hat{J}(h)$ for all $h \in \mathcal{H}$, i.e., $f \in G$.

Finally, from (i) and (ii) we conclude that $F = G$. □

9. Sufficient conditions.

We shall give some sufficient conditions ensuring that the set F of fixed points is nonempty. By Theorem 7.6, this nonemptiness guarantees weak convergence of Algorithm I.

DEFINITION 9.1. *A functional $Q : C \to [-\infty, \infty]$ is termed coercive (over C) if it satisfies*

$$\lim_{\|c_j\| \to \infty} Q(c_j) \to \infty \quad \textit{for all} \quad c_j \in C$$

and proper if it satisfies the following: (a) it does not assume the value $-\infty$ anywhere and (b) it is not identically $+\infty$.

We shall make use of the following theorem.

THEOREM 9.2 (Ekeland, [17, pp. 33–44]). *Let C be a closed convex subset of $\mathcal{H}$. Let Q be a convex, lower-semicontinuous and coercive proper functional on C. Then the set of global minimizers of Q over C is nonempty.*

With this theorem we may establish the following theorem.

LEMMA 9.3. *Let at least one of the sets C_i be bounded with respect to the metric d_i. Then the set G is nonempty.*

Proof. If C_i is bounded with respect to d_i then the functional $J_i(h) = d_i(h, P_i(h))$ is coercive. Assuming $\beta_i > 0$, it follows that $\hat{J}$ is coercive. The other properties of the functional $\hat{J}$ can be routinely established. □

Another important sufficient condition may be based on the following result.

Theorem 9.4 (Baillon, [1, Corollary 2.2]). *Let $T : C \to C$ be a nonexpansive mapping on a subset of a uniformly convex Banach space. Then for any $\lambda \in (0,1)$, the averaged mapping T_λ does not have any fixed point if and only if $\lim_{k\to\infty} \|T_\lambda^k(x)\| = \infty$ for all $x \in C$.*

It may be easily shown that for any $\lambda \in (-1,1)$, $P_{\beta,\lambda} = \gamma I + (1-\gamma)P_{\beta,\mu}$ for some $0 < \gamma < 1$ and $\mu \in (-1,1)$ (see proof of Corollary 7.5). In other words, every relaxation of P_β (and P_β itself) is an averaged mapping. Therefore, since a Hilbert space is a uniformly convex Banach space, we obtain the following corollary.

Corollary 9.5. *$P_{\beta,\lambda}$, $\lambda \in (-1,1)$, has a nonempty set of fixed points if there exists $h \in \mathcal{H}$ such that*

$$\lim_{k\to\infty} \|P_{\beta,\lambda}^k(h)\|_0 < \infty,$$

i.e., the iterates are bounded for some $h \in \mathcal{H}$.

This result is of paramount practical value, since nondivergence of a sequence can be easily detected in practice.

The following result is a summary of the sufficient conditions obtained so far.

Theorem 9.6. *Let any one of the following conditions be satisfied:*

(a) $C_o = \bigcap_{i=}^N C_i \neq \emptyset$.

(b) *At least one of the sets C_i is bounded in the respective norm $\|\cdot\|_i$.*

(c) *There exists $h \in \mathcal{H}$ such that $\lim_{k\to\infty} \|P_{\beta,\lambda}^k(h)\|_0 < \infty$.*

Then G is nonempty. Moreover, if C_o is empty and all sets C_i are strictly convex, G is at most a singleton.

Proof. Item (a) follows from the fact that inf $\hat{J}$ $(= 0)$ is obtained with functions in C_o, i.e., $G = C_o$. Item (b) follows from Lemma 9.3 and item (c) from Corollary 9.5. Finally, the last statement follows from Lemma 8.2 and the fact that any strictly convex function has at most one minimizer. □

Our results in this section generalize results of De Pierro and Iusem [15, 16] and Combettes [11, 12] to a *multidistance* problem in an *infinite-dimensional complex Hilbert space*. In the special case of unidistance projections, Theorem 9.6 is in agreement with results established in [16, Lemma 17] and noted in [11, section IV.C] and [12].

10. Conclusions and remarks. This paper considers feasibility problems which are both inconsistent and multidistance, with possibly many convex constraints. It is demonstrated that sequential projection algorithms are not proper for such a problem. An alternative *parallel* algorithm (Algorithm I) is proposed, and it is shown that whenever it converges weakly, the solution is optimal, in the sense that it is a global minimizer of a certain functional which averages the squared distances to the various constraint sets. Moreover, we show that weak convergence is guaranteed whenever a fixed point of the algorithm exists, and we give ample evidence that in general such a fixed point does exist (Theorem 9.6). In particular, whenever the problem is consistent, we prove weak convergence to C_o.

We use the product space formalism of Pierra in an infinite-dimensional setting to demonstrate that the multidistance algorithm (Algorithm I) is still equivalent (up to Hilbert space intertwining) to a relaxed unidistance cyclic algorithm involving two orthogonal projections.

For concreteness, our formulation is done in the space $L^2(\mathbb{R})$ of continuous-time one-dimensional signals. Similar formulations can easily be derived in other cases of interest, e.g., the space $L^2(\mathbb{R}^2)$ for image processing, the space l^2 for discrete-time one-dimensional signals, the space $l^2 \times l^2$, etc.

Windowing (in our continuous-time one-dimensional formulation: restricting the signal support to a finite interval $E \subset \mathbb{R}$) can be incorporated most elegantly by adding $C_{N+1} := L^2(E)$ to the list of constraint sets.

Our approach is based on the algorithm of Censor and Elfving, restricted to distance functions of the weighted norm type. Although we do not pursue this, we could actually have considered without any difficulty the slightly larger set of *measure-based norms* of the form

$$d^2(f,g) = \int_{-\infty}^{+\infty} |\hat{f}(t) - \hat{g}(t)|^2 d\mu(t),$$

in the sense of a Stieltjes integral with respect to a general nonnegative measure μ.

Besides its improved convergence behavior, the algorithm presented here provides ample tuning latitude. By changing the relative weights β_i, one may achieve some control on the expected location of the solution or express the reliability of the different constraints used. We hope that the results presented here will stir renewed interest in this class of algorithms. Indeed, promising results using this algorithm have been reported in [23, 26, 25], exploiting the multidistance (projection) latitude.

Acknowledgments. We wish to thank Prof. Yair Censor from the Department of Mathematics and Computer Science, Haifa University, and Prof. Simon Reich from the Faculty of Mathematics, Technion, for many useful and enlightening discussions.

Appendix A. We wish to demonstrate how the use of multiple distance functions (multidistance projections) can simplify the projection process. For a more complete discussion, see [22]. We use for this task an example which appeared recently in the literature, viz., the example in [12, Section V.A] entitled signal deconvolution. The task in this example is to deconvolve a signal which is blurred by a linear shift invariant blur and is modeled by

$$x = Lh + u,$$

where x is the recorded signal, u is additional noise, h is the original signal to be restored, and L is the blurring operator, e.g., $Lh = f * h$ where f is a Gaussian function and $*$ denotes convolution. In this work (conceptually), three primary sets are considered:

$$C_1 = \{a \in \mathbb{R}^N \quad | \; x[n] - \delta \leq (f * a)[n] \leq x[n] + \delta\}, \quad \forall n \in \{1,\ 2, \ldots N\},$$
$$C_2 = \{a \in \mathbb{R}^N \quad | \; \text{angle } (A[k]) = \text{ angle } (H[k])\}, \quad \forall k \in \{1,\ 2, \ldots N\},$$

where the Fourier phase of the original signal h is assumed known, and

$$C_3 = \{a \in \mathbb{R}^N \quad | \; 0 \leq a[n] \leq 12\}, \quad \forall n \in \{1,\ 2, \ldots N\},$$

where $A[k] = \mathcal{F}\{a[n]\}[k]$ and $\text{angle}(A[k]) := \frac{A[k]}{|A[k]|}$ (assuming $A[k]$ is nonzero). The projections onto C_2, C_3 with respect to the Euclidean norm-based distance function are simple (as the sets are explicit sets, using the terminology of [22]). However, the projection onto C_1 is complicated (as the set is characterized indirectly, through the outcome of a linear shift invariant operator, i.e., a convolutional function, applied to its members). Hence, C_1 is further decomposed into

$$C_1 = \bigcap_{i=1}^{N} S_i,$$

where

$$S_i = \{a \in \mathbb{R}^N \quad | \; x[i] - \delta \leq (f * a)[i] \geq x[i] + \delta\} \; .$$

The projection onto the *interval* set S_i is indeed simple. Hence, instead of performing the projections onto just three sets $C_1,\ C_2,\ C_3$, we project onto sixty-six sets, i.e., $C_2,\ C_3,\ S_1,\ S_2, \ldots S_{64}$ (in this case $N = 64$), which leads to many projections and, hence, a lengthy procedure.

We now describe how by using a different distance function we are able to perform the projection onto C_1 in a *single* iteration. Consider the following weighted norm-based distance function:

$$d_1(a_1, a_2) = \|a_1 - a_2\|_W, \quad \text{where} \quad \|a_1\|_W^2 = \sum_k |A[k]|^2 W[k]$$

and $W[k] = |F[k]|^2$, where $F[k] = \mathcal{F}\{f[n]\}[k]$. Also, we have the standard (unweighted) Euclidean norm-based distance function d_e :

$$d_e(a_1, a_2) = \|a_1 - a_2\|.$$

Then, we have that the projection onto C_1 with respect to d_1 assumes the simple form (for the full details see [22])

$$P_{C_1}^{d_1}(a) = \mathcal{F}^{-1}\left\{\frac{R'[k]}{F[k]}\right\},$$

where $R'[k] = \mathcal{F}\{\rho'[n]\},\ \rho[n] = (f * a)[n]$, and $\rho'[n] = P_{C_1^1}^{d_e}(\rho)[n]$ (the projection of ρ onto C_1^1 with respect to the usual Euclidean norm-based distance function), where

$$C_1^1 = \{\rho \in \mathbb{R}^N \quad | \; x[n] - \delta \leq \rho[n] \leq x[n] + \delta \quad \forall n \in \{1,\ 2, \ldots N\}$$

(a simple *interval* constraint on ρ).

Thus, we can obtain the projection of a onto C_1 in *one* iteration, via a standard simple projection of ρ onto C_1^1, rather than decomposing C_1 into 64 individual sets and projecting onto each set separately. However, the projection onto C_1 is simple (given by the above) only with respect to d_1, not with respect to d_e. Hence, using our parallel projection method, only three projections are employed, as opposed to the method employed in [12] which requires a uniform metric for all projections (e.g., d_e) and therefore 66 projections would have to be performed. See also a fully developed example in [26, Section 3a].

Thus, the liberty of using multiple distance functions reduces the number of projections and, hence, enhances the efficiency of the special parallel projection method.

Observing the results in Figures 5–10 of [12] (which discusses a uniform metric parallel projection method), it is clear that even the standard, uniform metric, parallel projection method outperforms the serial projection method, let alone the multidistance parallel projection method presented here.

REFERENCES

[1] J. B. Baillon, R. E. Bruck, and S. Reich, *On the asymptotic behavior of nonexpansive mappings and semigroups in Banach spaces*, Houston. J. Math., 4 (1978), pp. 1–9.

[2] H. H. Bauschke and J. M. Borwein, *On Projection Algorithms for Solving Convex Feasibility Problems*, Research report 93–12, Department of Mathematics and Statistics, Simon Fraser University, Burnaby, B.C., Canada, June, 1993.

[3] L. M. Bregman, *The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming*, U.S.S.R. Comput. Math. Comp. Phys., 7 (1967), pp. 200–217.

[4] F. E. Browder and W. Petryshyn, *The solution by iteration of non linear functional equations in Banach spaces*, Bull. Amer. Math. Soc., 72 (1966), pp. 571–575.

[5] Y. Censor, *Row action methods for huge and sparse systems and their applications*, SIAM Rev., 23 (1981), pp. 444–457.

[6] Y. Censor, P. P. B. Eggermont, and D. Gordon, *Strong underrelaxation in Kaczmarz's method for inconsistent systems*, Numer. Math., 41 (1983), pp. 83–92.

[7] Y. Censor and T. Elfving, *A multiprojection algorithm using Bregman projections in a product space*, Numer. Algorithms, 8 (1994), pp. 221–239.

[8] Y. Censor and G. T. Herman, *On some optimization techniques in image reconstruction from projections*, Appl. Numer. Math., 3 (1987), pp. 365–391.

[9] W. Cheney and A. A. Goldstein, *Proximity maps for convex sets*, Proc. Amer. Math. Soc., 10 (1959), pp. 448–450.

[10] N. Cohen and T. Kotzer, *Non expansive mappings*: *Relaxation and convergence*, EE Publication 921, Department of Electrical Engineering, Technion-I.I.T., October, 1994.

[11] P. L. Combettes, *The foundations of set theoretic estimation*, Proc. IEEE, 81 (1993), pp. 182–208.

[12] P. L. Combettes, *Inconsistent signal feasibility problems*: *Least-squares solutions in a product space*, IEEE Trans. Sig. Proc., 42 (1994), pp. 2955–2966.

[13] P. L. Combettes and H. Puh, *Parallel projection methods for set theoretic signal reconstruction and restoration*, ICASSP'93, paper WPF.10 1993, pp. V297–V300.

[14] P. L. Combettes and H. Puh, *A fast parallel projection algorithm for set theoretic image recovery*, ICASSP'94, paper 61.3 1994, pp. V473–V476.

[15] A. R. De Pierro and A. N. Iusem, *A simultaneous projections method for linear inequalities*, Linear Algebra Appl., 64 (1985), pp. 243–253.

[16] A. R. De Pierro and A. N. Iusem, *A parallel projection method of finding a common point of a family of closed convex sets*, Pesquisa Operacional, 5 (1985), pp. 1–20.

[17] I. Ekeland and R. Temam, *Convex Analysis and Variational Problems*, North–Holland, Amsterdam, 1976.

[18] M. Goldberg and R. J. Marks II, *Signal synthesis in the presence of an inconsistent set of constraints*, IEEE Trans. Circuits and Systems, 32 (1985), pp. 647–663.

[19] L. G. Gubin, B. T. Polyak, and E. V. Raik, *The method of projections for finding the common point of convex sets*, U.S.S.R. Comput. Math. and Phys., 7 (1967), pp. 1–24.

[20] G. T. Herman, *Image Reconstruction for Projections*: *The Fundamentals of Computerized Tomography*, Academic Press, New York, 1980.

[21] R. Kastner and R. Mitra, *A spectral-iteration technique for analyzing scattering from arbitrary bodies, part I*: *Cylindrical scatterers with E-wave incidence*, IEEE Trans. Antennas and Propagation, 31 (1983), pp. 499–506.

[22] T. Kotzer, N. Cohen, and J. Shamir, *Generalized approach to projections onto convex constraint sets*, Proc. IEEE, (1994), pp. 77–81.

[23] T. Kotzer, N. Cohen, and J. Shamir, *Image restoration by a novel parallel projection onto constraint sets method*, Opl. Lett., 20 (1995), pp. 1172–1174.

[24] T. Kotzer, N. Cohen, and J. Shamir, *A Projection Algorithm for Consistent and Iconsistent Constraints*, EE Publication 920, Department of Electrical Engineering, Technion-I.I.T., October, 1994.

[25] T. Kotzer, N. Cohen, J. Shamir, and Y. Censor, *Multi-distance, multi-projection parallel projection method*, The International Conference on Optical Computing, OC94, Edinburgh, 1994.

[26] T. Kotzer, J. Rosen, and J. Shamir, *Application of serial and parallel projection methods to correlation filter design*, Appl. Opt., 34 (1995), pp. 3883–3895.

[27] A. Levi and H. Stark, *Image restoration by the method of generalized projections with application to restoration from magnitude*, J. Opt. Soc. Amer. A, 1 (1984), pp. 932–943.

[28] U. Mahlab and J. Shamir, *Optical pattern recognition based on convex functions*, J. Opt. Soc. Amer. A, 8 (1991), pp. 1233–1239.

[29] S. Oh, R. J. Marks II, and L. E. Atlas, *Kernel synthesis for generalized time-frequency distributions using the method of alternating projections onto convex sets*, IEEE Trans. Sig. Proc., 42 (1994), pp. 1653–1661.

[30] Z. Opial, *Weak convergence of the sequence of successive approximations for non expansive mappings*, Bull. Amer. Math. Soc., 73 (1967), pp. 591–597.

[31] G. Pierra, *Decomposition through formalization in product space*, Math. Programming, 28 (1984), pp. 96–115.

[32] R. Piestun and J. Shamir, *Control of wave-front propagation with diffractive elements*, Opt. Lett., 19 (1994), pp. 771–773.

[33] J. Rosen, *Learning in correlators based on projections onto constraint sets*, Opt. Lett., 18 (1993), pp. 1183–1185.

[34] J. Rosen, *Synthesis of nondiffracting beams in free space*, Opt. Lett., 19 (1994), pp. 369–371.

[35] J. Rosen and J. Shamir, *Application of the projection-onto-constraint-sets algorithm for optical pattern recognition*, Opt. Lett., 16 (1991), pp. 752–754.

[36] H. J. Trussell and M. R. Civanlar, *The feasible solution in signal restoration*, IEEE Trans. Acou. Spe. and Sig. Proc., 32 (1984), pp. 201–212.

[37] D. C. Youla and H. Webb, *Image restoration by the method of convex projections*: *Part* 1 – *Theory*, IEEE Trans. Med. Im., 1 (1982), pp. 81–94.

SIAM J. Optim.
Vol. 7, No. 2, pp. 547–559, May 1997

SINGLE MACHINE SCHEDULING TO MINIMIZE BATCH DELIVERY AND JOB EARLINESS PENALTIES[*]

T. C. EDWIN CHENG[†], MIKHAIL Y. KOVALYOV[‡], AND BERTRAND M.-T. LIN[§]

Abstract. We study a problem in which a set of n jobs has to be batched as well as scheduled for processing on a single machine. A constant machine set-up time is required before the first job of each batch is processed. A schedule specifies the sequence of batches, where each batch comprises a sequence of jobs. The batch delivery time is defined as the completion time of the last job in a batch. The earliness of a job is defined as the difference between the delivery time of the batch to which it belongs and the job completion time. The objective is to find a number B of batches and a schedule so as to minimize the sum of the total weighted job earliness and mean batch delivery time. The problem is shown to be strongly NP-hard. It remains strongly NP-hard if the set-up time is zero and $B \leq U$ for any variable $U \geq 2$ or if $B \geq U$ for any constant $U \geq 2$. The problem is proved to be ordinary NP-hard even if the set-up time is zero and $B \leq 2$. For the case $B \leq U$, a dynamic programming algorithm is presented, which is pseudopolynomial for any constant $U \geq 2$. Algorithms with $O(n^2)$ running times are derived for the cases when all weights are equal or all processing times are equal. For the general problem, a family of heuristics is suggested. Computational experiments on the proposed heuristic algorithm are conducted. The results suggest that the heuristics are effective in generating near-optimal solutions quickly.

Key words. single machine scheduling, batch scheduling, NP-hardness, dynamic programming, polynomial algorithms

AMS subject classifications. 68Q25, 90C39

PII. S1052623494269540

1. Introduction.

Processing jobs in batches is a common practice in flexible manufacturing. Scheduling models which combine partitioning jobs into batches and sequencing jobs in each batch have been extensively studied lately. Most of the results in the batch scheduling area are obtained for the problem of scheduling jobs in batches on a single machine to minimize the total weighted job completion time. In this problem, there is a common set-up time between consecutively scheduled batches, and the completion time of a job is equal to the completion time of its batch, so all jobs in the same batch are completed at the same time. Albers and Brucker [1] proved that this problem is NP-hard but polynomially solvable when the job sequence is predetermined. Polynomial time algorithms have also been presented for the cases when all job weights are equal (Coffman, Yannakakis, Magazine, and Santos [8]) all processing times are equal (Albers and Brucker [1]), and both weights and processing times are equal (Nadeff and Santos [17]; Coffman, Nozari, and Yannakakis [9]; Shallcross [19]).

[*] Received by the editors June 13, 1994; accepted for publication (in revised form) November 2, 1995.
http://www.siam.org/journals/siopt/7-2/26954.html

[†] Office of the Vice President (Research & Postgraduate Studies), The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong (mscheng@polyu.edu.hk). This author was supported in part by the Hong Kong Research Grants Council under grant HKP54/94E.

[‡] Institute of Engineering Cybernetics, Belarus Academy of Sciences, Surganova 6, 220012 Minsk, Belarus (koval@newman.basnet.minsk.by). This author was supported in part by the International Association for the Promotion of Cooperation with Scientists from the Independent States of the former Soviet Union under grant 93-0257.

[§] Department of Information Management, Ming-Chuan College, Taipei 11120, Taiwan, Republic of China (mtlin@mcu.edu.tw). This author was supported in part by the Republic of China National Science Foundation under grant 84-2213-E13-5.

In this paper, we introduce a scheduling problem with batch delivery and job earliness penalties, which may be stated as follows. There are n jobs to be scheduled on a single machine. Each job j has an integer processing requirement $p_j > 0$ and a weight $w_j \geq 0$, which may be a noninteger. Jobs may be combined to form batches containing contiguously scheduled jobs. For each batch, a constant machine set-up time $s \geq 0$ is required before the first job of the batch is processed. The machine can handle only one job at a time and cannot process any jobs while a set-up is performed. All jobs in the same batch are delivered to the customer together upon the completion of the last job in the batch.

Given a number B of batches, a schedule specifies the sequence $1, \ldots, B$ of these batches, where each batch b is a sequence of jobs it contains. Given a number of batches and a schedule, the completion time C_j of each job j is easily determined. It is measured from the beginning of the scheduling horizon, i.e., from time zero. We define the batch b delivery time D_b as the completion time of the last job in the batch and the earliness E_j of job j in batch b as the difference between the delivery time of batch b and the completion time of job j: $E_j = D_b - C_j$ if $j \in b$.

The objective is to find an optimal number B of batches and an optimal schedule so as to minimize the sum of the total weighted job earliness and mean batch delivery time:

$$\sum_{j=1}^{n} w_j E_j + \sum_{b=1}^{B} D_b / B.$$

This problem is closely related to the single machine scheduling problem to minimize the total weighted job earliness plus a batch delivery penalty depending only on the number of batches: $\sum_{j=1}^{n} w_j E_j + \gamma(B)$, where $\gamma(B)$ is a certain nonnegative function. Cheng and Kahlbacher [7] first showed that the general version of this problem is ordinary NP-hard, while Cheng, Gordon, and Kovalyov [6] later proved that it is strongly NP-hard. Polynomial algorithms for special cases when all weights are equal or all processing times are equal are presented by Cheng and Gordon [5] and Cheng, Gordon, and Kovalyov [6].

Motivation of our problem comes from the very large-scale integrated circuit manufacturing, which can be divided into four main stages: wafer fabrication, wafer probe, assembly, and final testing. Scheduling problems arising at the wafer fabrication stage have been considered by Dayhoff and Atherton [10], Bitran and Tirupati [3], Chen et al. [4], Glassey and Resende [13], and Wein [20]. Scheduling models which are typical for the assembly stage have been addressed in Dobson, Karmarkar, and Rummel [11], Baker [2], and in the papers indicated at the beginning of this section. The problem of scheduling semiconductor burn-in operations at the final testing stage has been studied by Lee, Uzsoy and Martin–Vega [16]. The scheduling problem studied in this paper arises at the assembly stage. In this stage, chips of various types are attached and placed on a circuit board by a pick-and-place machine. Each circuit board represents a job; upon completion, it is loaded onto a pallet. Intermittently, pallets are moved to the soldering machine and then to the test area. A set of circuit boards loaded on a pallet corresponds to a batch. The time to move a previous pallet and to install a new one corresponds to a set-up time.

For the assembly stage, an important performance criterion is to minimize the finished product inventories which are related to the total weighted earliness $\sum w_j E_j$. For succeeding operations, safety stocks of the product which justify the consideration of the mean product flow time criterion are important. Since the product flows on pallets after assembly, the latter criterion is the mean batch delivery time $\sum D_b/B$. Our objective $\sum w_j E_j + \sum D_b/B$ is a linear combination of the above two criteria. By changing values for w_j, we can increase or decrease the impact of one of these criteria on the optimal schedule.

An analysis of our problem shows that the creation of small batches increases the total batch delivery time while the creation of large batches increases the job earliness within the batches. If only one of these strategies is applied to solve the problem, it is unlikely to find a solution with a reasonable objective value. This observation suggests that the batching decision is essential for our problem. As for the sequencing decision, we now show that we may restrict our search to schedules in which jobs in each batch are sequenced in *LWPT (longest weighted processing time)* order so that $p_{i_1}/w_{i_1} \geq p_{i_2}/w_{i_2} \geq \cdots \geq p_{i_k}/w_{i_k}$ if jobs $i_1, i_2, \ldots, i_k$ are sequenced in the batch in that order.

LEMMA 1.1. *In any optimal solution, jobs within each batch are sequenced in LWPT order.*

Proof. Consider an optimal solution and assume, without loss of generality, that jobs $i, i+1, \ldots, k$ are sequenced in a certain batch in that order. Assume that the statement of the lemma is not satisfied: $p_j/w_j < p_{j+1}/w_{j+1}$ for a certain $i \leq j \leq k-1$. It is easily checked that swapping j and $j+1$ decreases the total weighted job earliness by $w_j p_{j+1} - w_{j+1} p_j > 0$ and does not affect the batch delivery times. This contradicts the optimality of the original solution. □

Since jobs within each batch must be processed in LWPT order, the problem reduces to one of finding a number B of batches and a partition of the jobs into these batches.

The remainder of the paper is organized as follows. In the next section, we prove that the general problem is strongly NP-hard and that it remains strongly NP-hard when $s = 0$ and $B \leq U$ for a variable $U \geq 2$ or $B \geq U$ for any constant $U \geq 2$. We show that the problem is ordinary NP-hard even if $s = 0$ and $B \leq 2$. A dynamic programming algorithm is presented for the case when $B \leq U$. This algorithm runs in $O(nU^2(\sum_{j=1}^{n} p_j)^{U-1})$ time. In the following section, we derive $O(n^2)$ algorithms for the cases when all weights are equal or all processing times are equal. A heuristic approach for the general problem is then suggested. Computational results for the heuristics are also included. The paper concludes with some remarks and suggestions for further research.

2. NP-hardness proofs and dynamic programming.

It is convenient to adopt the three-field notation of Graham et al. [14] to denote our family of problems. In the notation $1/\beta/\gamma$, the first field denotes the single machine environment. The second field, $\beta \subset \{\emptyset, B \leq U, B \geq U, B = U, s = 0, p_j = p\}$, indicates the batch constraint and job characteristics. Here, $B \leq U$ and $B \geq U$ indicate that the number of batches is bounded from above or from below, respectively, by a number U; $B = U$ denotes that the number of batches is equal to U; $s = 0$ denotes a zero set-up time; $p_j = p$ denotes that all processing times are equal to p. The third field, $\gamma \in \{\sum w_j E_j + \sum D_b/B, w\sum E_j + \sum D_b/B, \sum E_j + \sum D_b/B\}$, refers to the optimality criterion. Here, $w\sum E_j$ and $\sum E_j$ arise when $w_j = w$ and $w_j = 1$, respectively, for $j = 1, \ldots, n$. Our original problem is represented by $1//(\sum w_j E_j + \sum D_b/B)$.

In this section, we prove that the general problem, $1//(\sum w_j E_j + \sum D_b/B)$, is strongly NP-hard and the problem $1/B \leq U/(\sum w_j E_j + \sum D_b/B)$ is ordinary NP-hard for any constant $U \geq 2$. The complexities of the problems $1/B \geq U/(\sum w_j E_j + \sum D_b/B)$ and $1/s = 0, B \leq U/(\sum w_j E_j + \sum D_b/B)$ are easily established using the same argument. Then, we present a dynamic programming algorithm for the problem $1/B \leq U/(\sum w_j E_j + \sum D_b/B)$. We begin with the strong NP-hardness proof.

THEOREM 2.1. *The problem* $1//(\sum w_j E_j + \sum D_b/B)$ *is strongly* NP*-hard.*

Proof. We show that the decision version of our problem is strongly NP-complete by a transformation from the strongly NP-complete problem 3-PARTITION (Garey and Johnson, [12]): given positive integers $a_1, \ldots, a_{3U}$ and A such that $A/4 < a_j < A/2$ for $j = 1, \ldots, 3U$ and $\sum_{j=1}^{3U} a_j = AU$, is there a partition of the set $X = \{1, \ldots, 3U\}$ into U disjoint sets $X_1, \ldots, X_U$ such that for $1 \leq b \leq U$, $\sum_{j \in X_b} a_j = A$?

Define $c_j = 3Ua_j$ for $j = 1, \ldots, 3U$ and $C = \sum_{j=1}^{3U} c_j/U = 3UA$. Given any instance of 3-PARTITION, we construct an instance of our problem in which the setup time

$$s = 2 \sum_{1 \leq i < j \leq 3U} c_i c_j - C^2U^2 + C^2U + 2CU + (C+1)(U+1)$$

and there are $4U$ jobs with $w_j = p_j = c_j$ for the *partition* jobs $j = 1, \ldots, 3U$ and $p_j = 1, w_j = y = (U+2)s/2$ for the *enforcer* jobs $j = 3U+1, \ldots, 4U$. We show that there exists a solution to 3-PARTITION if and only if there exists a solution to our problem with a value not exceeding y.

If X can be divided into U disjoint sets $X_1, \ldots, X_U$ such that $\sum_{j \in X_b} a_j = A$ for $b = 1, \ldots, U$, then we construct a schedule with U batches, where batch b consists of the partition jobs of the set X_b and one enforcer job scheduled last. Since $w_j = p_j$ for $j = 1, \ldots, 3U$, the order of the partition jobs in each batch does not affect the objective value F, which can be calculated as follows:

$$F = \sum_{j=1}^{4U} w_j E_j + \sum_{b=1}^{U} D_b/U,$$

where

$$\begin{aligned}\sum_{j=1}^{4U} w_j E_j &= \sum_{b=1}^{U} \left(\sum_{i<j,\ i,j \in X_b} c_i c_j + \sum_{j \in X_b} c_j \right) \\ &= \sum_{1 \leq i < j \leq 3U} c_i c_j - \sum_{1 \leq b < e \leq U} \left(\sum_{j \in X_b} c_j \right) \left(\sum_{j \in X_e} c_j \right) + \sum_{j=1}^{3U} c_j\end{aligned}$$

and

$$\sum_{b=1}^{U} D_b/U = (U+1)s/2 + \sum_{b=1}^{U} (U+1-b) \left(\sum_{j \in X_b} c_j + 1 \right) /U.$$

Since

$$(CU)^2 = \left(\sum_{j=1}^{3U} c_j \right)^2 = \sum_{b=1}^{U} \left(\sum_{j \in X_b} c_j \right)^2 + 2 \sum_{1 \leq b < e \leq U} \left(\sum_{j \in X_b} c_j \right) \left(\sum_{j \in X_e} c_j \right),$$

we have

$$F = (U+1)s/2 + \sum_{1 \leq i < j \leq 3U} c_i c_j - C^2U^2/2 \tag{1}$$

$$+\sum_{b=1}^{U}\left(\sum_{j\in X_b} c_j\right)^2/2+CU+\sum_{b=1}^{U}(U+1-b)\left(\sum_{j\in X_b} c_j+1\right)/U.$$

Setting $\sum_{j\in X_b} c_j = 3U\sum_{j\in X_b} a_j = 3UA = C$ for $b = 1,\dots,U$, we get $F = y$.

Assume that there is a solution to the problem $1//(\sum w_jE_j + \sum D_b/B)$ with a value $F \leq y$. It is apparent that there cannot be more than U batches, since then there are at least $U+1$ set-ups and we have $F > (U+2)s/2 = y$. If there are less than U batches, then at least one batch includes at least two enforcer jobs. In this case, at least one enforcer job is not scheduled last in one of the batches. Since the weight of each enforcer job is equal to y, we again get $F > y$. Thus, there are exactly U batches and each batch includes exactly one enforcer job which is scheduled last. Denote the set of the partition jobs in batch b by X_b. Then the value F of our solution can be calculated as shown in (1). By simplifying $F \leq y$, we obtain

$$\sum_{b=1}^{U}\left(\sum_{j\in X_b} c_j\right)^2/2+\sum_{b=1}^{U}(U+1-b)\sum_{j\in X_b} c_j/U \leq C^2U/2+C(U+1)/2.$$

The latter inequality can be represented as follows:

$$\sum_{b=1}^{U}\left(\sum_{j\in X_b} c_j - C\right)\left(\sum_{j\in X_b} c_j + C\right)+2(U+1-b)\left(\sum_{j\in X_b} c_j - C\right)/U \leq 0.$$

Define $\delta_b = \sum_{j\in X_b} c_j - C$ for $b = 1,\dots,U$. Clearly, $\sum_{b=1}^{U}\delta_b = 0$. We have $\sum_{b=1}^{U}(\delta_b^2 + 2(U+1-b)\delta_b/U) \leq 0$ or, equivalently,

$$\sum_{b=1}^{U}(\delta_b+(U+1-b)/U)^2 \leq \sum_{b=1}^{U}(U+1-b)^2/U^2 \leq U.$$

Thus, $\max_{1\leq b\leq U}|\delta_b| \leq U^{1/2}+1 \leq 2U$. The latter relations provide

$$C-2U \leq \sum_{j\in X_b} c_j \leq C+2U \text{ for } b = 1,\dots,U.$$

Substituting $3Ua_j$ for c_j and $3UA$ for C, we deduce that

$$A-2/3 \leq \sum_{j\in X_b} a_j \leq A+2/3 \text{ for } b = 1,\dots,U.$$

These inequalities and the integrality of a_j yield $\sum_{j\in X_b} a_j = A$ for $b = 1,\dots,U$, as required. □

Similar reductions show that the problem $1/B \geq U/(\sum w_jE_j + \sum D_b/B)$ is strongly NP-hard if U is a constant and the problems $1/B \leq U/(\sum w_jE_j + \sum D_b/B)$ and $1/s = 0, B \leq U/(\sum w_jE_j + \sum D_b/B)$ are strongly NP-hard if U is a given variable. For the former problem, the only modification of the above proof is that the (variable) number U of sets in 3-Partition is substituted by B, since B is a variable number of batches now and U is a constant. For the second problem, the proof is completely the same. For the third problem with zero set-up time, we should set

$$y = \sum_{1\leq i<j\leq 3U} c_ic_j - C^2U^2/2 + C^2U/2 + CU + (C+1)(U+1)/2$$

in order to show that $B \geq U$. In Theorem 2.1, a nonzero set-up time has been used to show only that $B \leq U$. Therefore, if $B \leq U$ is given a priori, we can set $s = 0$ in our proof.

THEOREM 2.2. *The problem* $1/B \leq U/(\sum w_j E_j + \sum D_b/B)$ *is ordinary* NP*-hard for any constant* $U \geq 2$.

Proof. Our proof is similar to the one in the previous theorem. A transformation from the NP-complete problem PARTITION (Garey and Johnson [12]) is used. □

Besides the above results, we have also proved that the problem with a zero set-up time, $1/s = 0, B \leq U/(\sum w_j E_j + \sum D_b/B)$, is ordinary NP-hard for any constant $U \geq 2$.

It should be noted that all of the above complexity results remain valid if the total set-up time is included in the objective function instead of the mean batch delivery time.

We now present a dynamic programming algorithm DP for the problem $1/B \leq U/(\sum w_j E_j + \sum D_b/B)$. This algorithm is based on Lemma 1.1. Assume that jobs are numbered in *SWPT (shortest weighted processing time)* order so that $p_1/w_1 \leq \cdots \leq p_n/w_n$. In Algorithm DP, jobs are considered in natural order $1, \ldots, n$. Job j is either assigned to the beginning of one of the current batches or it starts a new batch. Thus, jobs within each batch are sequenced in LWPT order. We recursively compute the value of $F_j(P_1, \ldots, P_B)$, which represents the minimal objective value subject to j jobs being scheduled in B batches, and the total processing time of the jobs in batch b is equal to P_b for $b = 1, \ldots, B$. Note that the set-up time is not included in P_b.

Set $T_j = \sum_{i=1}^{j} p_i$ for $j = 1, \ldots, n$. A formal description of Algorithm DP is as follows.

ALGORITHM DP.

Step 1 (Initialization) Number jobs in SWPT order so that $p_1/w_1 \leq \cdots \leq p_n/w_n$. Set $F_j(P_1, \ldots, P_B) = \infty$ for $j = 0, 1, \ldots, n, 0 \leq P_b \leq T_n, b = 1, \ldots, B$ and $B = 1, \ldots, U$. Set $F_0(0) = 0$. Set $j = 1$.

Step 2 (Recursion) Compute the following for all tuples $(P_1, \ldots, P_B)$ such that $p_j \leq P_b \leq T_j, b = 1, \ldots, B, B = 1, \ldots, \min\{j, U\}$.

$$F_j(P_1, \ldots, P_B) = \min_{1 \leq b \leq B} \min \tag{2}$$

$$\begin{cases} F_{j-1}(P_1, \ldots, P_{b-1}, P_b - p_j, P_{b+1}, \ldots, P_B) & \\ +w_j(P_b - p_j) + p_j(B - b + 1)/B & \text{if } P_b > p_j, \\ F_{j-1}(P_1, \ldots, P_{b-1}, P_{b+1}, \ldots, P_B) + s/2 + p_j(B - b + 1)/B & \\ +(\sum_{k=1, k \neq b}^{B} (k-1)P_k - \sum_{k=b+1}^{B} (B - k + 1)P_k)/(B^2 - B) & \text{if } P_b = p_j, \\ \infty & \text{if } P_b < p_j. \end{cases}$$

The three quantities in the right-hand side of equation (2) represent the three possible scheduling choices for job j with respect to batch b:

1. Add job j to the beginning of the existing batch b.
2. Form a new batch b consisting of the sole job j.
3. Do not assign job j to batch b.

If $j = n$, go to Step 3; otherwise set $j = j + 1$ and repeat Step 2.

Step 3 (Optimal solution) Define optimal solution value

$$F^* = \min\{F_n(P_1, \ldots, P_B) | 0 \leq P_b \leq T_n, B = 1, \ldots, U\}$$

and use backtracking to find the corresponding optimal solution.

THEOREM 2.3. *Algorithm DP solves the problem* $1/B \leq U/(\sum w_j E_j + \sum D_b/B)$ *in* $O(nU^2(\sum_{j=1}^n p_j)^{U-1})$ *time.*

Proof. Due to Lemma 1.1, there is always an optimal schedule with jobs arranged in LWPT order within each batch. Therefore, at each stage of the algorithm we need only decide whether to include job j in batch b and, if so, whether batch b is new or not. It is now easy to apply the general dynamic programming justification for scheduling problems (Rothkopf, [18]; Lawler and Moore, [15]) to show that DP solves the problem $1/B \leq U/(\sum w_j E_j + \sum D_b/B)$. The time complexity of this algorithm can be established as follows.

In each iteration of Step 2, only $B-1$ of the values $P_1, \dots, P_B$ are independent, since $P_1 + \cdots + P_B = T_j$. Hence, in iteration j of Step 2, the number of different tuples $(P_1, \dots, P_B)$ for $B = 1, \dots, U$ is at most UT_j^{U-1}. For each tuple $(P_1, \dots, P_B)$, the right-hand side of equation (2) can be calculated in $O(B)$ time. Thus, Step 2 requires $O(nU^2T_n^{U-1})$ time, which is the overall time complexity of Algorithm DP as well. ▯

Theorem 3 shows that the problem $1/B \leq U/(\sum w_j E_j + \sum D_b/B)$ is not strongly NP-hard for any constant $U \geq 2$.

3. Polynomially solvable cases. In this section, we present polynomial time algorithms for two special cases of our problem; namely, all weights are equal and all processing times are equal. We first show that the problem with equal weights, $1//(w \sum E_j + \sum D_b/B)$, can be solved in $O(n^2)$ time.

Consider a certain solution to the problem $1//(w \sum E_j + \sum D_b/B)$. To facilitate discussion, we represent it as a pair (U, x), where U is the number of batches, x is a sequence of the batches $1, 2, \dots, U$, and each batch b includes jobs $i_1^b, i_2^b, \dots, i_{j(b)}^b$ in that order. The total earliness of the jobs in batch b can be calculated as follows:

$$\sum_{k \in b} E_k = (j(b)-1)p_{i_{j(b)}^b} + (j(b)-2)p_{i_{j(b)-1}^b} + \cdots + p_{i_2^b} = \sum_{k=1}^{j(b)} (k-1)p_{i_k^b}.$$

For all n jobs, we have $\sum_{j=1}^n E_j = \sum_{b=1}^U \sum_{k=1}^{j(b)} (k-1)p_{i_k^b}$.

Recall the definition of the total processing time of jobs in batch b: $P_b = \sum_{k=1}^{j(b)} p_{i_k^b}$. For the mean batch delivery time, we have

$$\sum_{b=1}^U D_b/U = s(U+1)/2 + \sum_{b=1}^U (U+1-b)P_b/U = s(U+1)/2 + \sum_{b=1}^U \sum_{k=1}^{j(b)} p_{i_k^b}(U+1-b)/U.$$

Thus, the problem $1//(w \sum E_j + \sum D_b/B)$ reduces to one of minimizing $s(U+1)/2 + F(U, x)$, where

$$F(U,x) = \sum_{b=1}^U \sum_{k=1}^{j(b)} ((U+1-b)/U + w(k-1))p_{i_k^b}.$$

Let (U^*, x^*) be an optimal solution to this problem and let $x^{(U)}$ be an optimal solution to the problem of minimizing $F(U, x)$. We have

$$s(U^*+1)/2 + F(U^*, x^*) = \min\{s(U+1)/2 + F(U, x^{(U)}) | U = 1, \dots, n\}.$$

TABLE 1
(U).

$b\backslash k$	1	2	3	...	n
1	1	$1+w$	$1+2w$	...	$1+(n-1)w$
2	$(U-1)/U$	$(U-1)/U+w$	$(U-1)/U+2w$	...	$(U-1)/U+(n-1)w$
.	.	.	.	...	.
.	.	.	.	...	.
.	.	.	.	...	.
$U-2$	$3/U$	$3/U+w$	$3/U+2w$	...	$3/U+(n-1)w$
$U-1$	$2/U$	$2/U+w$	$2/U+2w$	...	$2/U+(n-1)w$
U	$1/U$	$1/U+w$	$1/U+2w$	...	$1/U+(n-1)w$

Consider the problem of minimizing $F(U,x)$. In this problem, $F(U,x)$ is a weighted sum of n number of p_j values where the weights are presented in Table $1(U)$.

In $F(U,x)$, each element in Table $1(U)$ may be used at most once in order to satisfy the restriction that each job should be assigned to exactly one batch, and at least one element from each row of this table should be used in order to satisfy the restriction that there are exactly U batches. To find an optimal solution $x^{(U)}$, it is obvious that we have to choose the smallest elements satisfying the above conditions, i.e., all U elements from the first column and the $n-U$ smallest elements from the remaining part of the table, and then match the smallest chosen elements with the largest processing requirements p_j. The procedure of choosing the r smallest elements $t^U_{bk}=(U+1-b)/U+w(k-1)$ can be implemented in $O(r)$ time. If p_j is matched with an element t^U_{bk}, then job j is sequenced kth in batch b. Thus, $x^{(U)}$ can be found in $O(n)$ time and (U^*,x^*) can be found in $O(n^2)$ time. Therefore, we have the following theorem.

THEOREM 3.1. *The problem* $1//(w\sum E_j+\sum D_b/B)$ *is solved in* $O(n^2)$ *time.*

We now study the problem with equal processing times, $1/p_j=p/(\sum w_jE_j+\sum D_b/B)$. For this problem, we first rearrange the jobs such that $w_1\geq w_2\geq\cdots\geq w_n$. Assume that there are exactly U batches, $B_1,B_2,\ldots,$ and $B_U, 1\leq U\leq n$. Because the jobs have the same processing time, there is an optimal solution in which $|B_i|\leq|B_j|$ if batch B_i precedes batch B_j. With this observation, we devise the following algorithm for a fixed U:

Step 1: Assign jobs $1,2,\ldots,$ and U to batch $B_U,B_{U-1},\ldots,$ and B_1 as a partial schedule.

Step 2: Loop for job j over $U+1,U+2,\ldots,$ and n: For each partial schedule, find the last batch B_r satisfying $|B_r|<|B_U|$, and then assign job j in accordance with the following cases.

Case 1. There is no such a batch, i.e., all batches have the same number of jobs: Assign job j as the first job of batch B_U.

Case 2. $|B_r|=|B_U|-1$:

- Enhance the partial schedule by assigning job j as the first job of batch B_U. If $j=n$, output the schedule as a candidate solution.
- Enhance the partial schedule by assigning job j as the first job of batch B_r. If $j=n$, output the schedule as a candidate solution.

Case 3. $|B_r|=|B_U|-2$: Assign jobs $j,j+1,\ldots,n$ to batch B_U in the order of non-decreasing weights. Output this schedule as a candidate solution.

Step 3: Amongst the candidate solutions, output one of those with the minimum cost.

To establish the correctness of the proposed algorithm, we first consider an important property. Let s_i denote the number of successors of job i in the batch containing job i. By a simple interchange argument, we readily see that there is an optimal solution where, for any two jobs i and j, if $w_i \geq w_j$, then $s_i \leq s_j$. Assume that schedule S is an optimal solution satisfying this property. We show that S can be transformed into a candidate solution delivered by the algorithm without increasing the costs. Suppose that the partial schedule for jobs 1, 2, $\dots, j-1$ in S is the same as some partial schedule proposed by the algorithm. Now, consider the assignment of job j. In Case 1, it is evident that job j can be assigned to batch B_U to minimize the delivery penalty. In analyzing Case 2, we know, by the property just stated, that job j should be in some batch B_p with either $|B_p| = |B_r|$ or $|B_p| = |B_U|$. Therefore, if job j is not in one of the two specified positions, i.e., one in B_r and the other in B_U, we can swap the job positions without increasing the costs. As for Case 3, we note that the first job in batch B_U must be job $j-1$. Suppose that job k, $j \leq k \leq n$, is assigned to batch B_p, $p \neq U$. We can assume, without loss of generality, that $p = U - 1$. By swapping the positions of jobs $j-1$ and k, the cost will not increase. Furthermore, the derived solution has a partial schedule for jobs 1, 2, $\dots$, and $j-1$ that is the same as a partial schedule proposed by the algorithm. Continuing the above interchange arguments, we finally obtain a schedule that is the same as a candidate solution proposed by the algorithm.

Now, we turn to the issue of the time complexity of the algorithm. Because the branching from a partial schedule terminates when the condition in Case 2 is satisfied, the total number of candidate solutions is bounded by $O(n)$. By performing a simple preprocessing step to compute the cumulative job weights, the objective values of all candidate solutions can be calculated in $O(n)$ time. Noting that there are n possible values for the variable U, we conclude with the following theorem.

THEOREM 3.2. *The problem $1/p_j = p/(\sum w_j E_j + \sum D_b/B)$ is solved in $O(n^2)$ time.*

Theorems 3.1 and 3.2 resolve the computational complexities of all special cases of our problem in which either all weights or all processing times are equal.

4. Heuristics.

In this section, we present a heuristic approach to solving the general problem $1//(\sum w_j E_j + \sum D_b/B)$.

We first describe a list-scheduling algorithm for the problem with a fixed number of batches, $1/B = U/(\sum w_j E_j + \sum D_b/B)$.

Let $LIST$ be a sequence of jobs and let $RULE$ be a rule of assigning a job from $LIST$ to a batch. In a list-scheduling algorithm, jobs are considered in an order determined by $LIST$. Each successive job is scheduled according to $RULE$. We consider $LIST \in \{LWPT, SWPT, SPT, LPT, SW, LW\}$, where the jobs are numbered so that

$$\begin{aligned}
p_1/w_1 \geq p_2/w_2 \geq \cdots \geq p_n/w_n &\text{ in } LWPT,\\
p_1/w_1 \leq p_2/w_2 \leq \cdots \leq p_n/w_n &\text{ in } SWPT,\\
p_1 \geq p_2 \geq \cdots \geq p_n &\text{ in } LPT,\\
p_1 \leq p_2 \leq \cdots \leq p_n &\text{ in } SPT,\\
w_1 \geq w_2 \geq \cdots \geq w_n &\text{ in } LW,\\
w_1 \leq w_2 \leq \cdots \leq w_n &\text{ in } SW.
\end{aligned}$$

We use two types of $RULE$: $RULE1$ and $RULE2$. According to $RULE1$, a job is assigned to the end of the earliest batch b with the minimal total processing time

TABLE 2
Computational results for $s = 500$ and $n = 100$.

RULE, *LIST*	$p' = 100$	$p' = 100$	$p' = 10$	$p' = 10$
	$w' = 10$	$w' = 1$	$w' = 10$	$w' = 1$
*RULE*1, *LWPT*	30399.40	22561.75	17931.31*	7210.09
*RULE*1, *SWPT*	30194.13	23130.77	18563.15	7536.56
*RULE*1, *LPT*	30436.84	22486.16*	18000.67	7251.58
*RULE*1, *SPT*	30157.10*	23021.43	18502.26	7496.78
*RULE*1, *LW*	30281.10	23105.58	18487.39	7495.54
*RULE*1, *SW*	30311.46	22664.68	18011.39	7251.39
*RULE*2, *LWPT*	30399.40	22648.63	17934.77	7204.55*
*RULE*2, *SWPT*	30194.13	23116.94	18557.74	7530.62
*RULE*2, *LPT*	30436.84	22641.48	18009.04	7252.37
*RULE*2, *SPT*	30157.10*	23021.43	18502.26	7489.33
*RULE*2, *LW*	30281.10	23089.33	18480.68	7487.63
*RULE*2, *SW*	30311.46	22669.87	18011.33	7258.79
EQUAL_W_AVG	30157.10	22423.06	18009.20	7245.77
EQUAL_W_MIN	30157.10	21824.43	17611.56	6958.70

$P_b = \sum_{j \in b} p_j$. According to $RULE2$, a job is assigned to the end of the earliest batch b with the minimal total weighted earliness $F_b = \sum_{j \in b} w_j E_j$. Values P_b or F_b, $b = 1, \ldots, U$, are stored in a heap. The heap can be initiated in $O(U \log U)$ time and updated in $O(\log U)$ time. Therefore, our list-scheduling algorithm can be implemented in $O(U \log U)$ time.

We apply the list-scheduling algorithm for all possible combinations of $LIST$ and $RULE$. Let $LIST(U)$ be an algorithm which performs all twelve combinations and chooses the best constructed schedule $S^{(U)}$ with the value $F(S^{(U)})$ with respect to the problem $1/B = U/(\sum w_j E_j + \sum D_b/B)$. Our final algorithm H is to apply $LIST(U)$ for $U = 1, \ldots, n$, and select the best schedule S^H with the value

$$F(S^H) = \min\{F(S^{(U)}) | U = 1, \ldots, n\}.$$

The complexity of the algorithm H is $O(n^2 \log n)$.

In the following, we conduct computational experiments to test the proposed heuristic algorithm. Because of the intractability of the general problem, it is hard to derive exact solutions. Therefore, we make use of the polynomial algorithm designed for the equal-weight case in the previous section.

In the experiments, four parameters (namely, set-up time (s), number of jobs (n), job length (p'), and job weight (w')), are taken into consideration, and two possible values for each parameter will be set. There are a total of 16 combinations from

$$\{s = 50 \text{ or } 500\} \times \{n = 20 \text{ or } 100\} \times \{p' = 10 \text{ or } 100\} \times \{w' = 1 \text{ or } 10\}.$$

Note the actual implication of p' and w'. For a given p' (w'), all processing times (weights), p_i (w_i), are randomly drawn from the uniform distribution $[p' - 0.1p', p' + 0.1p']$ ($[w' - 0.1w', w' + 0.1w']$). For example, $p' = 100$ means that all the job lengths, p_i, are randomly drawn from the uniform distribution $[100 - 10, 100 + 10]$. This indicates a ten percent variation in processing times. Such an assumption is reasonable in real-world applications because the processing times of jobs on a production line often exhibit some degree of variation.

The platform of our experiments is a personal computer that contains an Intel Pentium 75 processor and runs MS-DOS 6.2. All the programs are coded in Turbo Pascal 6.0. Tables 2–5 display the numerical results. For each parameter combination, 12 objective values are listed for all possible $RULE \times LIST$ pairs. Besides, we

Table 3
Numerical results for $s = 500$ and $n = 20$.

$RULE, LIST$	$p' = 100$	$p' = 100$	$p' = 10$	$p' = 10$
	$w' = 10$	$w' = 1$	$w' = 10$	$w' = 1$
$RULE1, LWPT$	6304.15	4807.89*	3824.19*	1671.08
$RULE1, SWPT$	6259.10	4922.49	3920.09	1715.72
$RULE1, LPT$	6308.85	4821.18	3853.32	1679.34
$RULE1, SPT$	6254.40*	4907.74	3871.92	1710.37
$RULE1, LW$	6286.15	4916.87	3906.86	1704.23
$RULE1, SW$	6278.30	4818.16	3836.68	1681.63
$RULE2, LWPT$	6304.15	4813.10	3826.13	1668.98*
$RULE2, SWPT$	6259.10	4918.23	3920.24	1715.55
$RULE2, LPT$	6308.85	4834.14	3857.70	1679.16
$RULE2, SPT$	6254.40*	4907.74	3871.92	1700.69
$RULE2, LW$	6286.15	4910.27	3907.97	1709.12
$RULE2, SW$	6278.30	4819.40	3836.59	1682.32
$EQUAL\_W\_AVG$	6254.40	4782.91	3837.49	1672.71
$EQUAL\_W\_MIN$	6254.40	4659.33	3749.47	1634.29

Table 4
Numerical results for $s = 50$ and $n = 100$.

$RULE, LIST$	$p' = 100$	$p' = 100$	$p' = 10$	$p' = 10$
	$w' = 10$	$w' = 1$	$w' = 10$	$w' = 1$
$RULE1, LWPT$	7706.68	7609.11	3040.94	2244.61
$RULE1, SWPT$	7462.55	7374.40	3023.33	2310.07
$RULE1, LPT$	7742.37	7646.22	3046.20	2240.97*
$RULE1, SPT$	7426.82*	7337.13*	3018.05*	2298.52
$RULE1, LW$	7574.22	7484.05	3033.05	2302.99
$RULE1, SW$	7587.10	7500.85	3031.20	2251.14
$RULE2, LWPT$	7706.68	7609.11	3040.94	2250.04
$RULE2, SWPT$	7462.55	7374.40	3023.33	2307.61
$RULE2, LPT$	7742.37	7646.22	3046.20	2253.24
$RULE2, SPT$	7426.82*	7337.13*	3018.05*	2298.52
$RULE2, LW$	7574.22	7484.05	3033.05	2302.79
$RULE2, SW$	7587.10	7500.85	3031.20	2253.51
$EQUAL\_W\_AVG$	7426.82	7337.13	3018.05	2230.09
$EQUAL\_W\_MIN$	7426.82	7337.13	3018.05	2169.22

Table 5
Numerical results for $s = 50$ and $n = 20$.

$RULE, LIST$	$p' = 100$	$p' = 100$	$p' = 10$	$p' = 10$
	$w' = 10$	$w' = 1$	$w' = 10$	$w' = 1$
$RULE1, LWPT$	1610.45	1569.60	630.01	473.78*
$RULE1, SWPT$	1571.05	1544.70	626.09	487.67
$RULE1, LPT$	1619.95	1583.55	630.77	477.60
$RULE1, SPT$	1561.55*	1530.75*	625.34*	486.05
$RULE1, LW$	1594.45	1562.35	628.27	486.21
$RULE1, SW$	1589.95	1550.20	627.94	478.25
$RULE2, LWPT$	1610.45	1569.60	630.01	476.33
$RULE2, SWPT$	1571.05	1544.70	626.09	487.09
$RULE2, LPT$	1619.95	1583.55	630.77	475.46
$RULE2, SPT$	1561.55*	1530.75*	625.34*	486.05
$RULE2, LW$	1594.45	1562.35	628.27	487.23
$RULE2, SW$	1589.95	1550.20	627.94	478.58
$EQUAL\_W\_AVG$	1561.55	1530.75	625.34	472.97
$EQUAL\_W\_MIN$	1561.55	1530.75	625.34	463.17

have two values that are categorized as $EQUAL\_W\_AVG$ and $EQUAL\_W\_MIN$. The $EQUAL\_W\_AVG$ and $EQUAL\_W\_MIN$ values are obtained by applying the $O(n^2)$ algorithm for $w = \sum_{i=1}^{n} w_i$ and $w = \min_{i=1}^{n}\{w_i\}$, respectively. For each column of the table, the entries with an asterisk denote the objective value output by the heuristic algorithm, H.

It is not hard to see that the $EQUAL\_W\_MIN$ values serve as lower bounds for H values. An analysis of Tables 2–5 shows that the results delivered by algorithm H are quite close to that delivered by the polynomial algorithm. The largest relative percentage deviation between H and $EQUAL\_W\_MIN$ occurs when $n = 100, s = 500, p' = 10,$ and $w' = 10$, and the value is around 3.61 percent, which is much smaller than the assumed 10 percent deviation among the data instances. In other words, the effectiveness of algorithm H in producing near-optimal solutions is convincingly evident. Detailed observations further show some interesting properties:

1. When the set-up time, s, is relatively small, the impact of the weight, w_i, is alleviated. In Tables 4 and 5, the cases in which $EQUAL\_W\_AVG = EQUAL\_W\_MIN$ indicate that all batches contain exactly one job and that the effect of the weights is null.

2. For most of the data instances (or columns), the minimal objective values for H occur when using $RULE1$.

3. For a specific list-scheduling policy, the difference between the objective values for $RULE1$ and $RULE2$ is small.

Finally, the above numerical results show no preference to any specific list-scheduling policy. Therefore, we do not expect to obtain satisfactory solutions by simply applying a specific combination of $RULE \times LIST$. Another supporting argument for employing algorithm H is that the running sessions take less than three seconds.

5. Conclusions. The problems $1//(\sum w_j E_j + \sum D_b/B)$, $1/B \geq U/(\sum w_j E_j + \sum D_b/B)$, $1/B \leq U/(\sum w_j E_j + \sum D_b/B)$, and $1/s = 0, B \leq U/(\sum w_j E_j + \sum D_b/B)$ have been shown to be strongly NP-hard. The problems $1/B \leq 2/(\sum w_j E_j + \sum D_b/B)$ and $1/s = 0, B \leq 2/(\sum w_j E_j + \sum D_b/B)$ have been proved to be ordinary NP-hard. Algorithms with $O(n^2)$ running times have been derived for the cases when all weights are equal or all processing times are equal. Thus, the computational complexities of all special cases of the problem in which all weights or all processing times are equal have been resolved. A dynamic programming algorithm has been presented for the case with a limited number of batches. A heuristic approach has been suggested for the general problem. The numerical results reveal the practical significance of this algorithm in producing near-optimal solutions quickly.

An interesting problem for further research is one for which there is a natural restriction that each batch can include no more than a given number of jobs. The complexity aspects of this problem are yet to be studied. However, our dynamic programming algorithm DP and heuristic algorithm H can easily be modified to solve this problem. These algorithms can also be adopted for the problem in which, besides the job weights, the batch weights are given and the total weighted batch delivery time is included in the objective function.

Acknowledgments. The authors wish to thank the referees for their constructive comments and helpful discussions of the presented model.

REFERENCES

[1] S. Albers and P. Brucker (1993), *The complexity of one-machine batching problems*, Discrete Appl. Math., 47, pp. 87–107.

[2] K. R. Baker (1988), *Scheduling the production of components at a common facility*, IIE Trans., 20, pp. 32–35.

[3] G. R. Bitran and D. Tirupati (1988), *Planning and scheduling for epitaxial wafer production*, Oper. Res., 36, pp. 34–49.

[4] H. Chen, J.M. Harrison, A. Mandelbaum, A. Van Ackere, and L. M. Wein (1988), *Empirical evaluation of a queueing network model for semiconductor wafer fabrication*, Oper. Res., 36, pp. 202–215.

[5] T. C. E. Cheng and V. S. Gordon (1994), *Batch delivery scheduling on a single machine*, J. Oper. Res. Soc., 45, pp. 1211–1215.

[6] T. C. E. Cheng, V. S. Gordon, and M. Y. Kovalyov (1996), *Single machine scheduling with batch deliveries*, European J. Oper. Res., 94, pp. 277–283.

[7] T. C. E. Cheng and H. G. Kahlbacher (1993), *Scheduling with delivery and earliness penalties*, Asia-Pacific J. Oper. Res., 10, pp. 145–152.

[8] E. G. Coffman, Jr., M. Yannakakis, M. J. Magazine, and C. Santos (1990), *Batch sizing and job sequencing on a single machine*, Ann. Oper. Res., 26, pp. 135–147.

[9] E. G. Coffman, A. Nozari, and M. Yannakakis (1989), *Optimal scheduling of products with two subassemblies on a single machine*, Oper. Res., 37, pp. 426–436.

[10] J. E. Dayhoff and R. W. Atherton (1987), *A model for wafer fabrication dynamics in integrated circuit manufacturing*, IEEE Trans. Systems Man Cybernet., 17, pp. 91–100.

[11] G. Dobson, U. S. Karmarkar, and J. L. Rummel (1987), *Batching to minimize flow times on one machine*, Management Sci., 33, pp. 784–799.

[12] M. R. Garey and D. S. Johnson (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, CA.

[13] C. R. Glassey and M. G. C. Resende (1988), *Closed-loop job release control for VLSI circuit manufacturing*, IEEE Trans. Semiconductor Manufacturing, 1, pp. 36–46.

[14] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan (1979), *Optimization and approximation in deterministic sequencing and scheduling*, Ann. Discrete Math., 5, pp. 287–326.

[15] E. L. Lawler and J. M. Moore (1969), *A functional equation and its application to resource allocation and sequencing problems*, Management Sci., 16, pp. 77–84.

[16] C.-Y. Lee, R. Uzsoy, and L. A. Martin–Vega (1992), *Efficient algorithms for scheduling semiconductor burn-in operations*, Oper. Res., 40, pp. 764–775.

[17] D. Nadeff and C. Santos (1988), *One-pass batching algorithms for the one machine problem*, Discrete Appl. Math., 21, pp. 133–145.

[18] M. H. Rothkopf (1966), *Scheduling independent tasks on parallel processors*, Management Sci., 12, pp. 437–447.

[19] D. Shallcross (1992), *A polynomial algorithm for a one machine batching problem*, Oper. Res. Lett., 11, pp. 213–218.

[20] L. M. Wein (1988), *Scheduling semiconductor wafer fabrication*, IEEE Trans. Semiconductor Manufacturing, 1, pp. 115–129.

SIAM J. OPTIM.
Vol. 7, No. 2, pp. 560–578, May 1997

A NETWORK DESIGN PROBLEM FOR A DISTRIBUTION SYSTEM WITH UNCERTAIN DEMANDS[*]

FRANCO BLANCHINI[†], FRANCA RINALDI[†], AND WALTER UKOVICH[‡]

Abstract. A class of production–distribution planning problems with nonstochastic uncertain demands is modeled as a dynamic game between two players who control flows on a network with node and arc capacity constraints. Simple conditions are derived for determining which player wins the game. These conditions are then used to design a minimum cost network with the property that its feasible control strategies are allowed to meet the demand without violating the capacity constraints.

Key words. dynamic networks, dynamic games, network design

AMS subject classifications. 90B05, 90B06, 90B10, 90B15, 90B30, 90C60, 90D43, 90D50

PII. S1052623494266262

1. Introduction.

Many important problems concerning production, transportation, and distribution of goods can be addressed by network models in which nodes represent storage capabilities and arcs represent production units or transportation links. Basically, such problems consist in determining a strategy to decide arc flows in order to ship the commodity from some nodes to other nodes of the network in order to satisfy a certain demand. The literature on this subject is very extensive and we refer the reader to several textbooks (among the most recent ones, see, for instance, [1], [6], [12], [19], [20], and [28]).

In particular, dynamic network problems have received great attention. In this case, flow values, storage levels, and demands are time-varying quantities. A typical problem concerning this kind of model consists of planning the commodity flow and storage at each time in order to minimize transportation and stocking costs. For an extensive survey of these topics, see [2]. If the demand is known in the assigned time horizon, the dynamic flow problem can be handled via the well-known time-expanded network method (see, again, [2] and [31]). Unfortunately, the demand is often unknown and this fact has led to the use of stochastic methods (see, for instance, [5], [30]) to handle problems of this kind. However, the stochastic approach to the control of dynamic networks requires stochastic information which can be unavailable in some cases.

In this paper, uncertainties are modeled in a different way. Production and demand are assumed to have a known range of allowed values, but no knowledge is given on which allowed values will actually be taken. These unknown-but-bounded specifications for uncertainties are quite realistic in several situations. In general, upper and lower bounds for production and demand can be inferred from historical data or decision makers' experience much more easily and with much more confidence than empirical probability distributions for the same quantities. Sometimes, they are a consequence of a particular operational condition or a technological characteristic

[*] Received by the editors April 18, 1994; accepted for publication (in revised form) November 1, 1995.

http://www.siam.org/journals/siopt/7-2/26626.html

[†] Dipartimento di Matematica ed Informatica, University of Udine, Italy (blanchini@uniud.it, rinaldi@dimi.uniud.it). These authors were supported by C.N.R. under research grant 94.00543.CT11.

[‡] Dipartimento di Elettrotecnica, Elettronica ed Informatica, University of Trieste, Italy (ukovich@univ.trieste.it). This author was supported by C.N.R. under contract 94.01472.PF74.

of production units. In other cases, these bounds are explicitly stipulated in supply contracts. On the basis of this information, the problem is to find a flow assignment strategy capable of meeting any allowed demand without incurring capacity and storage constraint violations.

This problem can be formulated as a dynamic game between two players controlling flows on different arcs of a network. The first player represents the manager of the system, who has the responsibility of complying with the supply, demand, and system capacities. He is referred to as the controller. The second player represents the demand and is referred to as such. Each of them has to decide, at each time instant, the flow values on each of the arcs he controls (two parallel arcs are allowed between each pair of nodes, each controlled by a different player). The goal of the first player is to keep the stored amount of the commodity in the admissible range, assigning time by time an admissible flow to each of his arcs, while his opponent has the malefic role of pushing the system to a constraint violation. The first player starts the game. This implies that, at each time, he has to decide his move without knowledge of the actual choice of his opponent.

For this situation, two problems will be considered. The first is that of giving a yes or no answer to the following question: does there exist a winning strategy for the first player with assigned arc and storage capacity constraints? The solution of this problem can be given following the approach proposed in [4], [7], [9], and [13]. However, due to the particular system structure, the solution can be strongly simplified in this case. Moreover, it will be shown that a winning strategy requires solving an admissible flow problem on-line.

The second problem is a network design problem. Assume that the capacities of the demand arcs are given. Then the problem is that of determining a minimum cost network, that is, storage bounds and capacities for the controlled arcs, under the condition that a winning strategy for the first player does exist. It will be shown that this problem can be split into two independent subproblems, one consisting of the minimization of the storage capacity cost, the other of the minimization of the transportation capacity cost for the controlled arcs. While the former problem is easy to solve, the latter one turns out to be NP-hard. However, it will be shown that, although it can be formulated as a linear programming problem involving a number of constraints which is exponential in the size of the network, this number can be dramatically reduced a priori if the controlled network is weakly connected; i.e., the difference between the number of arcs and the number of nodes of the graph is low.

The structure of the paper is as follows. In section 2, the two problems of interest are formulated and discussed. The first of them is solved in section 3, and the second is solved in section 5. The integer version of the latter problem is studied in section 6, and an approximate solution method for it is proposed in section 7. Section 4 contains some complexity results about the considered problems and section 8 presents an illustrative example. Some concluding remarks are pointed out in section 9.

Literature review. To the best of the authors' knowledge, the two problems addressed in this paper have never been considered before in the literature. In particular, the way uncertainty is modeled appears to be original in the dynamic network environment.

In the more general framework of the dynamic systems, the nonstochastic model of uncertainty adopted here traces back to 1971 with the concept of "set constrained disturbances," which was developed in the seminal papers by Bertsekas and Rhodes [7] and by Glover and Schweppe [13]. Further results in this more general area have been derived by Morris and Brown in 1976 [27], Gutman and Cwikel in 1986 [17], Keerthi and Gilbert in 1987 [21], and, more recently, Blanchini and Ukovich in 1993 [8].

Besides this peculiar way of tackling uncertainty, another basic ingredient of this paper are dynamic networks, that is, networks in which flows evolve with time. As has been pointed out, the usual approach to dynamic network problems is by the so-called time-expanded models [2]. Besides the fact of being rather cumbersome since they require to duplicate the network of interest as many times as the time horizon of the problem, time-expanded models are not suitable to tackle uncertainties such as they are considered in this paper. Instead, the approach adopted here stems from the concept of "target tube," introduced for set constrained disturbances in the same papers by Bertsekas and Rhodes and by Glover and Schweppe. Such a method, which is amenable to basic concepts of dynamic programming (see, for instance, [5] again), has been used in [9] for a problem similar to the first one of this paper, in which the demand pattern evolves periodically through time but in a deterministic way. In this sense, the first problem of this paper can be considered as an extension of the problem considered in [9] to the case of unknown demand but with no periodic evolution.

The second problem of this paper (that is, the design problem) belongs to the large class of the network design problems, which are widely studied in the literature. For an extensive review, see [23]. The problem considered here is original as is the approach proposed for it, which relies on the results derived for the first problem.

Practical applications. The practical interest for production–distribution systems does not need to be emphasized: the relevant literature is very large and well documented (see, for example, [14]). Incidentally, it is worth noticing that our approach, considering feedback control strategies, complies with the *Just–In–Time* philosophy in production management systems (see, for instance, [16], [18]). Indeed, production-replenishment orders are issued on the basis of available buffer/inventory levels.

It could be appropriate to briefly mention some examples of practical situations in which our model, and in particular the way we consider demand uncertainty, could be conveniently applied.

An interesting example of a possible practical application of the network design problem we study in this paper is in negotiations with suppliers [34]. Consider the case of a supply contract for the repeated delivery, on a long time horizon, of a given quantity of a certain commodity. Each time, part of the demanded quantity is requested at a certain delivery point and the rest at a different location. The splitting ratio is unpredictable, so the supplier must always be ready to face any demand shared between the two locations.

Clearly, such a condition requires some degree of flexibility for the supply system. That is, appropriate stocks should be maintained at the delivery points, extra total production capacity possibly should be provided (especially if production is performed in situ at the delivery points), and, finally, the possibility of transshipments of endproducts between the delivery points should be contemplated. Clearly, such conditions all imply costs.

Now the question is how much to charge for such a costly flexibility in the supply contract. In particular: can costs related to stock capacities be traded off with costs related to production or transportation capacities? Does there exist a particular distribution of the demand such that if capacities are provided to meet it, could any other feasible demand split also be faced? Or would it be wiser to be ready to face either of the two situations in which the whole demand concentrates on just one delivery point? A situation of this kind will be addressed in section 8.

Another possible practical application of the same model is to assess the cost of the flexibility necessary to guarantee that lost sales are never incurred [34].

From the point of view of practical applications, it is also worth pointing out that the demand model we adopt is convenient for dealing with product competition [26] or product substitution phenomena: they easily can be dealt with by demand arcs connecting the nodes associated with the competing products.

A quite different practical application of our models refers to human resource management problems (see, for instance, [24]) and deals with formulating a sequential plan for allocating personnel to jobs and roles. In this case, job positions are represented by nodes with given capacity limits. The uncontrolled arcs model the autonomous evolution of a workforce (automatic promotions, retirements, change of site, etc.), which may be unpredictable to some extent. Controlled arcs represent personnel acquisition, development, and allocation activities. The problem consists of determining bounds on personnel management activities that allow compensation of the actual autonomous evolution of workforce availability.

2. Model and problem statement.

Let $G = (N, E)$ be an oriented multigraph, where the nodes of N represent warehouses in which a certain commodity can be stored, and the arcs of E represent transportation links through which the commodity can be moved. The amount of commodity present in the ith node of N at the time t is denoted by $x_i(t)$ and the corresponding vector $x(t)$ is assumed to satisfy the constraint

$$x \in X = \left\{ x \in \Re^n : \ x^- \leq \ x \ \leq x^+, \ \sum_{i=1}^{n} x_i = 0 \right\}, \tag{1}$$

where x^- and x^+ are given vectors of $\Re^n$. The reason why it is assumed that the sum of all the components of x is 0 is that by possibly including the external environment in the model by adding an auxiliary node and proper arcs between this node and the original nodes of the network, the system can always be supposed to be isolated. This means that the global amount of the commodity present in the system is constant through time and, without restriction, this quantity may be assumed to be zero.

In this setting, a game between two players $\mathcal{P}$ and $\mathcal{Q}$ is considered. The set E is partitioned into two subsets $E_\mathcal{P}$ and $E_\mathcal{Q}$ in such a way that at each time, player $\mathcal{P}$ decides the flows $u(t)$ of the arcs of $E_\mathcal{P}$ and player $\mathcal{Q}$ decides the flows $d(t)$ of the arcs of $E_\mathcal{Q}$. Two parallel arcs between each pair of nodes are allowed; each one is controlled by a different player. The flows $u(t)$ and $d(t)$ have to satisfy the following constraints:

$$u(t) \in U = \{u \in \Re^p : \ u^- \ \leq \ u \ \leq \ u^+\}, \tag{2}$$

$$d(t) \in D = \{d \in \Re^q : \ d^- \ \leq \ d \ \leq \ d^+\}, \tag{3}$$

where $p = |E_\mathcal{P}|$, $q = |E_\mathcal{Q}|$, and $u^-, u^+ \in \Re^p$, $d^-, d^+ \in \Re^q$ are assigned vectors. The information about X, U, and D is known to each player.

The discrete-time dynamic model that describes the evolution of the system is

$$x(t+1) = x(t) - Pu(t) - Qd(t), \tag{4}$$

where P and Q are, respectively, the incidence matrices of the subgraphs $G_\mathcal{P} = (N, E_\mathcal{P})$ and $G_\mathcal{Q} = (N, E_\mathcal{Q})$ (that is, the (i, e) element of P and Q is $+1$ if the arc e leaves node i and -1 if arc e enters node i and 0 otherwise).

The following dynamic game is considered. For a certain initial distribution $x(0)$ of the commodity within the nodes at time $t = 0$, player $\mathcal{P}$ chooses a flow $u(0)$ according to (2) in the arcs of $E_{\mathcal{P}}$ and player $\mathcal{Q}$ chooses a flow $d(0)$ in the arcs of $E_{\mathcal{Q}}$ according to (3). These moves produce a new distribution $x(1)$ of the commodity according to (4). Then the two players choose new flows $u(1)$ and $d(1)$ in their feasible ranges in order to produce $x(2)$ and so on. The aim of player $\mathcal{P}$ is to assure that $x(t)$ is always feasible with respect to the constraints (1), while the effort of player $\mathcal{Q}$ is to drive $x(t)$ out of X.

The first problem considered in this paper is that of finding a winning feedback strategy for player $\mathcal{P}$, that is, a function $\Phi : X \times \mathcal{N} \to U$ of the form $\Phi(x(t), t) = u(t)$ which guarantees him to win the game.

PROBLEM A. *Given constraints* (1), (2), *and* (3), *determine (if it exists) a function* $\Phi : X \times \mathcal{N} \to U$ *and an initial condition set* $X_0 \subseteq X$ *such that for all* $x(0) \in X_0$ *and for all* $d(t) \in D$, $t \geq 0$, *the sequences* $x(t)$ *and* $u(t)$ *produced by* (4) *when* $u(t) = \Phi(x(t), t)$ *are always feasible, in the sense that* $u(t) \in U$ *and* $x(t) \in X$.

A set $X_0 \subseteq X$ and a function Φ that solve Problem A will be said to be *feasible initial condition set* and *feasible (or winning) strategy*, respectively. The assumption that the strategy Φ does not depend on d is equivalent to the fact that, at each time, $\mathcal{Q}$ moves after $\mathcal{P}$. In other words, we are considering the "control plays first" game in [7].

One easily realizes that a winning strategy for player $\mathcal{P}$ does exist if, roughly speaking, the warehouses are sufficiently large and the constraints for U are not too tight. Since in practice the boxes U and X are associated to arc capacity and warehouse size, making them large enough implies a cost. This leads to a design problem which aims at finding a minimum cost network for which a winning strategy for player $\mathcal{P}$ does exist. The decision variables of such a problem are in a natural way the lower and upper bounds x^-, x^+, u^-, u^+ that define the sets X and U in (1) and (2), respectively. We assume that the construction costs of the production/transportation lines are mutually independent functions. Moreover, in order to include in the model possible feasibility constraints, we consider lower and upper bounds on each variable. The design problem we consider can then be stated in the following form.

PROBLEM B. *Given an oriented multigraph* $G = (N, E)$ *and a partition* $E = E_{\mathcal{P}} \cup E_{\mathcal{Q}}$ *of the arc set* E, *let* $x_L^-, x_U^-, x_L^+, x_U^+ \in \Re^n$, $u_L^-, u_U^-, u_L^+, u_U^+ \in \Re^p$, *and* $d^-, d^+ \in \Re^q$ *be assigned vectors. Consider a cost function for the network* $G_{\mathcal{P}} = (N, E_{\mathcal{P}})$ *of the form*

$$J(x^-, x^+, u^-, u^+) = J_1(x^-, x^+) + J_2(u^-, u^+), \tag{5}$$

where J_1 *and* J_2 *are linear cost functions not decreasing in each component of* $-x^-, x^+$ *and* $-u^-, u^+$, *respectively.*

Minimize $J(x^-, x^+, u^-, u^+)$ *under the condition that*

(i) *Problem* A *has a solution*;

(ii) *the constraints*

$$x_L^- \leq x^- \leq x_U^-, \qquad x_L^+ \leq x^+ \leq x_U^+, \qquad x^+ - x^- \geq 0, \tag{6}$$

$$u_L^- \leq u^- \leq u_U^-, \qquad u_L^+ \leq u^+ \leq u_U^+, \qquad u^+ - u^- \geq 0 \tag{7}$$

are satisfied.

Assuming that the cost functions J_1 and J_2 are nondecreasing with respect to the components of $-x^-, x^+$ and $-u^-, u^+$ is a reasonable assumption since warehouses or production units with larger capacity usually imply larger costs. Note that whereas Problem B is an optimization problem involving design costs associated to arc and node capacities, Problem A is formulated as a mere feasibility problem, since it just requires that conditions (1) and (2) are always satisfied, without considering operational costs associated to control strategies. In fact, in section 3 a solution X_0 for Problem A will be provided which is optimal in the sense that it contains all the initial conditions in X for which a winning strategy for Player $\mathcal{P}$ exists (see Theorem 3.1). Moreover, the solution provided for Problem A easily can be exploited to find a strategy that on-line optimizes operational costs.

Problem B has a particular structure. With respect to the objective function and constraints (6) and (7), the problem is separable in the $x^{\pm}$ and $u^{\pm}$ variables. However, condition (i) does not show such a property. A basic result of this paper shows that Problem B actually can be split into two independent problems, one concerning the arc capacity and one concerning the storage capacity.

3. Solution of Problem A.

In this section the conditions are investigated for the existence of a strategy for player $\mathcal{P}$ which assures him to keep the system within its constraints on an infinite horizon. To this aim the same approach is used as in [7], [8], [9], and [13], where the more general case of linear discrete-time systems with control and state constraints is considered.

Given two sets $X, S \subseteq \Re^n$, the *erosion* of X with respect to S is defined as

$$X_S = \{\ x \in \Re^n : x + s \in X \quad \forall s \in S\ \}; \tag{8}$$

the *opposite* of X is defined as $-X = \{\ x \in \Re^n : x = -y \quad \text{for some} \quad y \in X\ \}$ and the *sum* is defined as $X + S = \{\ z \in \Re^n : z = x + s \quad \text{for some} \quad x \in X, s \in S\}$. Moreover, a set of the form $\{y \in \Re^n : y^- \leq y \leq y^+\}$ for assigned $y^-, y^+ \in \Re^n$ is said to be a *box*.

The following theorem provides necessary and sufficient conditions for the existence of a winning strategy for Player $\mathcal{P}$. These conditions require (i) the existence of feasible states that cannot be driven out of X by the disturbance in one step and (ii) that each move of the disturbance can be counteracted by a move of player $\mathcal{P}$. Moreover, the theorem provides the description of the set of all the initial conditions for which a winning strategy for Player $\mathcal{P}$ exists.

THEOREM 3.1. *Problem* A *has a solution if and only if the following two conditions are satisfied:*

$$X_{-QD} \neq \emptyset, \tag{9}$$

$$-QD \ \subseteq PU. \tag{10}$$

Moreover, the set of all the initial conditions for which the game is favorable to player $\mathcal{P}$ is given by

$$X_0 = (X_{-QD} + PU) \cap X, \tag{11}$$

and any function $\Phi(x,t)$ such that

$$\Phi(x,t) \in U, \tag{12}$$

and

$$x - P\Phi(x,t) \in X_{-QD} \qquad \textit{for all } x \in X_0,\ t \geq 0 \tag{13}$$

is a strategy that solves Problem A.

Proof. The necessity of condition (9) follows by noticing that if X_{-QD} is empty, then for each $x \in \Re^n$ there exists $d \in D$ such that $x - Qd \notin X$. In particular, for each $x \in X$ and $u \in U$, $x - Pu - Qd \notin X$ for a suitable $d \in D$. The necessity of condition (10) is also easy to prove. Indeed, let $d^* \in D$ such that $-Qd^* \notin PU$. Since PU is a closed convex subset of $\Re^n$, by the separation theorem (see [33]) there exists a hyperplane in $\Re^n$ that strongly separates PU from $-Qd^*$; that is, there exist $z \in \Re^n$ and $\epsilon > 0$ such that $-zQd^* \geq zPu + \epsilon$ for every $u \in U$. Then, if $x_0 \in X$ and we chose $d(t) = d^*$ for every $t \geq 0$, from (4) we obtain

$$x(t) = x_0 - \sum_{i=0}^{t-1}(Pu(i) + Qd^*)$$

for each possible sequence $\{u(i)\}_{i=0}^{t-1}$ such that $u(i) \in U$ for all i, and thus

$$zx(t) = zx_0 - \sum_{i=0}^{t-1} z(Pu(i) + Qd^*) \geq zx_0 + \epsilon t.$$

Since X is bounded, $x(t) \notin X$ for t sufficiently large.

Conditions (9) and (10) are also sufficient. First, they imply that the set X_0 defined in (11) is not empty. Indeed, by (9), there exists $x_0 \in X_{-QD}$ and, for each $d \in D$, $x_0 - Qd \in X_0$ since $x_0 - Qd \in X_{-QD} + PU$ by (10) and $x_0 - Qd \in X$ by (8). Now, for each $x(0) \in X_0$, there exists $u(0) \in U$ such that $x(0) - Pu(0) \in X_{-QD}$, which implies $x(1) = x(0) - Pu(0) - Qd(0) \in X$ for every $d(0) \in D$. By (10), for each $d(0) \in D$, there exists $u \in U$ such that $Pu = -Qd(0)$. Therefore $x(1) \in X_0$, too. By reproducing the same argument for $x(1)$, it may be shown that x_0, and thus all the points in X_0, define initial conditions that solve Problem A. The set X_0 is the maximal subset of X with respect to this property since, for every $x \in X \setminus X_0$, $x - Pu \notin X_{-QD}$ for every $u \in U$. The last statement of Theorem 3.1 follows in an obvious way. □

Note that the previous result holds in general for every system of the form (4) if U and D are closed convex sets and P and Q are real matrices. However, all the results which follow are consequences of the particular structure of the sets U and D, which are boxes, and of the fact that P and Q are incidence matrices. Note also that for the sake of generality, time-varying strategies $\Phi(x, t)$ have been considered, but from the conditions (12) and (13), it follows that if Problem A has a solution, then a time invariant strategy $\Phi(x)$ always exists. This function Φ does not have an explicit expression but is defined in an implicit way as a function which associates to x any one of the elements of the set

$$U(x) = \{u \in U : -Pu \in Y(x)\} \tag{14}$$

where

$$Y(x) \doteq \{-x\} + X_{-QD}.$$

Since, as will be shown in Lemma 3.5, the set $Y(x)$ is the intersection of a box with the hyperplane $\sum_{i=1}^{n} x_i = 0$, it turns out that the problem of determining an element in $U(x)$ reduces to a feasible flow problem in the graph $G_{\mathcal{P}}$. If an operational cost on the flow u is introduced, one has just to cope with a minimum cost flow problem, which has to be solved on-line and for which efficient algorithms exist.

We point out that the conditions for the existence of a winning strategy for player $\mathcal{P}$ have a fundamental *separation property* as they are given by two separate conditions (9) and (10), concerning the buffer capacities and the sets QD and PU. Moreover, it turns out that the complexity of the strategy which solves the problem is an instance-independent function of n and p, known a priori, and it requires algorithms whose complexity is polynomial in n and p.

Remark 3.2. A singular property of these results is that they hold only for our infinite-horizon game while they do not hold in general for the finite-horizon game. In other words, suppose we have the problem of finding a strategy which meets the constraints over an assigned horizon $0, 1, \dots, K$. Then, according to [7], we have to construct a sequence of "feasibility sets" called the target tube. It turns out that these sets are not boxes in general, even for the network case, and that the conditions for the existence of a finite-horizon winning strategy do not have the separation property of the infinite-horizon corresponding ones. Roughly speaking, the finite-horizon problem is much more difficult than the infinite-horizon one. This is an unusual situation in dynamic game theory (see, for instance, [3], [4]), where the infinite-horizon solution is usually derived as the limit for $K \to \infty$ of the finite-horizon one.

Remark 3.3. There is another version of the game in which player $\mathcal{Q}$ starts the game. In this case it is admitted that at each time instant, the controller knows the level of the demand before making his decision. This assumption may be reasonable in some cases. It easily can be proved that in this case the feasible initial condition set is $X_0 = X$, and Theorem 3.1 simply has to be modified by replacing condition (9) with the condition that X is not empty. This version of the game can be handled in a similar way as the original game, and it will not be further mentioned.

Theorem 3.1 says that in order to give a yes or no answer to the question "does a strategy exist that solves Problem A," one has just to check if the set X_{-QD} is not empty and if inclusion (10) holds. These conditions also have to be satisfied by every solution of Problem B, so the next task will be that of expressing them in terms of a minimal set of linear constraints in the variables $u^{\pm}$ and $x^{\pm}$ in order to reduce as much as possible the complexity of Problems A and B.

As will be seen, checking if X_{-QD} is not empty requires verifying whether or not a box intersects a hyperplane and this task can be easily accomplished. On the other hand, condition (10) requires checking the inclusion of the two polyhedra $-QD$ and PU. As will be explained in section 4, this problem is in fact an NP-hard problem, despite the particular structure of the polyhedra involved. Nevertheless, the structure of these sets allows us to rephrase condition (10) in a form which is convenient to solve Problems A and B. The complexity of this solution method consistently lessens when the controlled network has not too many arcs. These results rely on the particular structure of the sets PU and QD. To present them, we first introduce some notations.

Let $G = (N, E)$ be a directed graph with $|N| = n$ and $|E| = m$. For each subset S of N and for each vector $x \in \Re^n, x(S) = \sum_{i \in S} x_i$ is the sum of the components associated to all the nodes in S. We denote by $\delta(S)$ the cut corresponding to S, that is, the subset of E whose arcs have one extremity in S and the other one in $N \setminus S$. Evidently, $\delta(S) = \delta(N \setminus S)$. Let $\delta^+(S)$ $(\delta^-(S))$ denote the set of arcs in E having the initial (terminal) node in S and the terminal (initial) node in $N \backslash S$. For a vector $u \in \Re^p$ and an arc set $F \subseteq E$, let $u(F) = \sum_{e \in F} u_e$. When a flow capacity interval $[u_e^-, u_e^+]$ is assigned to each arc of E, we call the quantity $c(S) = u^+(\delta^+(S)) - u^-(\delta^-(S))$ the *capacity of the cut* $\delta(S)$. It represents the maximum amount of positive flow that can pass from S to $N \setminus S$, given the assigned arc capacity constraints.

If we denote by $\delta_{\mathcal{P}}(S)$ and $\delta_{\mathcal{Q}}(S)$ the cuts defined by a subset S of N in the graphs $G_{\mathcal{P}} = (N, E_{\mathcal{P}})$ and $G_{\mathcal{Q}} = (N, E_{\mathcal{Q}})$, respectively, then the vectors ξ, η, $\theta \in \Re^{2^n-2}$ defined by

$$\xi_S = u^+(\delta^+_{\mathcal{P}}(S)) - u^-(\delta^-_{\mathcal{P}}(S)), \tag{15}$$

$$\eta_S = d^+(\delta^+_{\mathcal{Q}}(S)) - d^-(\delta^-_{\mathcal{Q}}(S)), \tag{16}$$

$$\theta_S = d^+(\delta^-_{\mathcal{Q}}(S)) - d^-(\delta^+_{\mathcal{Q}}(S)) \tag{17}$$

for each proper subset S of N have components ξ_S and η_S that represent the capacities of $\delta_{\mathcal{P}}(S)$ and $\delta_{\mathcal{Q}}(S)$, respectively. Moreover, $\theta_S = \eta_{N \setminus S}$. Now, by the Gale–Hoffman theorem (see, for instance, [32]), it holds that

$$PU = \{x \in \Re^n : x(S) \leq \xi_S \ \forall S \subset N,\ x(N) = 0\}, \tag{18}$$

$$QD = \{x \in \Re^n : x(S) \leq \eta_S \ \forall S \subset N,\ x(N) = 0\}, \tag{19}$$

$$\text{and} \qquad -QD = \{x \in \Re^n : x(S) \leq \theta_S \ \forall S \subset N,\ x(N) = 0\}. \tag{20}$$

The sets PU, QD, and X are all *zero-base polyhedra.* This means that they have the form

$$B(f) = \{x \in \Re^n : x(S) \leq f(S) \ \forall S \subset N,\ x(N) = 0\}, \tag{21}$$

where $f : 2^N \to \Re$ is a *submodular function*, that is, a function which satisfies the condition

$$f(S \cup T) + f(S \cap T) \leq f(S) + f(T) \qquad \forall S, T \subseteq N. \tag{22}$$

We also assume that $f(\emptyset) = 0$ for every submodular function f.

The next proposition collects some properties of zero-base polyhedra that are used in the following.

PROPOSITION 3.4. *Let f, g be two submodular functions and $B(f)$, $B(g)$ be the corresponding zero-base polyhedra. Then the following properties hold:*

(i) *for each $S \subseteq N$, the inequality $x(S) \leq f(S)$ is* tight *in the sense that* $\max_{x \in B(f)} x(S) = f(S)$ (*this also implies* $\min_{x \in B(f)} x(S) = -f(N \setminus S)$);

(ii) *if f is an integer valued function, then the vertices of $B(f)$ are integer vectors*;

(iii) *$f + g$ is a submodular function and $B(f) + B(g) = B(f + g)$*;

(iv) *for each box X, $B(f) \cap X$ is a zero-base polyhedron and it has integer vertices if both X and $B(f)$ are integer polyhedra.*

Proof. See [10] and [11]. □

In the next lemma a description of the set X_{-QD} is given.

LEMMA 3.5. *The set X_{-QD} has the form*

$$X_{-QD} = \{x \in \Re^n : x_i^- + \theta_{N \setminus \{i\}} \ \leq x_i \leq \ x_i^+ - \theta_{\{i\}},\ \ i = 1, 2, ..., n,\ \ x(N) = 0\}. \tag{23}$$

Such a set is not empty if and only if the following conditions are satisfied:

$$x_i^- + \theta_{N \setminus \{i\}} \ \leq x_i^+ - \theta_{\{i\}} \qquad i = 1, 2, \ldots, n, \tag{24}$$

$$\sum_{i=1}^{n} (x_i^- + \theta_{N \setminus \{i\}}) \ \leq \ 0 \ \leq \ \sum_{i=1}^{n} (x_i^+ - \theta_{\{i\}}). \tag{25}$$

Proof. By (1), $X = X^* \cap \pi$ where X^* is the box defined by vectors x^-, x^+ and $\pi = \{x \in \Re^n : x(N) = 0\}$. It is easy to verify that since $-QD \subseteq \pi$, $X_{-QD} = X^*_{-QD} \cap \pi$. Now

$$\begin{aligned} X^*_{-QD} &= \{x \in \Re^n : x - Qd \in X^* \quad \forall\, d \in D\}, \\ &= \{x \in \Re^n : x^- + Qd \leq x \leq x^+ + Qd \quad \forall\, d \in D\}, \\ &= \{x \in \Re^n : x_i^- + \max_{d \in D}(Qd)_i \leq x_i \leq x_i^+ + \min_{d \in D}(Qd)_i, \quad 1 \leq i \leq n\}, \\ &= \{x \in \Re^n : x_i^- + \eta_{\{i\}} \leq x_i \leq x_i^+ - \eta_{N\setminus\{i\}}, \quad 1 \leq i \leq n\}, \\ &= \{x \in \Re^n : x_i^- + \theta_{N\setminus\{i\}} \leq x_i \leq x_i^+ - \theta_{\{i\}}, \quad 1 \leq i \leq n\}, \end{aligned}$$

and thus X_{-QD} has the form (23). It is immediate to see that $X^*_{-QD} \cap \pi \neq \emptyset$ implies conditions (24) and (25). Sufficiency follows by noticing that if (25) is satisfied, then $\min_{x \in X^*_{-QD}} x(N) \leq 0 \leq \max_{x \in X^*_{-QD}} x(N)$, and thus there exists $x \in X^*_{-QD}$ such that $x(N) = 0$. □

Note that since the set function h defined by $h(S) = \sum_{i \in S}(x_i^+ - \theta_{\{i\}})\ \forall S \subset N$, $h(N) = 0$ is a submodular function and, as easily follows from (23), $X_{-QD} = B(h)$, then X_{-QD} is a zero-base polyhedron, too.

The next result specifies condition (10).

LEMMA 3.6. *The condition $-QD \subseteq PU$ holds if and only if*

$$\theta_S \leq \xi_S \quad \forall\ S \subset N. \tag{26}$$

Proof. Sufficiency follows immediately from (18) and (20). To prove necessity, we just have to consider point (i) of Proposition 3.4, according to which each inequality in (18) and (20) is tight. □

4. Complexity of Problems A and B.

By Theorem 3.1 and Lemmas 3.5 and 3.6, in order to prove the existence of a solution for Problem A, one needs to check conditions (24) and (25) (which assure that the set X_{-QD} is not empty) and conditions (26) (which guarantee that each element of the form $-Qd$, $d \in D$ is also an element of PU). The first part requires, besides elementary operations, evaluating the function θ in the $2n$ sets of the form $\{i\}$ and $N \setminus \{i\}$ for each $1 \leq i \leq |N|$. On the other hand, despite the very simple form of conditions (26), it turns out that verifying if $-QD \subseteq PU$ requires checking as many as $2^n - 2$ constraints. One could hope that, in view of the fact that the functions ξ and θ are in fact defined in terms of the vectors $u^-, u^+ \in R^p$ and $d^-, d^+ \in R^q$, respectively, and thus have a polynomial representation with respect to the dimension of the problem, their comparison might be accomplished in polynomial time. Unfortunately, this is not the case. McCormick has indeed proved in [25] the following results.

THEOREM 4.1. *Let $G = (N, E)$ be a complete directed graph (that is, $(i, j) \in E$ for each $i, j \in N$). Given two nonnegative functions $u_i : E \to \Re$, $i = 1, 2$, consider the cut capacity functions f and g defined by u_1 and u_2, respectively. Then it is strongly NP-complete to decide if $B(f) \subseteq B(g)$ (network submodular containment problem).*

The condition (10) for Problem A corresponds to the network submodular containment problem (NSCP), and thus Problem A is NP-complete, too. Its complexity strongly affects the complexity of Problem B, since inequalities (26) also appear as constraints in the formulation of the design problem. In fact, the arc-related subproblems of Problems A and B correspond to the strong membership problem and the strong optimization problem, respectively, for the polyhedron given by

$\{(u^-, u^+) : \theta_S \leq \xi_S(u^-, u^+)$ for all $S \subset N\}$, where θ is an assigned cut capacity function. Then, by the previous result and Theorem 6.4.9 in [15], it immediately follows that since NSCP is NP-complete, Problem B also cannot be solved in polynomial time [25].

Corollary 4.2. *There is no polynomial algorithm for Problem* B *unless $P = NP$.*

5. Minimal characterization of the polyhedron PU and solution of Problem B.

The results of the previous section leave no hope of solving Problems A and B by a polynomial method. This is particularly awkward when large scale instances are addressed, since the computational burden they involve may turn out to be unrealistic. A first approach to overcome such a drawback is to try at least to reduce as much as possible the number of inequalities in (26). It is easy to realize that one can consider only those constraints that are nonredundant for the polyhedron PU, that is, those which cannot be removed from (18) without modifying the set they represent. In this section we look for the minimal description of the polyhedron PU. An alternative approach to handling large scale instances of Problem B by providing an approximate solution is proposed in section 7.

Now we study how the structure of the graph $G_{\mathcal{P}} = (N, E_{\mathcal{P}})$ reflects on the minimal characterization of the polyhedron PU in terms of the inequalities $x(S) \leq \xi_S$.

Definition 5.1. *Given a graph $G = (N, E)$, a cut $\delta(S)$ is said to be* disconnecting *G if one of the two disjoint subgraphs $G_S = (S, E_S)$ and $G_{N\setminus S} = (N \setminus S, E_{N\setminus S})$ obtained by removing all the arcs of $\delta(S)$ is not connected. Otherwise, the cut is said to be* nondisconnecting.

Lemma 5.2. *If the graph $G_{\mathcal{P}}$ is connected and the set U is full dimensional* (*i.e., $u_i^- < u_i^+$ for all i*)*, then*

(i) *the inequality $x(S) \leq \xi_S$ is nonredundant in* (18) *if and only if the cut $\delta(S)$ is not disconnecting $G_{\mathcal{P}}$.*

Moreover, if $n = |N|$ and $r = |E_{\mathcal{P}}| - n + 1$, then

(ii) *for each nonredundant inequality $x(S) \leq \xi_S$ of PU the corresponding cut $\delta(S)$ has at most $r + 1$ arcs*;

(iii) *the number of nonredundant inequalities of PU is bounded above by $O(n^{r+1})$.*

Proof. (i): For each $S \subset N$, the set $E_{\mathcal{P}}$ can be split as $E_{\mathcal{P}} = E_S \cup E_{N\setminus S} \cup \delta(S)$, where E_S and $E_{N\setminus S}$ are the subsets of arcs having both extremities in S and $N \setminus S$, respectively. In a similar way, the set U splits in $U = U_S \times U_{N\setminus S} \times U_{\delta(S)}$ and the incidence matrix P of $G_{\mathcal{P}}$ may be written (possibly by reordering the nodes) in the form

$$P = \begin{pmatrix} P_S & 0 & \\ 0 & P_{N\setminus S} & P_{\delta(S)} \end{pmatrix},$$

where the columns of P_S, $P_{N\setminus S}$, and $P_{\delta(S)}$ represent the arcs of E_S, $E_{N\setminus S}$, and $\delta(S)$, respectively. Since $G_{\mathcal{P}}$ is connected and U has full dimension, the polyhedron PU has dimension $n - 1$. Let

$$F_S = \{x \in PU : x(S) = \xi_S\}$$

be the face of PU defined by the inequality $x(S) \leq \xi_S$. This constraint is nonredundant for PU if and only if dim F_S = dim $PU - 1 = n - 2$. The dimension of F_S is univocally determined by the rank of the incidence matrices P_S and $P_{N\setminus S}$. Indeed, let $u^0 \in U_{\delta(S)}$ be the vector whose components are $u_j^0 = u_j^+$ if $j \in \delta^+(S)$ and $u_j^0 = u_j^-$

if $j \in \delta^-(S)$. Immediately we see that for every $u = (u_S, u_{N\setminus S}, u_{\delta(S)}) \in U$, the condition $Pu(S) = \xi_S$ holds if and only if $u_{\delta(S)} = u^0$, and thus

$$F_S = \left\{ x \in \Re^n : \quad x = \begin{pmatrix} P_S & 0 \\ 0 & P_{N\setminus S} \end{pmatrix} \begin{pmatrix} u_S \\ u_{N\setminus S} \end{pmatrix} + P_{\delta(S)} u^0 : \quad u_S \in U_S, \quad u_{N\setminus S} \in U_{N\setminus S} \right\}.$$

In particular, dim F_S = rank P_S + rank $P_{N\setminus S}$ and thus dim $F_S = n - 2$ if and only if both of the graphs $G_S = (S, E_S)$ and $G_{N\setminus S} = (N \setminus S, E_{N\setminus S})$ are connected.

(ii) Let $x(S) \leq \xi_S$ be a nonredundant inequality. Since by (i) the graphs G_S and $G_{N\setminus S}$ are both connected, then both the conditions $|E_S| \geq |S| - 1$ and $|E_{N\setminus S}| \geq |N \setminus S| - 1$ hold, so that $\delta(S)$ has at most $|E_{\mathcal{P}}| - n + 2 = r + 1$ elements.

(iii) By (ii), the number of cuts corresponding to nonredundant inequalities is trivially bounded above by the number of subsets of $E_{\mathcal{P}}$ which have at most $r + 1$ elements. Since any cut is associated with the pair of constraints $x(S) \leq \xi_S$ and $x(N \setminus S) \leq \xi_{N\setminus S}$, it is clear that the number of nonredundant inequalities of the polyhedron PU cannot be larger than $k(n, r)$, where $k(n, r)$ is the polynomial function in n of degree $r + 1$ given by

$$k(n,r) = 2\left[\binom{n+r-1}{r+1} + \binom{n+r-1}{r} + \binom{n+r-1}{r-1} + \cdots + \binom{n+r-1}{1}\right]. \tag{27}$$

□

Remark 5.3. It is important to note that if U is not full dimensional, the "only if" part of proposition (i) still holds; namely, $x(S) \leq \xi_S$ is redundant for PU if the cut $\delta(S)$ is disconnecting $G_{\mathcal{P}}$.

Part (i) of Lemma 5.2 has been independently proved by Wallace and Wets in [35]. However, the above proof seems simpler and it is reported here for sake of completeness.

By Lemma 5.2, if the number of independent circuits r is fixed, then the number of independent constraints in (18) is polynomial in n. In particular, it is linear when the graph is a tree, which is a typical situation in distribution systems [29]. The upper bound in (iii) is, in general, very conservative, and in almost every case the number of nonredundant inequalities is much smaller and can be a priori determined by performing a connectivity test on the subgraphs generated by each cut. However, there exist families of graphs for which the given bound is tight, and thus it cannot be improved.

Lemma 5.2 leads to the following corollary, which reduces the number of the inequalities (26) that actually need to be verified.

COROLLARY 5.4. *Let I be the subset of 2^N defined by*

$$I = \{S \subset N : \ \textit{the cut } \delta(S) \textit{ is nondisconnecting } G_{\mathcal{P}}\}. \tag{28}$$

Under the hypotheses of Lemma 5.2, *condition* (10) *is satisfied if and only if*

$$\theta_S \leq \xi_S \quad \textit{for all } S \in I. \tag{29}$$

Proof. Due to Lemma 3.6, it is sufficient to show that (29) implies (26). Let conditions (29) hold and consider a disconnecting cut $\delta(S)$ of $G_{\mathcal{P}}$. By Lemma 5.2,

$x(S) \leq \xi_S$ is a tight redundant constraint for PU. Then there exist $S_1, S_2, ..., S_k \subset N$, a vector $z \in \Re^k_+$, and $w \in \Re$ such that $x(S_j) = \xi_{S_j}$ is a facet of PU for each $1 \leq j \leq k$, $x(S) = \sum_{j=1}^k z_j x(S_j) + wx(N)$ for all $x \in \Re^n$, and $\xi_S = \sum_{i=1}^k z_i \xi_{S_i}$. Since $S_j \in I$ for each $j = 1, \dots, k$ and $x(S) \leq \theta(S)$ is a tight constraint for $-QD$, we finally obtain

$$\theta_S \quad \leq \sum_{i=1}^{k} z_i \theta_{S_i} \;\leq\; \sum_{i=1}^{k} z_i \xi_{S_i} = \xi_S. \qquad \square$$

The next theorem summarizes the results of this section and gives a complete solution for Problem B.

THEOREM 5.5. *For each instance of Problem* B*, let* θ *be the function defined in* (17)*,* I *the set introduced in* (28)*, and* P_1 *and* P_2 *the polyhedra defined by*

$$P_1 = \{(x^-, x^+) \in \Re^{2n} \;\; \textit{satisfying } (6), (24), \textit{and } (25)\},$$
$$P_2 = \{(u^-, u^+) \in \Re^{2p} \;\; \textit{satisfying } (7) \textit{ and } (29)\}.$$

Problem B *has a solution if and only if both* P_1 *and* P_2 *are not empty. In this case, the solution may be found by solving the two independent programming problems* $\{\min\ J_1 : (x^-, x^+) \in P_1\}$ *and* $\{\min\ J_2 : (u^-, u^+) \in P_2\}$*, where the former one has* $6n+2$ *linear constraints and the latter, besides conditions* (7)*, has a number of linear constraints that does not exceed the quantity* $k(n, r)$ *introduced in* (27)*.*

Proof. The assertion follows from Theorem 3.1, Lemmas 3.5, 3.6, Corollary 5.4, and Remark 5.3. $\square$

6. The integer game. In several practical problems, only integer quantities can be considered. In view of the structure of equation (4), if the initial state $x(0)$ is an integer, then $x(t)$ remains an integer for $t \geq 0$ as long as $u(t)$ and $d(t)$ are integer vectors. It is then natural to formulate an integer version for Problem A by requiring that $x(t), u(t)$, and $d(t)$ can assume only integer values. In this case, it is obvious that the bounds for U, X, and D are integers. Accordingly, a strategy Φ is said to be an integer strategy if $\Phi : \mathcal{Z}^n \rightarrow \mathcal{Z}^q$. This version of the game is referred to as the *integer game* and the original version of Problem A as the *real game.* The results obtained in the previous sections allow us to prove the following theorem.

THEOREM 6.1. *Assume that the sets* U*,* X*, and* D *have integer vertices. Then the integer game has a solution if and only if the real game formulated on the same data has a solution. Moreover, in this case the feasible initial condition set for the integer game is the set of all the integer points of the set* X_0 *defined in* (11)*, and an integer strategy may be found by solving on-line a feasible flow problem.*

Proof. Suppose that the real game has a solution, so that conditions (9) and (10) are satisfied. Then, since the data are integers and, as outlined after Lemma 3.5, X_{-QD} is a zero-base polyhedron, by (ii) in Proposition 3.4, X_{-QD} has integer vertices. Also, the polyhedron X_0 defined in (11) is an integer zero-base polyhedron as follows from Proposition 3.4, since X_0 is the intersection of the sum of the two zero-base polyhedra X_{-QD} and PU with a box. Thus, there exist integer initial conditions from which Player $\mathcal{P}$ wins the game. Each integer-feasible strategy requires choosing, for an assigned integer state $x_0 \in X_0$, an element u in the set $U(x)$ defined in (14). A flow u belongs to $U(x)$ if and only if it satisfies the capacity constraints defined by U and produces a divergence vector Pu contained in $Y(x) = X_{-QD} + \{-x_0\}$, that is, by (23), if and only if it satisfies the constraints

$$(30)\;\; x_i^- + \theta_{N \setminus \{i\}} - (x_0)_i \;\leq (Pu)_i \leq\; x_i^+ - \theta_{\{i\}} - (x_0)_i, \quad i = 1, 2, ..., n, \quad x(N) = 0,$$

$$u^- \le u \le u^+. \tag{31}$$

Finding such a solution simply reduces to solving a feasible flow problem (see [32]). Indeed, the integrality theorem for flows (see [32]) assures that when data are integers and an admissible flow does exist, an integer-admissible flow exists, too. Moreover, such an integer solution may be found by using common algorithms for network flow problems. □

Concerning Problem B, it easily follows from Lemma 3.5 that if the cost function J_1 is not decreasing in each component of $-x^-$ and x^+ and all the data are integers, then the optimal real solution has integer components x^- and x^+. It is interesting to prove that the same property does not hold with respect to u^+ and u^- as the following simple example shows.

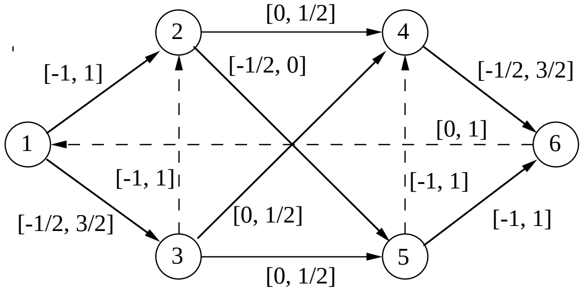


FIG. 1. *The network structure for the example.*

Example 6.2. Let us consider the network in Fig. 1, where the dotted arrows represent the demand arcs and the solid arrows represent the controlled arcs. Near each demand arc the range $[d_e^-, d_e^+]$ appears, in which the corresponding demand may vary. The capacity intervals $[u_e^-, u_e^+]$ associated with each controlled arc represent the optimal solution of Problem B with respect to the linear cost function defined by $J_2(u^-, u^+) = \sum_{e \in E} c_e(-u_e^- + u_e^+)$, where $c_{(2,5)} = c_{(3,4)} = 6$ and $c_e = 5$ for each $e \in E_{\mathcal{P}} \setminus \{(2,5), (3,4)\}$. The cost of this solution is 51. It is easy to see that there cannot be any integer solution with the same cost. First, note that the cuts $\delta^+(\{1,2,5\})$ and $\delta^+(\{1,3,4\})$ are disjoint and contain only arcs with cost capacity 5 and that the capacity of both of them must be at least 3 in every admissible solution. This implies that any integer solution of cost 51 could use only one capacity unit of capacity cost 6. But by deleting any one of the two arcs $(2,5)$ and $(3,4)$, we obtain an instance of the problem whose optimum value is 55.

Although the solution of Problem B may not be integral, the following property nevertheless holds.

PROPOSITION 6.3. *If Problem* B *with integer data is feasible, then it also admits an integer-feasible solution.*

Starting from a real optimal solution (u^-, u^+), an integer solution may be obtained simply by setting $\tilde{u}_j^+ = \lceil u_j^+ \rceil$ and $\tilde{u}_j^- = \lfloor u_j^- \rfloor$. Obviously, this may not be the optimal integer solution. For instance, in the previous example, this procedure leads to a solution of cost 72, while the optimal integer solution uses only arcs of capacity cost 5 and has value 55.

7. Special cases and an approximate algorithm. In the previous sections it has been shown that both Problems A and B can be split in two subproblems, one concerning node capacities only and the other concerning arc capacities only. While the former is easy to solve even under integrality conditions, the latter is an NP-hard problem. We have already seen that for the families of graphs with a fixed number of circuits, Problem B is polynomial. If the graph is a tree, the solution is extremely simple and it is integral for the integer problem, as the following result shows.

PROPOSITION 7.1. *Let us consider an instance of Problem* B *with integer data. If* $G_{\mathcal{P}}$ *is a tree and Problem* B *has a solution, then it admits an integer optimal solution.*

Proof. By Lemma 5.2, the only nonredundant constraints in the variables $u^{\pm}$ are $u_j^+ \geq \theta_{S_{\{j\}}}$ and $u_j^- \leq -\theta_{N \setminus S_{\{j\}}}$, where S_j identifies the cut containing only the arc e_j. So, if Problem B is feasible, then the solution satisfying these constraints as equality is optimal. □

A further interesting question is to determine if there exists a "worst case" demand, in the sense that it is sufficient to provide node and arc capacity to contrast it in order to solve Problem B. This happens in the particular, but meaningful, case in which all the demand arcs have a common final node and zero lower capacity. This case represents the situation in which there is no product competition in the sense that demands in different nodes are independent and all lead to the external node. In this case, it is immediate to see that the worst case demand exists, and it is given by $d_i(t) = d_i^+$ for all t. This is an interesting case because once such a worst case demand has been identified, it suffices to solve a minimum cost flow problem to solve Problem B, and therefore an integer optimal solution exists. However, a worst case demand does not exist in general, as will be shown for the example of the next section.

Now we present a simple procedure to obtain an approximate solution for the arc-related subproblem of the design Problem B when the cost function has the form $J_2(u^-, u^+) = \sum_{e \in E_{\mathcal{P}}} c_e^+ u_e^+ - c_e^- u_e^-$ for assigned nonnegative costs $c^-, c^+ \in \Re^p$ and there are not upper bounds on the variables $-u_e^-$ and u_e^+.

The main idea of the algorithm relies on the fact that any path $P(i,j)$ between two nodes i and j in the graph $G_{\mathcal{P}}$ intersects every cut $\delta(S)$ such that $i \in S$ and $j \notin S$. In order to find a feasible solution for Problem B, it is then sufficient to increase, for each demand arc $f = (k,l)$ with $d_f^- \leq 0 \leq d_f^+$, the capacity of each directed arc of a path $P(l,k)$ from l to k by the amount d_f^+ and the capacity of each directed arc of a path $P(k,l)$ from k to l by the amount $-d_f^-$. In this way, the capacity of each cut $\delta(S)$ such that $k \in S$ and $l \notin S$ is increased by at least $-d_f^-$, and the capacity of the cut $\delta(N \setminus S)$ is increased by at least d_f^+. In the case $d_f^- \geq 0$ (or $d_f^+ \leq 0$), the demand acts only in one direction. It is then sufficient to increase the capacity of the arcs in $P(l,k)$ ($P(k,l)$) of d_f^+ ($-d_f^-$), that is, against the worst possible case. Since we look for a low cost solution, it is natural to choose $P(l,k)$ and $P(k,l)$ as the shortest paths with respect to the capacity costs. These remarks lead to the following algorithm.

```
APPROXIMATE ALGORITHM FOR PROBLEM B.
1. For each arc $e = (i,j) \in E_{\mathcal{P}}$ set
   $u_e^- := 0$; $u_e^+ := 0$;
   $c_{ij} := c_e^+$; $c_{ji} := c_e^-$.
2. For all $f = (k,l) \in E_{\mathcal{Q}}$ do
       if $d_f^+ > 0$, find the shortest path $P^*(l,k)$ from $l$ to $k$ in $G_{\mathcal{P}}$ with
       respect to the costs $c_{ij}$;
       for all $(i,j) \in P^*(l,k)$ update:
           $u_e^+ := u_e^+ + d_f^+$ if $e = (i,j) \in E_{\mathcal{P}}$;
```

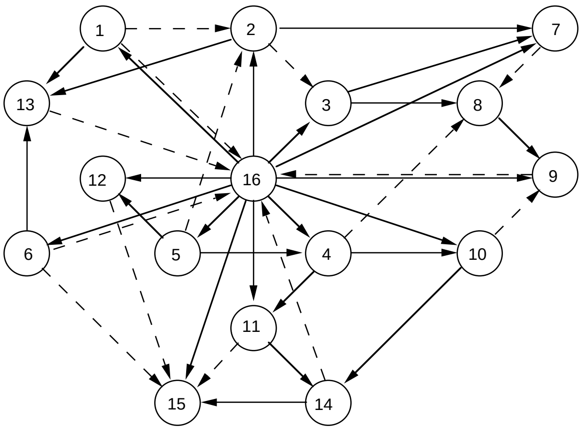


FIG. 2. *The network structure for the example.*

```
        $u_e^- := u_e^- - d_f^+$ if $e = (j,i) \in E_{\mathcal{P}}$;
    if $d_f^- < 0$, find the shortest path $P^*(k,l)$ from $k$ to $l$ in $G_{\mathcal{P}}$ with
    respect to the costs $c_{ij}$;
    for all $(i,j) \in P^*(k,l)$ update:
        $u_e^+ := u_e^+ - d_f^-$ if $e = (i,j) \in E_{\mathcal{P}}$;
        $u_e^- := u_e^- + d_f^-$ if $e = (j,i) \in E_{\mathcal{P}}$;
end do.
```

Arguing by induction, it easily can be seen that the solution $(-u^-, u^+)$ corresponding to each step of the procedure is an admissible solution for Problem B with respect to the partial uncontrolled network whose arcs are the demand arcs already processed. So the procedure ends after $|E_{\mathcal{Q}}|$ steps giving an admissible solution for Problem B. Its time complexity is $|E_{\mathcal{Q}}|O(SPP)$, where $O(SPP)$ denotes the running time of any algorithm for the shortest path problem. We note that when all the data of the problem are integers, this procedure finds in fact an integers solution. Moreover, since in a tree two nodes are connected exactly by one path, then when $G_{\mathcal{P}}$ is a tree, the algorithm gives the optimal integer solution (see also Proposition 7.1). For the example of Fig. 1 this procedure finds the solution that uses only arcs of capacity cost 5 and has value 55. This is indeed the optimal integer solution.

8. An example. In order to illustrate the results derived in the previous sections, we present a nontrivial example. Consider the network in Fig. 2 with 16 nodes, 27 controlled arcs, and 14 uncontrolled arcs.

For the node problem, we fix the lower capacity of all nodes to 0, with the exception of node 16 (the external environment), for which the upper capacity is fixed to 0. We do not impose constraints on the upper capacities of the other nodes and assume that there is no cost for the lower capacity of node 16. Then, for any nondecreasing cost function J_1, the optimal solution for the node problem is that reported in Table 3.

Concerning the arc problem, we consider a cost of the form

$$J_2(u^+, u^-) = \sum_{e \in E_{\mathcal{P}}} \alpha_e u_e^+ - \beta_e u_e^-,$$

where the values of the coefficients α_e and β_e for each controlled arc e are shown in Table 1. Finally, Table 2 contains departure and arrival nodes of the uncontrolled arcs together with their lower and upper capacity.

Table 1
Data for the controlled arcs.

arc	1	2	3	4	5	6	7	8	9	10	11	12	13
dep. node	16	16	16	16	16	16	16	16	16	16	16	16	16
arr. node	1	2	3	4	5	6	7	8	9	10	11	12	15
α_i	7	6	2	2	10	4	2	4	10	20	10	35	30
β_i	2	2	2	2	2	5	10	3	10	13	10	20	17

arc	14	15	16	17	18	19	20	21	22	23	24	25	26	27
dep. node	5	2	3	3	8	4	4	5	6	1	2	10	11	14
arr. node	4	7	7	8	9	10	11	12	13	13	13	14	14	15
α_i	7	15	1	15	4	1	14	2	8	18	4	12	11	2
β_i	3	4	1	7	3	1	3	2	2	3	4	2	3	2

We assume that there are no bounds on the upper and lower capacities of each controlled arc as well as for the upper capacities of the nodes. The number of constraints which define the polyhedron PU in (18) is 65535. However, if we apply the necessary and sufficient conditions stated in Lemma 5.2 in order to eliminate redundant constraints, we obtain that only 180 constraints are nonredundant. Then, in order to find the solution of Problem B with respect to the given instance, we have to solve a linear problem with 54 variables (the upper and the lower constraints of each controlled arc) and 180 constraints. This can be done in a straightforward way. The optimal solution is reported in Table 4 and has the cost $J_2^{opt} = 958$.

If we apply the approximated algorithm of section 7 to the same instance, we have to solve a shortest path problem 28 times. The approximate solution it finds has the cost $J_2^{apr} = 980$, which is quite close to the optimal one (about 2 %).

We note that for the case of the example, there is not a "worst case" demand; that is, there does not exist any $\bar{d} \in D$ such that the optimal solution of the problem can be obtained by replacing the constraint $QD \subseteq PU$ with $Q\bar{d} \in PU$. This can be shown by considering the uncontrolled arc d_3. If we set $d_3^- = d_3^+ = 4$ and we solve the corresponding problem, we achieve an optimal solution whose cost is $J_2 = 918$. On the other hand, if we fix $d_3^- = d_3^+ = -4$, we achieve an optimal cost of $J_2 = 902$. By convexity arguments, we deduce that the optimal solution of any problem obtained by assigning to d_3 a fixed value in $[-4, 4]$ has a cost J^* such that $J^* \leq 918 < J_2^{opt}$.

9. Conclusions.

We have studied the problem of determining a feedback control strategy for a class of single commodity production–distribution systems with nonstochastic uncertain demands using a model expressed in terms of a dynamic game on a network.

Conditions guaranteeing that a solution exists have been derived with the aim of providing means that are convenient from a computational point of view. In this sense, it has been shown how the topology of the network of interest does affect the amount of computations required.

The results obtained for this feasibility problem have then been used to solve a network design problem consisting of determining the minimum cost node and arc capacities that guarantee that a feasible control strategy exists.

TABLE 2
Data for the uncontrolled arcs.

arc	1	2	3	4	5	6	7	8	9	10	11	12	13	14
dep. node	6	5	4	9	10	14	11	12	1	1	2	7	13	6
arr. node	16	2	8	16	9	16	15	15	2	16	3	8	16	15
d_i^-	-8	-7	-8	-2	-1	-2	0	0	-4	0	-5	-3	0	-4
d_i^+	5	9	11	4	8	5	5	9	4	6	5	3	10	0

TABLE 3
The optimal values for the node problem.

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
x_i^+	14	34	10	19	16	17	6	25	15	9	5	9	10	7	18	∞

TABLE 4
The optimal solution.

arc	1	2	3	4	5	6	7	8	9	10	11	12	13
u_i^+	2	10	0	37	0	5	14	16	0	0	0	0	0
u_i^-	-4	-10	0	-2	-16	-12	0	24	0	0	0	0	0

arc	14	15	16	17	18	19	20	21	22	23	24	25	26	27
u_i^+	0	0	5	0	5	17	0	9	0	0	18	9	0	4
u_i^-	-18	-16	-5	0	-10	-12	0	0	0	-8	0	-11	-5	-14

The integer version of the problem also has been considered, in which all the variables have to assume integer values. It has been proved that if the data are integers and the two considered problems have a solution, then integer solutions also exist.

Acknowledgment. The authors are grateful to Professor McCormick for his interest in this paper and for providing them with his unpublished result.

REFERENCES

[1] R. K. AHUJA, T. L. MAGNANTI, AND J. B. ORLIN, *Network Flows: Theory, Algorithms and Applications*, Prentice–Hall, Englewood Cliffs, NJ, 1993.

[2] J. E. ARONSON, *A survey of dynamic network flows*, Ann. Oper. Res., 20 (1989), pp. 1–66.

[3] T. BASAR AND P. BERNHARD, *H^∞-Optimal Control and Related Minimax Design Problems*, Birkhäuser, Boston, MA, 1991.

[4] D. P. BERTSEKAS, *Infinite-time reachability of state-space regions by using feedback control*, IEEE Trans. Automat. Control, 17 (1972), pp. 604–613.

[5] D. P. BERTSEKAS, *Dynamic Programming: Deterministic and Stochastic Models*, Prentice–Hall, Englewood Cliffs, NJ, 1987.

[6] D. P. BERTSEKAS, *Linear Network Optimization*, MIT Press, Cambridge, MA, 1991.

[7] D. P. BERTSEKAS AND I. B. RHODES, *On the minmax reachability of target set and target tubes*, Automatica J. IFAC, 7 (1971), pp. 233–247.

[8] F. BLANCHINI AND W. UKOVICH, *A linear programming approach to the control of discrete-time periodic systems with uncertain inputs*, J. Optim. Theory Appl., 78 (1993), pp. 523–539.

[9] F. BLANCHINI, M. QUEYRANNE, F. RINALDI, AND W. UKOVICH, *A feedback strategy for periodic network flows*, Networks, 27 (1996), pp. 25–34.

[10] A. FRANK AND E. TARDOS, *Generalized polymatroids and submodular flows*, Math. Programming, 42 (1988), pp. 489–563.

[11] S. FUJISHIGE, *Submodular functions and optimization*, Ann. Discrete Math., 47, North Holland, Amsterdam, 1991.

[12] J. D. GLOVER, D. KLINGMAN, AND N. V. PHILLIPS, *Network Models in Optimization and Their*

Applications in Practice, John Wiley & Sons, New York, 1992.

[13] J. D. Glover and F. C. Schweppe, *Control of linear dynamic systems with set constrained disturbances*, IEEE Trans. Automat. Control, 16 (1971), pp. 411–423.

[14] S. C. Graves, A. H. G. Rinnooy Kan, and P. H. Zipkin, eds., *Handbooks in Operations Research and Management Science, Vol.* 4*: Logistics of Production and Inventory*, North–Holland, Amsterdam, 1993.

[15] M. Grötschel, L. Lovasz, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin, New York, 1988.

[16] H. Groenevelt, *The just–in–time system*, in Handbooks in Operations Research and Management Science, Vol. 4: Logistics of Production and Inventory, S. C. Graves, A. H. G. Rinnooy Kan, and P. H. Zipkin, eds., North–Holland, Amsterdam, 1993, pp. 629–670.

[17] P. O. Gutman and M. Cwikel (1986), *Admissible sets and feedback control for discrete-time linear dynamical systems with bounded controls and states*, IEEE Trans. Automat. Control, 31 (1986), pp. 373–376; part 2 in IEEE Trans. Automat. Control, 31 (1986), pp. 457–459.

[18] R. W. Hall, *Zero Inventories*, Dow–Jones Irwin, Homewood, IL, 1983.

[19] A. Iftar and E. J. Davison, *Decentralized robust control for dynamic routing of large scale networks*, in Proc. American Control Conference, San Diego, 1990, pp. 441–446.

[20] A. Iftar and E. J. Davison, *A decentralized discrete-time controller for dynamic routing*, in Proc. 29th Conference on Decision and Control, Honolulu, 1990, pp. 1362–1366.

[21] S. S. Keerthi and E. G. Gilbert, *Computation of minimum-time feedback control laws for discrete-time systems with state-control constraints*, IEEE Trans. Automat. Control, 32 (1987), pp. 432–435.

[22] L. Lovasz, *Submodular functions and optimization*, in Mathematical Programming: The State of Art, Springer-Verlag, Berlin, New York, 1982, pp. 253–257.

[23] T. L. Magnanti and R. T. Wong, *Network design and transportation planning: Models and algorithms*, Transportation Sci., 18 (1984), pp. 1–55.

[24] R. O. Mason and E. G. Flamholtz, *Human resource management*, in Handbook of Operations Research: Models and Applications, Vol. 2, J. J. Moder and S. E. Elmaghraby, eds., Van Nostrand Reinhold, New York, 1978, pp. 92–126.

[25] S. T. McCormick, *Submodular containment is hard, even for networks*, UBC Faculty of Commerce Working Paper 94-MSC-010, OR Letters, 19 (1996), pp. 95–99.

[26] K. S. Moorthy, *Competitive marketing strategies: Game–theoretic models*, in Handbooks in Operations Research and Management Science, Vol. 5: Marketing, J. Eliasberg and G. L. Lilien, eds., North–Holland, Amsterdam, 1993, pp. 143–190.

[27] R. J. T. Morris and R. F. Brown, *Extension of validity of the GRG method in optimal control calculation*, IEEE Trans. Automat. Control, 21 (1976), pp. 420–422.

[28] F. H. Moss and A. Segall, *An optimal control approach to dynamic routing in networks*, IEEE Trans. Automat. Control, 27 (1982), pp. 329–339.

[29] J. A. Muckstadt and R. O. Roundy, *Analysis of multistage production systems*, in Handbooks in Operations Research and Management Science, Vol. 4: Logistics of Production and Inventory, S. C. Graves, A. H. G. Rinnooy Kan, and P. H. Zipkin, eds., North–Holland, Amsterdam, 1993, pp. 59–131.

[30] A. Prèkopa and E. Boros, *On the existence of feasible flow in a stochastic transportation networks*, Oper. Res., 39 (1991), pp. 119–129.

[31] J. B. Orlin, *Minimum convex cost dynamic network flow*, Math. Oper. Res., 9 (1984), pp. 190–207.

[32] R. T. Rockafellar, *Network Flows and Monotropic Optimization*, John Wiley & Sons, New York, 1984.

[33] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[34] E. A. Silver and R. Peterson, *Decision Systems for Inventory Management and Production Planning*, Wiley, New York, 1985.

[35] S. W. Wallace and R. J. B. Wets, *The facets of the polyhedral set determined by the Gale–Hoffman inequalities*, Math. Programming, 62 (1993), pp. 215–222.

SIAM J. OPTIM.
Vol. 7, No. 3, pp. 579–594, August 1997

OPTIMALITY CONDITIONS FOR THE MINIMIZATION OF A QUADRATIC WITH TWO QUADRATIC CONSTRAINTS*

JI-MING PENG[†] AND YA-XIANG YUAN[†]

Abstract. The trust region method has been proven to be very successful in both unconstrained and constrained optimization. It requires the global minimum of a general quadratic function subject to ellipsoid constraints. In this paper, we generalize the trust region subproblem by allowing two general quadratic constraints. Conditions and properties of its solution are discussed.

Key words. trust region method, global minimizer, constrained optimization, subproblem, quadratic constraint

AMS subject classifications. 65, 90

PII. S1052623494261520

1. Introduction.

Many trust region algorithms for constrained optimization require solving subproblems of the following form:

$$\min\{q(x) :\| Dx \|_2\leq \delta, \| A^Tx + c \|_2\leq \xi, x \in \Re^n\}, \tag{1.1}$$

where $q : \Re^n \to R$ is a quadratic model of the objective function in a neighborhood of the current iterate, D is a positive definite scaling matrix, $c \in \Re^m$ is a vector whose elements are the values of the constraints, $A^T \in \Re^{m\times n}$ is the Jacobian matrix of the constraints computed at the current iterate, and the numbers δ and ξ are determined by the trust region method (for example, see [1] and [11]). For unconstrained optimization problems, the trust region subproblem is to minimize a quadratic function in an ellipsoid, namely

$$\min\{q(x) :\| Dx \|_2\leq \delta,\ x \in \Re^n\}. \tag{1.2}$$

Many results for problem (1.2) have been obtained, including Gay [4], Moré and Sorensen [10], Martínez [7], and Sorensen [12]. Most authors study the global minimizer of (1.2), but Martínez [7] also studies local minimizers of (1.2). One motivation for studying nonglobal local minimizers is that a global minimizer of (1.1) at which the constraint $||A^Tx + c|| \leq \xi$ is inactive must be a local minimizer of (1.2) (see [7]).

Problem (1.1) has also been studied by many researchers; for example, see Celis, Dennis, and Tapia [1], Crouzeix, Martínez, Legaz, and Seeger [2], Heinkenschloss [5], Yuan [15], [16], Zhang [17], and the references therein. It is interesting to note that unlike the case of one constraint, for the two constraint case it is possible that the Hessian of the Lagrangian has negative eigenvalues, even when only one constraint is active at the global minimizer. For details, see Yuan [15].

Several extensions of problem (1.1) are of interest. The simplest type is to consider the problem

$$\min\{q(x) : c_1(x) \leq 0, c_2(x) \leq 0, x \in \Re^n\}, \tag{1.3}$$

* Received by the editors January 4, 1994; accepted for publication (in revised form) March 5, 1996. This work was supported by the state key project "Scientific and Engineering Computing."
http://www.siam.org/journals/siopt/7-3/26152.html

[†] State Key Laboratory of Scientific and Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, POB 2719, Beijing 100080, China (pjm@lsec.cc.ac.cn, yyx@lsec.cc.ac.cn).

where $q(x)$, $c_1(x)$, and $c_2(x)$ are quadratic functions. Several special cases of (1.3) have been discussed in the literature. For example, Heinkenschloss [5] considered the case that $q(x)$, $c_1(x)$, $c_2(x)$ are all convex quadratics; Martínez and Santos [8] considered (1.3) with a general quadratic $q(x)$ and c_1, c_2 strictly convex quadratics. More can also be found in [15], [16], [17]. In this paper we consider the case where $q(x)$, $c_1(x)$, and $c_2(x)$ all are general quadratic functions. Our paper is motivated by a recent work of Moré [9], in which he studied the problem of minimizing a quadratic function subject to one general quadratic constraint which has the form

$$\min\{q(x) : c(x) \leq 0, x \in \Re^n\}, \tag{1.4}$$

where $q(x), c(x)$ are general quadratic functions. Stern and Wolkowicz [13] also studied the above problem with a two-sided (upper and lower bound) quadratic constraint; they also discussed the characterizations of optimality and gave some conditions for the existence of solutions.

Our paper can be viewed as a generalization of Yuan [15] from convex constraints to general constraints. Our results are also related to Martínez [7], as his analysis on nonglobal local minimizers of problem (1.2) are applicable to problem (1.1) when the constraint $||A^T x + c|| \leq \xi$ is inactive at the solution. However, our results are more general because we study general quadratic functions $c_1(x)$ and $c_2(x)$ while Martínez [7] and Yuan [15] require convex constraints.

Throughout this paper, we assume that the object function $q(x)$ and the constrained functions $c_1(x)$ and $c_2(x)$ are all quadratic:

$$q(x) = \gamma + w^T x + \frac{1}{2} x^T Q x, \tag{1.5}$$

$$c_1(x) = \gamma_1 + w_1^T x + \frac{1}{2} x^T C_1 x, \tag{1.6}$$

$$c_2(x) = \gamma_2 + w_2^T x + \frac{1}{2} x^T C_2 x, \tag{1.7}$$

where $\gamma, \gamma_1, \gamma_2 \in \Re$, $w, w_1, w_2 \in \Re^n$, and Q, C_1, C_2 are symmetric matrices in $\Re^{n\times n}$. We also use the following notations:

$$E_1 = \{x : x \in \Re^n,\ c_1(x) \leq 0\}, \tag{1.8}$$

$$E_2 = \{x : x \in \Re^n,\ c_2(x) \leq 0\}, \tag{1.9}$$

$$E = E1 \cap E_2. \tag{1.10}$$

Some of our results depend on the following conditions:

$$\inf_{x\in E_1} \{c_2(x)\} < 0 < \sup_{x\in E_1} \{c_2(x)\}, \tag{1.11}$$

$$\inf_{x\in E_2} \{c_1(x)\} < 0 < \sup_{x\in E_2} \{c_1(x)\}, \tag{1.12}$$

which can be viewed as a generalization of a condition given by Moré for one constraint problem (see (2.3) below). The above conditions are not restrictive for problem (1.3). In fact, if the left part of (1.11) is not true, it follows from Theorem 3.2 of [9] (given as Theorem 2.3 below) that there exists $\lambda \in \Re^+$ such that $c_2(x) + \lambda c_1(x)$ is equal to a convex quadratic, which means that $C_2 + \lambda C_1$ is positive semidefinite. Then problem

(1.3) reduces to minimizing $q(x)$ subject to $c_1(x) = 0$ in the subspace $N_{C_2+\lambda C_1}$. If the right inequality of (1.11) fails, (1.3) reduces to the one constraint problem studied by Moré [9]. Therefore, it is no loss of generality in assuming (1.11)–(1.12).

The paper is organized as follows. In the next section we state some known results which we will use repeatedly in the paper. In section 3, we give a condition that ensures the existence of a global minimizer and derive some optimality conditions for problem (1.3) when both constraints are active and the gradients are zeros at the solution. In order to further our analysis, we also explore some relations between optimality and certain definiteness of matrix pencils. In section 4, we consider optimality for problem (1.3) when $q(x)$, $c_1(x)$, and $c_2(x)$ are all general quadratics. Necessary conditions for local minimizers and global minimizers are given. It is shown that the Hessian of the Lagrangian at the solution has at most one negative eigenvalue if the Jacobian of the constraints is not zero and that for some special cases it has no negative eigenvalue. These results are not trivial, as directly applying standard second order necessary conditions can only show that the Hessian of the Lagrangian has at most two negative eigenvalues. A few remarks are also made in last section.

2. Some important results.

In this section we state some known results which will be used in our analysis.

THEOREM 2.1 (see Moré [9]). *If $A \in \Re^{n\times n}$ and $C \in \Re^{n\times n}$ are symmetric matrices, then $A + \lambda C$ is positive definite for some $\lambda \in \Re$ if and only if*

$$w \neq 0, \quad w^T C w = 0 \Longrightarrow \qquad w^T A w > 0. \tag{2.1}$$

THEOREM 2.2 (see Moré [9]). *Assume that $A \in \Re^{n\times n}$ and $C \in \Re^{n\times n}$ are symmetric matrices and that C is indefinite. Then*

$$w^T C w = 0 \Longrightarrow \qquad w^T A w \geq 0 \tag{2.2}$$

if and only if $A + \lambda C$ is positive semidefinite for some $\lambda \in \Re$.

THEOREM 2.3 (see Moré [9]). *Let $q(x)$ and $c(x)$ be quadratic functions defined on $\Re^n$. Assume that*

$$\inf_{x\in\Re^n} c(x) < 0 < \sup_{x\in\Re^n} c(x) \tag{2.3}$$

holds and that $\nabla^2 c \neq 0$. A vector x^* is a global minimizer of the problem

$$\min\{q(x) : c(x) = 0, x \in \Re^n\} \tag{2.4}$$

if and only if $c(x^*) = 0$ and there is a multiplier $\lambda^* \in \Re$ such that the Kuhn–Tucker condition

$$\nabla q(x^*) + \lambda^* \nabla c(x^*) = 0 \tag{2.5}$$

is satisfied with

$$\nabla^2 q(x^*) + \lambda^* \nabla^2 c(x^*) \tag{2.6}$$

positive semidefinite.

THEOREM 2.4 (see Yuan [15]). *Let $C, D \in \Re^{n\times n}$ be two symmetric matrices and let A and B be two closed sets in $\Re^n$ such that $A \cup B = \Re^n$. If we have*

$$x^T C x \geq 0, x \in A, \quad x^T D x \geq 0, x \in B, \tag{2.7}$$

then there exists a $t \in [0, 1]$ such that the matrix $tC + (1-t)D$ is positive semidefinite.

3. Optimality and matrices pencils. In this section, we first give a condition which implies that the global minimum of problem (1.3) can be attained. Then, we study a special case of (1.3) when both constraints are active and the Jacobian of the constraints are zero at the solution.

Denote

$$S_1 = \{x : x \in \Re^n, x^T C_1 x \leq 0\}, \tag{3.1}$$

$$S_2 = \{x : x \in \Re^n, x^T C_2 x \leq 0\}, \tag{3.2}$$

$$S = S_1 \cap S_2. \tag{3.3}$$

Lemma 3.1. *Assume that the feasible set* (1.10) *is nonempty*; *if*

$$x \neq 0, x \in S \Longrightarrow x^T Q x > 0, \tag{3.4}$$

then (1.3) *has a global minimizer.*

Proof. If problem (1.3) does not have a global minimizer, then there exists $\{x_k, k = 1, 2, ...\}$ such that $\lim_{k\to\infty} ||x_k|| \to \infty$ and

$$q(x_k) \leq q(x_1),\ c_1(x_k) \leq 0;\ c_2(x_k) \leq 0. \tag{3.5}$$

Let $d_k = \frac{x_k}{||x_k||}$. Without loss of generality (w.l.o.g.), we assume that $\lim_{k\to\infty} d_k = d_0$. It then follows from (1.5)–(1.7) and (3.5) that

$${d_0}^T Q d_0 \leq 0,\ {d_0}^T C_1 d_0 \leq 0,\ {d_0}^T C_2 d_0 \leq 0,$$

which contradicts (3.4). Thus, the lemma is true. □

It should be noted that (3.4) is not a necessary condition for problem (1.3) to have a global minimizer. For example, let $x = (x_1, x_2)^T \in \Re^2$; we define $q(x) = x_1^2 - x_2^2$, $c_1(x) = x_2$, and $c_2(x) = \frac{1}{2}x_1 - x_2$. This problem has a global minimizer $(0,0)^T$. Obviously $S = \Re^2$, but for $\bar{x} = (0,1)^T \in S$, $\bar{x}^T Q \bar{x} < 0$ holds.

Lemma 3.1 indicates that there are connections between Q, C_1, C_2, and the global minimizer of (1.3). Moré [9] and Stern and Wolkowicz [13] have derived relations between matrix pencils and the optimization problem with one general quadratic constraint. In the rest of this section, we will discuss the relation between matrix pencils and a special case of problem (1.3) when both constraints are active and the Jacobian of the constraints are zero at the solution.

Assume that x^* is a local minimizer of problem (1.3) at which $c_1(x^*) = c_2(x^*) = 0$ and $\nabla c_1(x^*) = \nabla c_2(x^*) = 0$. It is easy to see that the null vector 0 is a local minimizer of the following problem:

$$\min\{q(x^* + x) : x \in S\}, \tag{3.6}$$

where S is defined by (3.3).

For any A which is an $n \times n$ symmetric matrix $A \in \Re^{n\times n}$, we define $N_A = \{x : x^T A x = 0\}$. Denote $F = N_{C_1} \cap N_{C_2}$. The following result is the first conclusion of the main theorem of Uhlig [14].

Theorem 3.2. *Assume that* $A, B \in \Re^{n\times n}$ *and* $n \geq 3$; *then, there exist* α, $\beta \in \Re$ *satisfying* $\alpha^2 + \beta^2 > 0$ *such that* $\alpha A + \beta B$ *is positive definite if and only if* $N_A \cap N_B = \{0\}$.

In what follows we state a result about the pair of matrices (C_1, C_2).

Lemma 3.3. *$\alpha C_1 + \beta C_2$ is indefinite for any $\alpha,\ \beta \in \Re$ satisfying $\alpha^2 + \beta^2 > 0$ if and only if*

$$\inf_{x \in N_{C_1}} \{x^T C_2 x\} < 0 < \sup_{x \in N_{C_1}} \{x^T C_2 x\}, \tag{3.7}$$

$$\inf_{x \in N_{C_2}} \{x^T C_1 x\} < 0 < \sup_{x \in N_{C_2}} \{x^T C_1 x\}. \tag{3.8}$$

Proof. First suppose that (3.7)–(3.8) hold. For any $\alpha,\ \beta \in \Re$ satisfying $\alpha^2 + \beta^2 > 0$, w.l.o.g. assume $\alpha > 0$. It follows from (3.8) that

$$\inf_{x \in N_{C_2}} x^T(\alpha C_1 + \beta C_2)x \ < 0 < \sup_{x \in N_{C_2}} x^T(\alpha C_1 + \beta C_2)x, \tag{3.9}$$

which shows that $\alpha C_1 + \beta C_2$ is indefinite.

Now we assume that $\alpha C_1 + \beta C_2$ is indefinite for any $\alpha,\ \beta \in \Re$ satisfying $\alpha^2 + \beta^2 > 0$. If (3.7)–(3.8) is not true, there is no loss of generality in assuming that

$$\inf_{x \in N_{C_1}} x^T C_2 x = 0. \tag{3.10}$$

Our assumption that $\alpha C_1 + \beta C_2$ is indefinite for any $\alpha,\ \beta \in \Re^n$ satisfying $\alpha^2 + \beta^2 > 0$ implies that C_1 is indefinite; thus, it follows from (3.10) and Theorem 2.2 that there exists $\lambda \in \Re$ such that $C_2 + \lambda C_1$ is positive semidefinite, which is a contradiction. This completes our proof. □

For the special problem (3.6), conditions (1.11) and (1.12) are equivalent to

$$\inf_{x \in S_1} \{x^T C_2 x\} < 0 < \sup_{x \in S_1} \{x^T C_2 x\}, \tag{3.11}$$

$$\inf_{x \in S_2} \{x^T C_1 x\} < 0 < \sup_{x \in S_2} \{x^T C_1 x\}. \tag{3.12}$$

We see that our conditions (3.11)–(3.12) are strictly weaker than (3.7)–(3.8). If

$$C_1 = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad C_2 = \begin{pmatrix} 1 & 0 \\ 0 & -4 \end{pmatrix}, \tag{3.13}$$

then (3.11)–(3.12) are satisfied, but (3.7)–(3.8) fail.

One direct consequence of (3.11)–(3.12) is the following lemma.

Lemma 3.4. *If* (3.11)–(3.12) *hold, then*

$$\text{span}\,(S_1\ \cap\ S_2) = \Re^n. \tag{3.14}$$

Proof. By (3.11)–(3.12), both C_1 and C_2 are indefinite. If $\max(x^T C_1 x, x^T C_2 x) \geq 0$ for every $x \in \Re^n$, it follows from Theorem 2.4 that there exists $\lambda \in (0, 1)$ such that $C_1 + \lambda(C_1 - C_2)$ is positive semidefinite, which implies

$$x^T C_2 x \geq 0 \quad \text{whenever } x^T C_1 x \leq 0 \tag{3.15}$$

and

$$x^T C_1 x \geq 0 \quad \text{whenever } x^T C_2 x \leq 0. \tag{3.16}$$

Inequalities (3.15)–(3.16) contradict (3.11)–(3.12). Thus, there exists $\bar{x} \in \Re^n$ such that

$$\bar{x}^T C_1 \bar{x} < 0, \quad \bar{x}^T C_2 \bar{x} < 0. \tag{3.17}$$

Define

$$\delta(\bar{x},\ \varepsilon) = \{x : \| x - \bar{x} \| \leq \varepsilon\ \}. \tag{3.18}$$

It follows from (3.17) and the continuity of quadratic functions that for sufficiently small $\epsilon > 0$,

$$x \in S_1 \cap S_2 \quad \forall\ x \in\ \delta(\bar{x}, \epsilon). \tag{3.19}$$

The above relation implies (3.14). This proves our lemma. □

The above lemma implies the following result.

LEMMA 3.5. *If* (3.11)–(3.12) *hold and if* $y^* = 0$ *is a local minimizer of* (3.6)*, then* $\nabla q(x^*) = 0$.

Proof. Because $y^* = 0$ is a local minimizer of (3.6), it follows that

$$x^T \nabla q(x^*) \geq 0 \quad \forall\ x \in S_1 \cap S_2. \tag{3.20}$$

Due to $S_1 \cap S_2 = -(S_1 \cap S_2)$, (3.20) implies that

$$x^T \nabla q(x^*) = 0 \quad \forall x \in S_1 \cap S_2. \tag{3.21}$$

It follows from (3.21) and (3.14) that $\nabla q(x^*) = 0$. □

Motivated by the results of Morè (see Theorems 2.1–2.3), one may guess that if $x^* = 0$ is a global minimizer of problem (3.6) and conditions (3.11)–(3.12) hold, then there may exist $\alpha, \beta \in \Re$ such that $Q + \alpha C_1 + \beta C_2$ is positive definite or semidefinite. However, our next example shows that even when conditions (3.7)–(3.8) are true and $x^* = 0$ is a global minimizer of problem (3.6), $Q + \alpha C_1 + \beta C_2$ may be indefinite for any $\alpha, \beta \in \Re$.

Example 1.

$$\min\{-(x^2 + y^2)/3 + y^2 - x^2 - 2xy :\ x^2 - y^2 \leq 0, 2xy \leq 0\}. \tag{3.22}$$

For this problem, we have

$$Q = -\frac{1}{3}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{3.23}$$

$$C_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \qquad C_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{3.24}$$

It is easy to show that $\alpha C_1 + \beta C_2$ is indefinite for any $\alpha,\ \beta \in \Re$ satisfying $\alpha^2 + \beta^2 > 0$. Thus, conditions (3.7)–(3.8) hold. One can also easily verify that $x^* = 0$ is a unique solution of problem (3.22). However, for any $\alpha, \beta \in \Re$, it holds that $Q + \alpha C_1 + \beta C_2 = -\frac{1}{3}I + (\alpha - 1)C_1 + (\beta - 1)C_2$, which implies that $Q + \alpha C_1 + \beta C_2$ cannot be positive definite or semidefinite.

To study the optimal conditions at a global minimizer of (3.6), we also need the following result due to Hestenes and Mcshane [6].

LEMMA 3.6. *Let* $C_1, C_2 \in \Re^{n \times n}$ *be symmetric matrices satisfying* (3.7)–(3.8)*. Let* $m(\alpha, \beta)$ *be the least eigenvalue of the matrix* $Q + \alpha C_1 + \beta C_2$*. Then, there exists* $(\alpha_0, \beta_0) \in \Re^2$ *which maximizes the function* $m(\alpha, \beta)$.

Our next result is a small modification of Lemma B in [6]. For completeness, we rewrite it and give a detailed proof.

LEMMA 3.7. *Assume the matrices C_1 and C_2 satisfying* (3.7)–(3.8) *and* $(\alpha_0, \beta_0) \in \Re^2$ *maximize the function* $m(\alpha, \beta)$. *Set* $m_0 = m(\alpha_0, \beta_0)$, *with* X *as the subspace spanned by all the eigenvectors of the matrix* $Q + \alpha_0 C_1 + \beta_0 C_2$ *related to* m_0. *Then for any linear space L which contains* X, *there is no* $\alpha C_1 + \beta C_2$ *positive definite on* L.

Proof. Assume there exists $\bar{C} = \alpha C_1 + \beta C_2$ positive definite on L. Let K be the unit sphere $x^T x = 1$, and L_1 is the set of points in L on K. Choose $b > 0$ such that $x^T \bar{C} x > b$ on L_1, and let $\bar{N}$ be a neighborhood of L_1 related to K on which $x^T \bar{C} x > b$, m_1 is the minimum of $x^T(Q + \alpha_0 C_1 + \beta_0 C_2)x$ on the closed set $K - \bar{N}$; then, $m_1 > m_0$. It follows that for a sufficiently small positive constant t one will have

$$x^T(Q + \alpha_0 C_1 + \beta_0 C_2 + t\bar{C})x > m_0 \tag{3.25}$$

on $K - \bar{N}$. But,

$$x^T(Q + \alpha_0 C_1 + \beta_0 C_2 + t\bar{C})x > m_0 + tb \tag{3.26}$$

on $\bar{N}$. Thus, it holds that $m(\alpha_0 + t\alpha, \beta_0 + t\beta) > m(\alpha_0, \beta_0)$, which contradicts the choice of (α_0, β_0). This proves the lemma. □

Now we can give one of our main results in this section.

THEOREM 3.8. *If* (3.7)–(3.8) *hold and if* $y^* = 0$ *is a local minimizer of problem* (3.6), *then* $\nabla q(x^*) = 0$ *and there exist* $\alpha, \beta \in \Re$ *such that* $Q + \alpha C_1 + \beta C_2$ *has at most two negative eigenvalues.*

Proof. It follows from Lemma 3.5 that $\nabla q(x^*) = 0$. Because $\nabla q(x^*) = 0$ and the optimality of $y^* = 0$, we have that $x^T Q x \geq 0$ for all $x \in S$.

If the theorem is not true, assume that for any $\alpha, \beta \in \Re$, $Q + \alpha C_1 + \beta C_2$ has three or more negative eigenvalues. Let (α_0, β_0) maximize the function $m(\alpha, \beta)$, and let L be the subspace spanned by the eigenvectors of the matrix $Q + \alpha_0 C_1 + \beta_0 C_2$ corresponding to its negative eigenvalues. For example, $L = \text{span}\{x_1, x_2, \ldots, x_l : (Q + \alpha_0 C_1 + \beta_0 C_2)x_i = a_i x_i, a_i < 0, \|x_i\|_2 = 1\}$ and $l = \dim(L) \geq 3$. It follows that in L, we have

$$Q + \alpha_0 C_1 + \beta_0 C_2 = \sum_{i=1}^{l} a_i x_i x_i^T. \tag{3.27}$$

If there exists $x_0 \in F \neq 0$ in L, w.l.o.g. we assume that $\|x_0\|_2 = 1$. Then, by the definition of F, we get

$$x_0^T(Q + \alpha_0 C_1 + \beta_0 C_2)x_0 \geq 0, \tag{3.28}$$

which contradicts the definition of L. It follows that $F \cap L = \{0\}$. However, since $l \geq 3$, it follows from Theorem 3.2 that there exist $\alpha, \beta \in \Re$ such that $\alpha C_1 + \beta C_2$ is positive definite on L, which contradicts Lemma 3.7. □

In fact, under the conditions of Theorem 3.8, let α_0 and β_0 as defined in Lemma 3.6 and $m_0 = m(\alpha, \beta)$ denote L_1 as the subspace spanned by the eigenvectors of $Q + \alpha_0 C_1 + \beta_0 C_2$ related to m_0. If $m_0 < 0$, then by Theorem 3.8 we have $\dim(L_1) < 3$. By Lemma 3.7, $\dim(L_1) \neq 1$; thus, it must hold that $\dim(L_1) = 2$. This can also be verified by our Example 1, where $Q + C_1 + C_2 = -\frac{1}{3}I$, $m(1, 1) = -\frac{1}{3}$. But for any $(\alpha, \beta) \in \Re^2$, $Q + \alpha C_1 + \beta C_2 = -\frac{1}{3}I + (\alpha - 1)C_1 + (\beta - 1)C_2$. If $(\alpha, \beta) \neq (1, 1)$, then $(\alpha - 1)C_1 + (\beta - 1)C_2$ is indefinite, which implies that the least eigenvalue of $Q + \alpha C_1 + \beta C_2$ is less than $-\frac{1}{3}$. Thus, for Example 1, it holds that $m_0 = m(1, 1) = -\frac{1}{3}$.

Now we only consider the case $m_0 < 0$ under the conditions of Theorem 3.8. Let L_1 be the subspace defined by $L_1 = \{x \in \Re^n : \ (Q + \alpha_0 C_1 + \beta_0 C_2)x = m_0 x\}$. It easy to see that in L_1, $Qx = (-\alpha_0 C_1 - \beta_0 C_2 + m_0 I)x$. Thus, $x^* = 0$ is a global minimizer of the following problem:

$$\min\{x^T(-\alpha_0 C_1 - \beta_0 C_2 + m_0 I)x : \ x^T C_1 x \le 0, x^T C_2 x \le 0, \ x \in L_1\}. \tag{3.29}$$

Since $m_0 < 0$, $x^T C_1 x$ and $x^T C_2 x$ vanish simultaneously only at the point 0. By Lemma 3.7, there is no $\alpha, \beta \in \Re$ such that $\alpha C_1 + \beta C_2$ is positive definite. Thus, in L_1 we have

$$x^T C_1 x \ne 0 \quad \forall x^T C_2 x = 0, x \ne 0 \tag{3.30}$$

and

$$x^T C_2 x \ne 0 \quad \forall x^T C_1 x = 0, x \ne 0. \tag{3.31}$$

If $x^T C_1 x > 0$ for all $x^T C_2 x = 0, x \ne 0 \in L_1$, then it follows from Theorem 2.1 that there exists $\lambda \in \Re$ such that $C_1 + \lambda C_2$ is positive definite, which is a contradiction. Therefore, there exists $\bar{x} \in L_1$ such that

$$\bar{x}^T C_2 \bar{x} = 0, \ \bar{x}^T C_1 \bar{x} \le 0. \tag{3.32}$$

The fact that $\bar{x}^T(-\alpha_0 C_1 + m_0 I)\bar{x} \ge 0$ implies that $\alpha_0 > 0$. Similarly, one can show that $\beta_0 > 0$.

If conditions (3.11)–(3.12) are true and (3.7)–(3.8) fail, then we have the following result.

THEOREM 3.9. *If* (3.11)–(3.12) *hold and* (3.7)–(3.8) *fail and if* $y^* = 0$ *is a local minimizer of problem* (3.6), *then* $\nabla q(x^*) = 0$ *and there exist* $\lambda_1, \lambda_2 \in \Re$ *such that* $Q + \lambda_1 C_1 + \lambda_2 C_2$ *is positive semidefinite.*

Proof. It follows from Lemma 3.5 that $\nabla q(x^*) = 0$. Since conditions (3.7)–(3.8) are not satisfied, it follows from Lemma 3.3 that there exist $\alpha, \beta \in \Re$ such that $\alpha^2 + \beta^2 \ne 0$ and that $\alpha C_1 + \beta C_2$ is positive semidefinite. Without loss of generality, we assume that $\alpha \ne 0$. Define $\lambda = \beta/\alpha$. First we assume that $\alpha > 0$, which implies that $C_1 + \lambda C_2$ is positive semidefinite. This leads to the following two cases: if $\lambda > 0$ then $x^T C_1 x \le 0 \Longrightarrow x^T C_2 x \ge 0$, which contradicts (3.11)–(3.12); if $\lambda < 0$ then $x^T C_1 x \le 0 \Longrightarrow x^T C_2 x \le 0$, which contradicts (3.11).

Now we assume that $\alpha < 0$, which implies that $C_1 + \lambda C_2$ is negative semidefinite. If $\lambda > 0$ then $x^T C_1 x = 0 \Longrightarrow x^T C_2 x \le 0$; $y^* = 0$ is a local minimizer of problem $\min\{x^T Q x : \ x^T C_1 x = 0, \ x \in \Re^n\}$. Thus, our theorem follows from Theorem 2.2. If $\lambda \le 0$ then $x^T C_2 x \le 0 \Longrightarrow x^T C_1 x \le 0$, which contradicts (3.12). Therefore, the theorem is proved. □

If C_i is positive definite then we can choose the corresponding Lagrange multiplier λ_i large enough so that $Q + \lambda_i C_i$ is positive definite. But in the case that C_1 is positive semidefinite and $N_{C_1} \ne \emptyset$, then even (3.11) holds, and there may be no $\lambda_1, \lambda_2 \in \Re$ such that $Q + \lambda_1 C_1 + \lambda_2 C_2$ is positive semidefinite. This can be verified by the following example.

$$Q = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \qquad C_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad C_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

C_1 is semidefinite and (3.11) holds; $y^* = 0$ is a global minimizer of

$$\min\{x^T Qx : x \in\ S_1 \cap S_2\}, \tag{3.33}$$

but for any $\lambda_1,\ \lambda_2 \in \Re$, $Q + \lambda_1 C_1 + \lambda_2 C_2$ is not positive semidefinite. For the case where $C_1,\ C_2$ are indefinite, if (3.11)–(3.12) do not hold, Theorem 3.9 may also fail. For example,

$$Q = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -3 \end{pmatrix}, \qquad C_1 = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad C_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Now we give a lemma which will be used in the next section.

LEMMA 3.10. *If* $C\ \in \Re^{n\times n}$ *is a symmetric indefinite matrix, then* $\operatorname{span}(N_C)\ =\ \Re^n$.

Proof. Without loss of generality, we assume that

$$C = \operatorname{diag}\left(\alpha_1, \ldots, \alpha_I; -\beta_1, \ldots, -\beta_J; 0, \ldots, 0\right), \tag{3.34}$$

where $\alpha_i (i = 1, \ldots, I)$ and $\beta_j (j = 1, \ldots, J)$ are positive numbers and $I \geq 1,\ J \geq 1$. It is easy to see that

$$\frac{\beta_j}{\alpha_1} e_1 + e_{I+j} \in N_C \quad (j = 1, \ldots, J), \tag{3.35}$$

$$e_i - \frac{\alpha_i}{\beta_1} e_{I+1} \in N_C \quad (i = 1, \ldots, I), \tag{3.36}$$

$$e_k \in N_C \quad (k = I + J + 1, \ldots, n), \tag{3.37}$$

and these vectors are linearly independent. Thus, $\operatorname{span}(N_C)\ = \Re^n$. □

The following result is a direct consequence of Theorem 2.1.

COROLLARY 3.11. *If* $y^* = 0$ *is an isolated minimizer of the problem*

$$\min\{x^T Qx + g^T x :\ x^T Cx = 0\}, \tag{3.38}$$

then there exists $\lambda \in \Re$ *such that* $Q + \lambda C$ *is positive definite.*

Proof. For any nonzero $x \in \Re^n$ such that $x^T Cx = 0$, we have $(-x)^T C(-x) = 0$; thus, our assumption implies that

$$x^T Qx = \frac{1}{2}(x^T Qx + g^T x) + \frac{1}{2}(-x^T Q(-x) + g^T(-x)) > 0. \tag{3.39}$$

Therefore, the corollary follows from Theorem 2.1. □

Similarly, we can show the following theorem.

THEOREM 3.12. *If* $y^* = 0$ *is an isolated minimizer of problem* (3.6) *and conditions* (3.7)–(3.8) *fail, then there exist* $\lambda_1,\ \lambda_2 \in \Re$ *such that* $Q + \lambda_1 C_1 + \lambda_2 C_2$ *is positive definite.*

Proof. For any feasible point x of (3.6), the point $-x$ is also a feasible point. Thus, $y^* = 0$ is also an isolated local minimizer of (3.33). Therefore, we have that $x^T Qx > 0$ for all nonzero x, which satisfies $x^T C_1 x = x^T C_2 x = 0$. Since conditions (3.7)–(3.8) are not satisfied, w.l.o.g. we assume that (3.7) is not true. First, we assume that

$$\sup_{x \in N_{C_1}} x^T C_2 x = 0. \tag{3.40}$$

Thus, $y^* = 0$ is also the unique global minimizer of

$$\min\{x^T Qx : \ x^T C_1 x = 0\}. \tag{3.41}$$

It follows from Theorem 2.1 that there exists $\lambda \in \Re$ such that $Q + \lambda C_1$ is positive definite.

To complete our proof, we assume that

$$\min_{x \in N_{C_1}} x^T C_2 x = 0. \tag{3.42}$$

We consider three different cases: C_1 is positive semidefinite, negative semidefinite, or indefinite.

If C_1 is positive semidefinite, then the feasible region $\{x^T C_1 x \leq 0\}$ is the subspace N_{C_1}. Thus, the null vector 0 is an isolated local minimizer of

$$\min\{x^T Qx : \ x^T C_2 x = 0, \quad x \in N_{C_1}\}, \tag{3.43}$$

which shows that there exists $\mu \in \Re$ such that $Q + \mu C_2$ is positive definite in N_{C_1}. Thus,

$$x \neq 0, \quad x^T C_1 x = 0 \quad \Longrightarrow x^T(Q + \mu C_2)x > 0. \tag{3.44}$$

Hence, there exists $\lambda \in \Re$ such that $Q + \mu C_2 + \lambda C_1$ is positive definite.

If C_1 is negative semidefinite, we have that

$$x \neq 0, \quad x^T C_2 x = 0 \Longrightarrow x^T Qx > 0. \tag{3.45}$$

Therefore, it follows from Theorem 2.1 that there exists $\mu \in \Re$ such that $Q + \mu C_2$ is positive definite.

Finally, we consider if C_1 is indefinite. It follows from (3.42) and Theorem 2.2 that there exists $\alpha \in \Re$ such that $C_2 + \alpha C_1$ is positive semidefinite. Because $y^* = 0$ is an isolated local minimizer of (3.6), we have for all $x \in N_{C_2 + \alpha C_1}$,

$$x \neq 0, \quad x^T C_1 x = 0 \quad \Longrightarrow \quad x^T Qx > 0. \tag{3.46}$$

Thus, it follows from Theorem 2.1 that there exists $\beta \in \Re$ such that $Q + \beta C_1$ is positive definite in the subspace $N_{C_2 + \alpha C_1}$. Using Theorem 2.1 again, we can show that there exists $\gamma \in \Re$ such that $Q + \beta C_1 + \gamma(C_2 + \alpha C_1)$ is positive definite. Hence, the theorem is true. □

4. Optimal conditions.

In this section we mainly give necessary conditions for minimizers of problem (1.3). Necessary conditions for optimality are already given in the previous section, when both constraints are active and gradients of the constraints are zeros at the solution.

First, the following result is obvious.

THEOREM 4.1. *Assume that* $c_1(x^*) < 0$ *and* $c_2(x^*) < 0$. x^* *is a local minimizer of problem* (1.3) *if and only if* $\nabla q(x^*) = 0$ *and* Q *is positive semidefinite.*

Hence, in the following we assume that at least one of the constraints is active at a minimizer.

If only one constraint is active at the global minimizer x^*, w.l.o.g. we assume that

$$c_1(x^*) = 0, \ c_2(x^*) < 0. \tag{4.1}$$

THEOREM 4.2. *Assume that* (4.1) *holds and that* x^* *is a local minimizer of problem* (1.3)*. If* $\nabla c_1(x^*) \neq 0$*, then there exists* $\lambda_1 \in \Re^+$ *such that*

$$\nabla q(x^*) + \lambda_1 \ \nabla c_1(x^*) = 0 \tag{4.2}$$

holds and $Q + \lambda_1 C_1$ *has at most one negative eigenvalue. If* $\nabla c_1(x^*) = 0$*, then* $\nabla q(x^*) = 0$ *and there exists* $\lambda_1 \in \Re^+$ *such that* $Q + \lambda_1 C_1$ *is positive semidefinite.*

Proof. If $\nabla c_1(x^*) \neq 0$, (4.2) follows from the Kuhn–Tucker theory. It follows from the second order necessary condition (see, for example, Fletcher [3]) that $Q + \lambda_1 C_1$ is positive semidefinite in the subspace

$$W = \{d : \nabla c_1(x^*)^T d = 0, \ d \in \Re^n \ \}. \tag{4.3}$$

Therefore, $Q + \lambda_1 C_1$ has at most at most one negative eigenvalue.

Now we assume that $\nabla c_1(x^*) = 0$, since $y^* = 0$ is a local minimizer of

$$\min_{d \in S_1} q(x^* + d). \tag{4.4}$$

Thus, it follows that

$$d^T \nabla q(x^*) = 0 \quad \forall d \in N_{C_1}.$$

The fact that $\nabla c_1(x^*) = 0$ and (1.12) imply that C_1 is indefinite. Lemma 3.10 shows that $\text{span}(N_{C_1}) = \Re^n$, which gives $d^T \nabla q(x^*) = 0$ for all $d \in \Re^n$. Therefore, $\nabla q(x^*) = 0$. This shows that (4.2) holds for all $\lambda_1 \in \Re$. $\nabla q(x^*) = 0$, Theorem 2.4, and the fact that $y^* = 0$ solves (4.4) imply that there exists $\lambda_1 \in \Re^+$ such that $Q + \lambda_1 C_1$ is positive semidefinite. □

Using the second order necessary conditions, it can be proven that the Hessian of the Lagrangian has at most two negative eigenvalues if both constraints $c_1(x) \leq 0$ and $c_2(x) \leq 0$ are active at the solution.

For convex problems, Yuan [15] shows that the Hessian of the Lagrangian has at most only one negative eigenvalue at a global minimizer. In the following, Yuan's results are extended to general cases. For the rest of this section we assume that x^* is a global minimizer of problem (1.3) and both constraints are active at x^*, which means that $c_1(x^*) = c_2(x^*) = 0$. First, we consider the case when $\nabla c_1(x^*)$ and $\nabla c_2(x^*)$ are linearly independent.

THEOREM 4.3. *If* x^* *is a global minimizer of problem* (1.3) *and if* $\nabla c_1(x^*)$ *and* $\nabla c_2(x^*)$ *are linearly independent, then there exist* $\lambda_1, \lambda_2 \in \Re^+$ *such that*

$$\nabla q(x^*) + \lambda_1 \nabla c_1(x^*) + \lambda_2 \nabla c_2(x^*) = 0 \tag{4.5}$$

and $Q + \lambda_1 C_1 + \lambda_2 C_2$ *has at least* $n - 1$ *nonnegative eigenvalues.*

Proof. Let $\lambda_1, \lambda_2 \in \Re^+$ be the corresponding Lagrange multipliers $H = Q + \lambda_1 C_1 + \lambda_2 C_2$. Then, by the second order necessary condition we know that

$$x^T H x \geq 0 \quad \forall x \in \Re^n, \ x \perp \nabla c_1(x^*), \ x \perp \nabla c_2(x^*).$$

If H has two negative eigenvalues, similar to Yuan [15], there exist $e_1, \ e_2 \in \Re^n$ such that for all nonzero $d \in \text{span}\{e_1, \ e_2\}$, $d^T H d < 0$. Because x^* is the global minimizer, $(0, 0)^T$ is the unique solution of

$$\hat{c}_1(\alpha, \beta) = c_1(x^* + \alpha e_1 + \beta e_2) = 0, \quad \hat{c}_2(\alpha, \beta) = c_2(x^* + \alpha e_1 + \beta e_2) = 0,$$

where $(\alpha,\ \beta) \in \Re^2,\ x = x^* + \alpha e_1 + \beta e_2$. So the curves $\hat{c}_1(\alpha,\beta) = 0,\ \hat{c}_2(\alpha,\beta) = 0$ meet only at $(0,0)$. Define $\bar{F}$ as the set of all feasible points that are connected to $(0,0)$; thus, the boundary of $\bar{F}$ consists of two curves. One is $\hat{c}_1(\alpha,\beta) = 0$; the other is $\hat{c}_2(\alpha,\beta) = 0$. Let the asymptotic direction of these two curves be $\bar{d}_1,\ \bar{d}_2$; then we have

$$\bar{d_1}^T \nabla^2 \hat{c}_1 \bar{d}_1 = 0;\ \bar{d_1}^T \nabla^2 \hat{c}_2 \bar{d}_1 \le 0, \tag{4.6}$$

$$\bar{d_2}^T \nabla^2 \hat{c}_1 \bar{d}_2 \le 0;\ \bar{d_2}^T \nabla^2 \hat{c}_2 \bar{d}_2 = 0. \tag{4.7}$$

Due to the optimality of x^*, we know that

$$\bar{d_1}^T \nabla^2 \hat{q}(x^*) \bar{d}_1 \ge 0;\ \bar{d_2}^T \nabla^2 \hat{q}(x^*) \bar{d}_2 \ge 0, \tag{4.8}$$

where $\hat{q}(\alpha,\beta) = q(x^* + \alpha e_1 + \beta e_2)$. Since $d^T H d < 0$ for all nonzero $d \in \text{span}\{e_1,\ e_2\}$, it follows that

$$\bar{d_1}^T \nabla^2 \hat{c}_2 \bar{d}_1 < 0;\ \bar{d_2}^T \nabla^2 \hat{c}_1 \bar{d}_2 < 0. \tag{4.9}$$

By considering a sequence of interior points of $\bar{F}$, one can see that for any direction d between $\bar{d}_1,\ \bar{d}_2$,

$$d^T \nabla^2 \hat{c}_2 d < 0;\ d^T \nabla^2 \hat{c}_1 d < 0. \tag{4.10}$$

Otherwise, assume there exists $d \in \text{int}(K),\ d^T \nabla^2 \hat{c}_1 d = 0$; then, $(\alpha,\beta)\nabla^2 \hat{c}_1 (\alpha,\beta)^T$ has a local maximum at d. Hence, $\nabla^2 \hat{c}_1$ is negative semidefinitive, which shows that $\hat{c}_1(\alpha,\beta) = 0$ is a parabolic curve. Because the two curves have only one cross and the asymptotic direction of a parabolic curve is the same one, we know that $\bar{d}_1$ is parallel to $\bar{d}_2$, which contradicts (4.9). Hence, there exists a cone K whose boundary direction is $\bar{d}_1,\ \bar{d}_2$, and for any interior direction of K, (4.10) holds. Now, for large enough $t > 0,\ -td$ is a feasible point. Because the two curves meet only at $(0,0)$, $-td \notin \bar{F}$. Let the connected part of the feasible set which includes $-td$ be $\hat{F}$; then, $\bar{F} \cap \hat{F} = \phi$. Because $(0,0)$ is the unique cross of two curves, the boundary of $\hat{F}$ is defined by only one curve. Without loss of generality, assume that the boundary is defined by $\hat{c}_1(\alpha,\beta) = 0$. Let the asymptotic directions of $\hat{F}$ be $\hat{d}_1,\ \hat{d}_2$, and the corresponding cone is $\hat{K}$. Since (4.10) holds for all $\bar{d} \in K$, it holds that $-K \subset \hat{K}$, so $-\bar{d}_2 \in \hat{K}$. Furthermore, for all $\hat{d} \in \hat{K}$ we have

$$\hat{d}^T \nabla^2 \hat{c}_2 \hat{d} \le 0, \quad \hat{d}^T \nabla^2 \hat{c}_1 \hat{d} \le 0.$$

One can also show that there exists no $\hat{d} \in \hat{K}$ such that

$$\hat{d}^T \nabla^2 \hat{c}_2 \hat{d} = 0, \quad \hat{d}^T \nabla^2 \hat{c}_1 \hat{d} = 0$$

and that

$$\hat{d_1}^T \nabla^2 \hat{c}_1 \hat{d}_1 = 0,\ \hat{d_2}^T \nabla^2 \hat{c}_1 \hat{d}_2 = 0, \tag{4.11}$$

$$\hat{d_1}^T \nabla^2 \hat{c}_2 \hat{d}_1 < 0,\ \hat{d_2}^T \nabla^2 \hat{c}_2 \hat{d}_2 < 0. \tag{4.12}$$

Hence, $-\bar{d}_2$ is an interior direction of $\hat{K}$, which implies that $\hat{c}_2(\alpha,\beta) = 0$ is a parabolic curve. This contradicts (4.9). So, H has at most one negative eigenvalue. $\square$

The condition that the Hessian of the Lagrangian has at most one negative eigenvalue is not a sufficient condition for x^* being a local minimizer. For example, point $(1,1,0)^T$ is a Kuhn–Tucker point of the following 3-dimensional problem:

$$\min -4y + (x-1)^2 + y^2 - 10z^2 \tag{4.13}$$

$$\text{s.t. } x^2 + y^2 + z^2 \le 2, \tag{4.14}$$

$$(x-2)^2 + y^2 + z^2 \le 2. \tag{4.15}$$

It is easy to see that the Lagrange multipliers are $(1,1)$. The Hessian of the Lagrangian is

$$\begin{pmatrix} 6 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & -6 \end{pmatrix},$$

which has $2(= n-1)$ positive eigenvalues. But one can easily show that the point $(1,1,0)^T$ is not a local minimizer because the second order necessary condition is not satisfied.

In the following we deal with the case when $\nabla c_1(x^*)$ and $\nabla c_2(x^*)$ are linearly dependent. Because we have already studied the case when $\nabla c_1(x^*) = \nabla c_2(x^*) = 0$ in the previous section, we can assume that either $\nabla c_1(x^*)$ or $\nabla c_2(x^*)$ is not zero. Without loss of generality, we assume that $\nabla c_1(x^*) \neq 0$ and $\nabla c_2(x^*) = \alpha \nabla c_1(x^*)$ for the rest of the section. First we discuss the case when $\alpha > 0$.

THEOREM 4.4. *If x^* is a global minimizer of problem* (1.3) *and if there exists $\alpha > 0$ such that $\nabla c_2(x^*) = \alpha \nabla c_1(x^*) \neq 0$, then there exist $\lambda_1, \lambda_2 \in \Re^+$ such that* (4.5) *holds and the matrix $Q + \lambda_1 C_1 + \lambda_2 C_2$ is positive semidefinite.*

Proof. Since $\nabla c_2(x^*) = \alpha \nabla c_1(x^*) \neq 0$ for some $\alpha > 0$, the optimality of x^* implies that $d^T \nabla q(x^*) \ge 0$ for all d such that $d^T \nabla c_1(x^*) < 0$. Therefore, there exists $\beta \le 0$ such that $\nabla q(x^*) = \beta \nabla c_1(x^*)$. If $\beta < 0$, there is no loss of generality in assuming that $\nabla c_1(x^*) = \nabla c_2(x^*) = -\nabla q(x^*) \neq 0$. First, we show that

$$\max(x^T(Q+C_1)x, x^T(Q+C_2)x) \ge 0 \quad \forall x \in \Re^n. \tag{4.16}$$

If it fails, there exists $\hat{d} \in \Re^n$ such that

$$\hat{d}^T \nabla c_1(x^*) \neq 0, \quad \hat{d}^T(Q+C_1)\hat{d} < 0, \quad \hat{d}^T(Q+C_2)\hat{d} < 0. \tag{4.17}$$

The fact that x^* is a global minimizer of (1.3) and (4.17) imply that either $\hat{d}^T C_1 \hat{d}$ or $\hat{d}^T C_2 \hat{d}$ is not zero. Thus, we can choose $\lambda \neq 0 \in \Re$ so that

$$c_1(x^* + \lambda \hat{d}) = 0, \quad c_2(x^* + \lambda \hat{d}) \le 0 \tag{4.18}$$

or

$$c_1(x^* + \lambda \hat{d}) \le 0, \quad c_2(x^* + \lambda \hat{d}) = 0. \tag{4.19}$$

Without loss of generality, we assume that (4.18) is true; it then follows that

$$q(x^* + \lambda \hat{d}) - q(x^*) = \lambda^2 \hat{d}^T(Q+C_1)\hat{d} < 0, \tag{4.20}$$

which is a contradiction. Thus, (4.16) holds. Hence, our theorem follows from (4.16) and Theorem 2.4.

If $\beta = 0$, (4.5) holds for $\lambda_1 = \lambda_2 = 0$. The optimality of x^* implies that $d^T Qd \geq 0$ for all d such that $d^T \nabla c_1(x^*) < 0$. Since $\text{span}\{d : d^T \nabla c_1(x^*) < 0\} = \Re^n$, it follows that Q is positive semidefinite. □

In what follows we will consider the case when $\nabla c_2(x^*) = \alpha \nabla c_1(x^*)$ for some $\alpha \leq 0$.

THEOREM 4.5. *Assume that x^* is a global minimizer of problem* (1.3) *and that $c_1(x)$ and $c_2(x)$ satisfy* (1.11)–(1.12). *If $\nabla c_1(x^*) \neq 0$ and $\nabla c_2(x^*) = \alpha \nabla c_1(x^*)$ for some $\alpha \leq 0$ and $\nabla q(x^*) = \gamma \nabla c_1(x^*)$, then there exist $\lambda_1, \lambda_2 \in \Re^+$ so that* (4.5) *holds and $Q + \lambda_1 C_1 + \lambda_2 C_2$ is positive semidefinite.*

Proof. First, we consider the case when $\nabla c_2(x^*) = \alpha \nabla c_1(x^*)$ for some $\alpha < 0$. Without loss of generality, we assume that $\nabla c_1(x^*) = -\nabla c_2(x^*) \neq 0$ and $\gamma \leq 0$. Now we show that

$$\max(x^T(Q - \gamma C_1)x,\ x^T(C_1 + C_2)x) \geq 0 \quad \forall x \in \Re^n. \tag{4.21}$$

Otherwise, we can choose $\hat{d} \in \Re^n$ such that

$$\hat{d}^T \nabla c_1(x^*) < 0, \quad \hat{d}^T(Q - \gamma C_1)\hat{d} < 0, \quad \hat{d}^T(C_1 + C_2)\hat{d} < 0. \tag{4.22}$$

If $\hat{d}^T C_1 \hat{d} = 0$, then $\hat{d}^T Q \hat{d} < 0, \hat{d}^T C_2 \hat{d} < 0$. We can let $\lambda \in \Re^+$ sufficiently large so that $\lambda \hat{d}$ is feasible and $q(x^* + \lambda \hat{d}) - q(x^*) < 0$, which is a contradiction. If $\hat{d}^T C_1 \hat{d} \neq 0$, we can choose $\lambda \in \Re$ so that

$$c_1(x^* + \lambda \hat{d}) = 0, \quad c_2(x^* + \lambda \hat{d}) < 0. \tag{4.23}$$

It follows that

(4.24)
$$q(x^* + \lambda \hat{d}) - \gamma c_1(x^* + \lambda \hat{d}) - q(x^*) = q(x^* + \lambda \hat{d}) - q(x^*) = \lambda^2 \hat{d}^T (Q - \gamma C_1)\hat{d} < 0,$$

which is a contradiction. Thus, (4.21) holds. Since conditions (1.11)–(1.12) imply that $C_1 + C_2$ cannot be positive semidefinite, our theorem follows from (4.21) and Theorem 2.4.

Now we turn to the case when $\nabla c_2(x^*) = 0$. The assumptions in our theorem imply that $\nabla q(x^*) = \gamma \nabla c_1(x^*)$ for some $\gamma \leq 0$. By a similar process, we can show that

$$\max(x^T(Q - \gamma C_1)x, x^T C_2 x) \geq 0 \quad \forall x \in \Re^n, \tag{4.25}$$

which means that our theorem still holds when $\nabla c_2(x^*) = 0$. □

In the above two theorems, we have discussed optimal properties of the Hessian of a generalized Lagrangian functions when $\nabla c_1(x^*)$ and $\nabla c_2(x^*)$ are linearly dependent and $\nabla q(x^*) \in \text{span}\{\nabla c_1(x^*)\}$. But if $\nabla q(x^*) \notin \text{span}\{\nabla c_1(x^*)\}$, then the Kuhn–Tucker theory and (4.5) fail. In this case, we need to assume that x^* is a unique solution to continue our analysis.

THEOREM 4.6. *Assume that x^* is a unique global minimizer of problem* (1.3) *and that $c_1(x)$ and $c_2(x)$ satisfy* (1.11)–(1.12). *If $\nabla c_1(x^*) \neq 0$ and $\nabla c_2(x^*) = -\alpha \nabla c_1(x^*)$ for some $\alpha \geq 0$ and if $\nabla q(x^*)$ and $\nabla c_1(x^*)$ are linearly independent, then there exist $\lambda_1, \lambda_2 \in \Re$ such that $Q + \lambda_1 C_1 + \lambda_2 C_2$ has at least $n - 1$ positive eigenvalues.*

Proof. Let W be defined by (4.3). It follows from the definition of x^* that $y^* = 0$ is the unique solution of the following problem:

$$\min\{x^T Qx + x^T \nabla q(x^*) :\ x^T C_1 x \leq 0,\ x^T C_2 x \leq 0,\ x \in W\}. \tag{4.26}$$

We now show that

$$\max(x^T C_1 x, x^T C_2 x) \geq 0 \quad \forall x \in W. \tag{4.27}$$

Otherwise, there exists $x \in W$ such that

$$x^T C_1 x < 0, \; x^T C_2 x < 0. \tag{4.28}$$

Without loss of generality, we assume that $x^T \nabla q(x^*) \leq 0$. Therefore, we can choose sufficiently small $\epsilon > 0$ such that

$$\bar{x}^T \nabla q(x^*) < 0, \; \bar{x}^T C_1 \bar{x} \leq 0, \; \bar{x}^T C_2 \bar{x} \leq 0, \; \bar{x} = x - \epsilon \nabla q(x^*), \tag{4.29}$$

which contradicts the basic assumptions of the theorem. Thus, (4.27) is true. It follows from Theorem 3.12 that there exist $\lambda_1, \; \lambda_2 \in \Re$ such that $Q + \lambda_1 C_1 + \lambda_2 C_2$ is positive definite in W. This proves our theorem. □

5. Discussion. We have shown that the Hessian of the Lagrangian at the solution of problem (1.3) has at most only one negative eigenvalue if the Jacobian of the constraints is not zero. For some special cases, it is shown that the Hessian is positive semidefinite or definite. We have also derived some relations between matrix pencils and optimality. The necessary conditions given in the paper are stronger than the standard second order necessary condition, which says the Hessian is positive semidefinite in the null space of the constraint gradients. It is pointed out that the necessary conditions obtained are not sufficient conditions for optimality. It is interesting to investigate whether there are sufficient conditions that are weaker than the standard second order sufficient condition, which requires the Hessian of the Lagrangian to be positive definite at the null space of the constraint gradients. We believe that our theoretical results will help us to understand problem (1.3) better; they also will be useful for development of numerical algorithms for trust region subproblems.

Acknowledgment. We would like to thank Professor J.M. Martínez and two referees for their valuable comments.

REFERENCES

[1] M. R. Celis, J. E. Dennis, Jr., and R. A. Tapia, *A trust region strategy for nonlinear equality constrained optimization*, in Numerical Optimization, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., SIAM, Philadelphia, PA, 1985, pp. 71–82.

[2] J. P. Crouzeix, J. E. Martínez-Legaz, and A. Seeger, *An Alternative Theorem for Quadratic Forms and Extensions*, Preprint 98, Department of Math, University of Barcelona, Spain, 1991.

[3] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., John Wiley and Sons, Chichester, 1987.

[4] D. M. Gay, *Computing optimal locally constrained steps*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 186–197.

[5] M. Heinkenschloss, *On the Solution of a Two Ball Trust Region Subproblem*, Tech. rep. 92-16, Universität Trier, Trier, Germany, 1992.

[6] M. R. Hestenes and E. J. Mcshane, *A theorem on quadratic forms and its application in the calculus of variations*, Trans. Amer. Math. Soc., 40 (1940), pp. 501–512.

[7] J. M. Martínez, *Local minimizers of quadratic functions on Euclidean balls and spheres*, SIAM J. Optim., 4 (1994), pp. 159–176.

[8] J. M. Martínez and S.A. Santos, *A Trust Region Method for Minimization on Arbitrary Domains*, Tech. rep., State University of Campinas, Campinas, Brazil, 1991.

[9] J. J. Moré, *Generalization of the trust region problem*, Optim. Methods Software, 2 (1993), pp. 189–209.

[10] J. J. Moré and D. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[11] M. J. D. Powell and Y. Yuan, *A trust region algorithm for equality constrained optimization*, Math. Programming, 49 (1991), pp. 189–211.

[12] D. C. Sorensen, *Newton's method with a model trust region modification*, SIAM J. Numer. Anal., 19 (1982), pp. 409–426.

[13] R. Stern and H. Wolkowicz, *Indefinite Trust Region Subproblems and Nonsymmetric Eigenvalue Perturbations*, Tech. rep. CORR 92-38, Department of Combinatorics and Optimization, University of Waterloo, Canada, 1993.

[14] F. Uhlig, *A recurring theorem about pairs of quadratic forms and extensions: A survey*, Linear Algebra Appl., 25 (1979), pp. 219–237.

[15] Y. Yuan, *On a subproblem of trust region algorithms for constrained optimization*, Math. Programming, 47 (1990), pp. 53–63.

[16] Y. Yuan, *A dual algorithm for minimizing a quadratic function with two quadratic constraints*, J. Comput. Math., 9 (1991), pp. 348–359.

[17] Y. Zhang, *Computing a Celis–Dennis–Tapia trust-region step for equality constrained optimization*, Math. Programming, 55 (1992), pp. 109–124.

SIAM J. OPTIM.
Vol. 7, No. 3, pp. 595–619, August 1997

A NEW ALGORITHM FOR SOLVING STRICTLY CONVEX QUADRATIC PROGRAMS[*]

WU LI[†] AND JOHN SWETITS[†]

Abstract. We reformulate convex quadratic programs with simple bound constraints and strictly convex quadratic programs as problems of unconstrained minimization of convex quadratic splines. Therefore, any algorithm for finding a minimizer of a convex quadratic spline can be used to solve these quadratic programming problems. In this paper, we propose a Newton method to find a minimizer of a convex quadratic spline derived from the unconstrained reformulation of a strictly convex quadratic programming problem. The Newton method is a "natural mixture" of a descent method and an active-set method. Moreover, it is an iterative method, yet it terminates in finite operations (in exact arithmetic).

Key words. convex quadratic programs, convex quadratic splines, active-set methods, Newton methods, exact penalty functions

AMS subject classifications. Primary, 90C20; Secondary, 49M40

PII. S1052623493246045

1. Introduction.

In this paper, we present new ideas and algorithms for solving the convex quadratic programming problem

$$\min\left\{\frac{1}{2}x^TMx-b^Tx : l\le Ax\le u\right\}, \tag{1.1}$$

where M is an $n\times n$ symmetric positive semidefinite matrix, A is an $m\times n$ matrix, $b\in\mathbb{R}^n$ (a vector of n components), and $l,u\in\mathbb{R}^m$ (vectors of m components). Our approach is to reformulate (1.1) as an unconstrained minimization problem with a convex quadratic spline (i.e., a differentiable convex piecewise quadratic function) as the objective function and to solve the new unconstrained problem. The unconstrained reformulation is possible whenever M is nonsingular or A is a nonsingular square matrix ($m=n$).

The main efforts in developing numerical algorithms for quadratic programs are focused on the following three types of methods: active-set methods, matrix splitting methods, and interior-point methods. Interior-point methods are particularly efficient for solving linear programs, and efforts have been made to use interior-point methods for solving quadratic programs (cf., [36], [37], [29], [44], [8], and references therein). Matrix splitting methods contain a large class of algorithms for solving linear equations, quadratic programs, and linear complementarity problems [20], [28], [31], [32], [33], [34]. Even though matrix splitting methods and interior-point methods for solving quadratic programs attracted much attention in recent years, active-set methods are still the dominant approach in the development of software for quadratic programs (cf. [36]). Active-set methods solve a quadratic problem in finite steps through pivoting and determination of "active sets," if applicable (cf., for example, [39], [12], [14], [41]). In general, active-set methods use an add-or-drop-one-constraint strategy

[*] Received by the editors March 23, 1993; accepted for publication (in revised form) April 10, 1996.
http://www.siam.org/journals/siopt/7-3/24604.html

[†] Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA 23529 (wuli@math.odu.edu, swetits@math.odu.edu).

in each iteration and are slow if the initial estimate of the active set is markedly different from the optimal one. For example, subroutine QPROG in the IMSL library (cf. [18]) is Powell's implementation [41] of Goldfarb and Idnani's active-set method [14] and is a stable and fairly efficient general purpose algorithm for solving strictly convex quadratic programs. However, for problems with a large number of active constraints, QPROG is slow to find the solution because it starts with the unconstrained minimizer of the objective function (i.e., its initial estimate of the active set is the empty set). Two kinds of remedies have been proposed to improve the performance of active-set methods: one is to find a good starting point [4] and another is to allow the swapping of many constraints in each iteration [46], [35], [11].

In studying r-convex approximation problems [27], we have discovered the simple but subtle fact that the system of piecewise linear equations associated with the Karush–Kuhn–Tucker conditions of (1.1) is a linear transformation of the gradient of a convex quadratic spline function whenever M is nonsingular or A is a nonsingular square matrix ($m = n$). In particular, we have given an explicit unconstrained reformulation of (1.1) when M (or A) is the identity matrix [27]. Under the additional assumption that AA^T (or M) is nonsingular, the objective function of the unconstrained minimization problem is actually a strictly convex quadratic spline. In this case, we can use a Newton method with line search to solve the unconstrained minimization problem. The algorithm is a descent method that terminates in finite operations (in exact arithmetic). The method was tested on r-convex approximation problems, and the numerical results showed that the method is quite stable and efficient [27]. However, the disadvantage of our method is that it is too restrictive to require the nonsingularity of both M and AA^T. The objective of the present paper is to design an algorithm that enjoys all the nice properties of the Newton method but does not require the nonsingularity of AA^T. Our new algorithm is a mixture of an active-set method and a dual descent method, which has the following properties:

1. it terminates in a finite number of operations (in exact arithmetic);
2. as an active-set method, it requires no primal or dual feasibility;
3. as an active-set method, it allows the swapping of many constraints at each iteration;
4. as a dual descent method, it can use any starting point;
5. as a dual descent method, it can easily recover from severe numerical errors in computation.

Note that a reformulation of (1.1) as an unconstrained minimization problem is not a new idea. One can use penalty functions to get unconstrained reformulations of (1.1) (cf. [6]). However, the merit of our reformulation is that the new objective function is a convex quadratic spline function on $\mathbb{R}^n$ or $\mathbb{R}^m$. This facilitates various unconstrained minimization techniques for solving the unconstrained reformulation of (1.1).

In summary, our goal is to show that unconstrained reformulations of (1.1) lead to development of new algorithms for solving (1.1). In section 2, we provide the explicit unconstrained reformulations of (1.1) when M is nonsingular or A is a nonsingular square matrix. Section 3 contains two basic results about descent methods for solving the problem of unconstrained minimization of a convex quadratic spline. In section 4, we present our new algorithm, called QPspline, for solving (1.1) when M is nonsingular and prove its finite termination property. Comparison of numerical performance of QPspline and QPROG in the IMSL library is given in section 5, which shows the potential of QPspline as a general purpose algorithm for solving strictly convex quadratic programming problems. Finally, in section 6, we discuss advantages and disadvantages of QPspline and point out potential research directions.

For an index set J, x_J (or B_J) denotes the vector (or the matrix) consisting of components (or rows) of x (or B) whose indices are in J. The transpose of a vector x (or a matrix B) is written as x^T (or B^T). The gradient of a function f on $\mathbb{R}^m$ is denoted by f'. For a vector x, $(x)_+$ is the vector whose ith component is $\max\{x_i, 0\}$ and $(x)_l^u$ is the vector whose ith component is $\max\{\min\{x_i, u_i\}, l_i\}$. A differentiable function f on $\mathbb{R}^m$ is called a quadratic spline if f' is a piecewise linear mapping. That is, f is a quadratic spline if and only if f is differentiable and there are finitely many convex polyhedral subsets $\{W_i\}_{i=1}^r$ such that $\bigcup_{i=1}^r W_i = \mathbb{R}^m$, and f is a quadratic function on each W_i. We write $x \leq y$ if $x_i \leq y_i$ for all i. The 2-norm on $\mathbb{R}^n$ is defined as $\|x\| := (\sum_{i=1}^n x_i^2)^{\frac{1}{2}}$ and the ℓ_∞ norm on $\mathbb{R}^n$ is defined as $\|x\|_\infty := \max_{1\leq i\leq n} |x_i|$.

2. Unconstrained reformulations.

In this section, we show that (1.1) can be reformulated as an unconstrained minimization problem with a convex quadratic spline function as the objective function whenever M is nonsingular or A is a nonsingular square matrix ($m = n$). In order to provide a unified approach for our reformulations, we first consider a special class of piecewise linear mappings that are gradients of quadratic splines.

LEMMA 2.1. *Suppose that P, Q are $s \times s$ matrices, P is symmetric, β is any constant, $p, q \in \mathbb{R}^s$, and l, u are vectors of s components such that $l_i \in [-\infty, \infty)$, $u_i \in (-\infty, \infty]$, and $l_i \leq u_i$. Then $\gamma(z) := Pz + p + \beta Q^T(Qz + q)_l^u$ is the gradient of the quadratic spline*

$$\Gamma(z) := \frac{1}{2}z^T(P+\beta Q^TQ)z+z^T(p+\beta Q^Tq)-\frac{\beta}{2}\|(l-(Qz+q))_+\|^2-\frac{\beta}{2}\|((Qz+q)-u)_+\|^2.$$

Moreover, if both P and $P+\beta Q^TQ$ are positive semidefinite, then $\Gamma(z)$ is also convex.

Proof. Since $(v)_l^u \equiv v + (l - v)_+ - (v - u)_+$, we have

$$\gamma(z) = (P + \beta Q^TQ)z + (p + \beta Q^Tq) + \beta Q^T(l - (Qz + q))_+ - \beta Q^T((Qz + q) - u)_+.$$

By $\frac{d}{dt}(t)_+^2 = 2(t)_+$ and the chain rule, we know that $\gamma(z)$ is the gradient of $\Gamma(z)$.

Let $S_{-1,i} := (-\infty, l_i]$, $S_{0,i} = [l_i, u_i]$, and $S_{1,i} := [u_i, \infty)$. For $\tau := (\tau_1, \ldots, \tau_s)$ with $\tau_i \in \{-1, 0, 1\}$ (i.e., $\tau \in \{-1, 0, 1\}^s$), define

$$Z_\tau := \{z \in \mathbb{R}^s : (Qz + q)_i \in S_{\tau_i,i} \quad \text{for } 1 \leq i \leq s\}.$$

Obviously,

$$\mathbb{R}^s := \bigcup_{\tau\in\{-1,0,1\}^s} Z_\tau.$$

For any τ, Z_τ is a closed set that may be empty. If Z_τ is not empty, then

$$\gamma(y) - \gamma(z) = (P + \beta Q^T\Sigma_\tau Q)(y - z) \quad \text{for } y, z \in Z_\tau,$$

where Σ_τ is the diagonal matrix whose jth diagonal entry is 1 if $\tau_j = 0$ and 0 if $\tau_j \neq 0$. Let $y, z \in \mathbb{R}^s$. Then the intersection of the line segment $[y, z] := \{y_\theta := \theta z + (1 - \theta)y : 0 \leq \theta \leq 1\}$ and each convex set Z_τ is also a line segment if not empty. Therefore, there exist $0 =: \theta_0 < \theta_1 < \cdots < \theta_r := 1$ and $\tau^i \in \{-1, 0, 1\}^s$ for $1 \leq i \leq r$ such that

$[y_{\theta_{i-1}}, y_{\theta_i}] \subset Z_{\tau^i}$. Thus,

$$\begin{aligned}\gamma(y) - \gamma(z) &= \sum_{i=1}^{r} (\gamma(y_{\theta_{i-1}}) - \gamma(y_{\theta_i})) \\ &= \sum_{i=1}^{r} (P + \beta Q^T \Sigma_{\tau^i} Q)(y_{\theta_{i-1}} - y_{\theta_i}) \\ &= \sum_{i=1}^{r} (\theta_i - \theta_{i-1})(P + \beta Q^T \Sigma_{\tau^i} Q)(y - z).\end{aligned}$$

Now suppose that both P and $P + \beta Q^T Q$ are positive semidefinite. If $\beta \geq 0$, $(P + \beta Q^T \Sigma_{\tau^i} Q)$ is positive semidefinite, since it is a summation of two positive semidefinite matrices. If $\beta < 0$, then

$$\begin{aligned}v^T(P + \beta Q^T \Sigma_{\tau^i} Q)v &= v^T P v + \beta \sum_{\substack{j=1 \\ \tau^i_j = 0}}^{s} |(Qv)_j|^2 \\ &\geq v^T P v + \beta \|Qv\|^2 = v^T(P + \beta Q^T Q)v \geq 0,\end{aligned}$$

where the last inequality follows from the positive semidefiniteness of $P + \beta Q^T Q$. Therefore,

$$(y - z)^T(\gamma(y) - \gamma(z)) = \sum_{i=1}^{r} (\theta_i - \theta_{i-1})(y - z)^T (P + \beta Q^T \Sigma_{\tau^i} Q)(y - z) \geq 0.$$

That is, $\gamma(z)$ is monotone. Hence, $\Gamma(z)$ is convex (cf. [38]). □

Remark. A special case of the above lemma was first given in [26].

It is well known that the Karush–Kuhn–Tucker conditions of (1.1) form a system of piecewise linear equations. More generally, an affine variational inequality problem is equivalent to a system of piecewise linear equations, even though the system of piecewise linear equations involves a projection to a convex polyhedral set (cf. [7] or Proposition 2.3 in [17]). A special case of the following explicit reformulation of the Karush–Kuhn–Tucker conditions with an arbitrary positive constant α was given in [27], which led to the reformulation of (1.1) (when $M = I$ or $A = I$) as an unconstrained minimization problem with a convex quadratic spline as the objective function.

LEMMA 2.2. *Let α be a positive constant. Then x is a solution of* (1.1) *if and only if there exists $w \in \mathbb{R}^m$ such that*

$$Mx - b - A^T w = 0, \quad Ax = (Ax - \alpha w)_l^u. \tag{2.1}$$

Proof. It is well known from the Karush–Kuhn–Tucker conditions that x is a solution of (1.1) if and only if there exists $w \in R^m$ such that, for $1 \leq i \leq m$,

$$Mx - b - A^T w = 0, \tag{2.2}$$

$$\begin{aligned}&(Ax)_i = l_i \quad \text{if} \quad w_i > 0, \\ &(Ax)_i = u_i \quad \text{if} \quad w_i < 0, \\ &l_i \leq (Ax)_i \leq u_i \quad \text{if} \quad w_i = 0.\end{aligned} \tag{2.3}$$

Therefore, it suffices to show that the complementarity conditions in (2.3) are equivalent to the system of piecewise linear equations: $Ax = (Ax - \alpha w)_l^u$.

Obviously, (2.3) implies $Ax = (Ax - \alpha w)_l^u$. Now, assume $Ax = (Ax - \alpha w)_l^u$. By the definition of $(z)_l^u$, $Ax = (Ax - \alpha w)_l^u$ implies $l \leq Ax \leq u$. Hence, (2.3) holds for $w_i = 0$. If $w_i > 0$ and $(Ax)_i > l_i$, then $((Ax - \alpha w)_l^u)_i < (Ax)_i$, which is impossible. Thus, $w_i > 0$ implies $(Ax)_i = l_i$. Finally, $w_i < 0$ and $(Ax)_i < u_i$ imply $((Ax - \alpha w)_l^u)_i > (Ax)_i$, which contradicts the assumption that $Ax = (Ax - \alpha w)_l^u$; we must have $(Ax)_i = u_i$ when $w_i < 0$. □

Remark. In the above reformulation, we treat an equality constraint as a special two-sided inequality constraint (with $l_i = u_i$). This provides a unified treatment of both inequality and equality constraints. The advantage is to avoid treating one two-sided constraint as two one-sided constraints, since it unnecessarily increases the dimension of the dual space. In the case that M is nonsingular, (1.1) is equivalent to a problem of unconstrained minimization of $\Phi(w)$ (cf. Theorem 2.5) and one can see that the increase of the dimension of w-space is undesirable.

The proof of Lemma (2.2) is still valid if we replace α by a diagonal matrix with positive diagonal entries.

PROPOSITION 2.3. *Let D be a diagonal matrix with positive diagonal entries; then, x is a solution of* (1.1) *if and only if there exists $w \in \mathbb{R}^m$ such that*

$$Mx - b - A^T w = 0, \quad Ax = (Ax - Dw)_l^u. \tag{2.4}$$

Proposition 2.3 is very similar to Mangasarian's reformulation of linear complementarity problems (cf. Lemma 2.1 in [33]). Special cases of the above proposition have been used and exploited by many people, for example, Moré and Toraldo [35], Bertsekas [1], and Conn, Gould, and Toint [3].

PROPOSITION 2.4. *For any $n \times n$ matrix M and b, $w = Mx - b \geq 0, x \geq 0, w^T x = 0$ if and only if $x = (x - Dw)_+$ (i.e., $x = (x - D(Mx - b))_+$).*

When M is symmetric positive semidefinite, $l = 0$, $u = +\infty$, and $A = I$, Propositions 2.3 and 2.4 are equivalent, since the complementarity conditions are the Karush–Kuhn–Tucker conditions of (1.1). In this case, Proposition 2.3 and Lemma 2.2 are due to Mangasarian.

The reason we prefer Lemma 2.2 to Proposition 2.3 is its simplicity. In the following discussion, one can always replace α by D and get similar unconstrained reformulations of (1.1) involving an arbitrary D.

Now, based on Lemmas 2.1 and 2.2, we can have two unconstrained reformulations of (1.1), depending on whether M is nonsingular or A is a nonsingular square matrix.

First, suppose that M is nonsingular. Let $x(w) := M^{-1}(A^T w + b)$. Then we obtain the following equivalent system of (2.4):

$$Ax(w) = (Ax(w) - \alpha w)_l^u;$$

i.e.,

$$\varphi(w) := Ax(w) - (Ax(w) - \alpha w)_l^u = 0. \tag{2.5}$$

Let

$$B = \alpha I - AM^{-1}A^T.$$

Then, $AM^{-1}A^T = \alpha I - B$ and $B\varphi(w) = (\alpha B - B^2)w + BAM^{-1}b - B(-Bw + AM^{-1}b)_l^u$. Let $P := \alpha B - B^2$, $Q = -B$, $p = BAM^{-1}b$, $q = AM^{-1}b$, and $\beta = 1$. Then

$P+\beta Q^TQ=\alpha B$, $p+\beta Q^Tq=0$, $c:=l-q=l-AM^{-1}b$, and $d:=q-u=AM^{-1}b-u$. By Lemma 2.1, $B\varphi(w)$ is the gradient of the following quadratic spline function:

$$\Phi(w):=\frac{\alpha}{2}w^TBw-\frac{1}{2}\|(Bw+c)_+\|^2-\frac{1}{2}\|(d-Bw)_+\|^2. \tag{2.6}$$

That is, $\Phi'(w)=B\varphi(w)$.

If $\alpha>\|AM^{-1}A^T\|$, where $\|AM^{-1}A^T\|$ is the spectral radius of $AM^{-1}A^T$, then B is a symmetric positive definite matrix whose eigenvalues are in the interval $(0,\alpha]$. Therefore, $P=\alpha B-B^2$ is symmetric positive semidefinite and $P+\beta Q^TQ=(\alpha B-B^2)+(-B)^T(-B)=\alpha B$ is positive definite. By Lemma 2.1, $\Phi(w)$ is a convex quadratic spline. Therefore, $\varphi(w^*)=0$ if and only if $B\varphi(w^*)=0$, which is equivalent to

$$\Phi(w^*)=\inf_{w\in\mathbb{R}^m}\Phi(w). \tag{2.7}$$

The above analysis leads to the following unconstrained reformulation of (1.1) with a positive definite M.

THEOREM 2.5. *Suppose that M is a symmetric positive definite matrix and $\alpha>\|AM^{-1}A^T\|$. Then $\Phi(w)$ is a convex quadratic spline function. Moreover, x^* is the solution of* (1.1) *if and only if $x^*=M^{-1}(A^Tw^*+b)$, where w^* is a solution of* (2.7) (*or* (2.5)).

If M is singular, then we cannot solve $Mx-b-A^Tw=0$ for x. However, if A is a nonsingular square matrix, then we can write w in terms of x as follows:

$$w=(A^T)^{-1}A^Tw=(A^T)^{-1}(Mx-b). \tag{2.8}$$

Substituting (2.8) into the second equation in (2.4), we obtain

$$Ax=((A-\alpha(A^T)^{-1}M)x+\alpha(A^T)^{-1}b)_l^u. \tag{2.9}$$

In this case, one cannot find a quadratic spline function whose gradient is a linear transformation of $Ax-((A-\alpha(A^T)^{-1}M)x+\alpha(A^T)^{-1}b)_l^u$, due to the asymmetry of $A-\alpha(A^T)^{-1}M$. Fortunately, with the substitution $y=Ax$, we can rewrite (2.9) as

$$y=(Ey+q)_l^u, \tag{2.10}$$

where $E:=I-\alpha(A^{-1})^TMA^{-1}$ and $q:=\alpha(A^T)^{-1}b$. Note that y is a solution of (2.10) if and only if $x=A^{-1}y$ is a solution of (2.9), which is equivalent to the fact that x and $w:=(A^T)^{-1}(Mx-b)$ satisfy the Karush–Kuhn–Tucker conditions (2.4). Hence, y is a solution of (2.10) if and only if $x:=A^{-1}y$ solves (1.1).

Let $\psi(y):=y-(Ey+q)_l^u$. Then, $E\psi(y)=Ey-E(Ey+q)_l^u$. Let $P\equiv Q:=E$ and $\beta=-1$. Then, by Lemma 2.1, $E\psi(y)$ is the gradient of the following quadratic spline:

$$\Psi(y):=\frac{1}{2}y^T(E-E^2)y-y^TEq+\frac{1}{2}\|(l-(Ey+q))_+\|^2+\frac{1}{2}\|((Ey+q)-u)_+\|^2.$$

If $0<\alpha<\|(A^{-1})^TMA^{-1}\|^{-1}$, then E is a symmetric positive definite matrix whose eigenvalues are in the interval $(0,1]$. Hence, $P+\beta Q^TQ=E-E^2$ is symmetric positive semidefinite. By Lemma 2.1, $\Psi(y)$ is convex. In this case, $\psi(y^*)=0$ if and only if $E\psi(y^*)=0$, which is equivalent to

$$\Psi(y^*)=\inf_{y\in\mathbb{R}^n}\Psi(y). \tag{2.11}$$

The following unconstrained reformulation of (1.1) with a nonsingular A is the consequence of the above discussions.

THEOREM 2.6. *Let A be a nonsingular square matrix ($m = n$) and $0 < \alpha < \|(A^{-1})^T M A^{-1}\|^{-1}$. Then $\Psi(y)$ is a convex quadratic spline function. Moreover, x^* is the solution of* (1.1) *if and only if $y^* = Ax^*$ is a solution of* (2.11) (*or* (2.10)).

Remark. Special cases of Theorems 2.5 and 2.6 were given in [27].

We would like to mention that, based on an ℓ_1 penalty function, Coleman and Hulbert [2] reformulated (1.1) with $l = -1, u = 1$, $A = I$ (the identity matrix), and a positive definite M as an unconstrained minimization of a convex piecewise quadratic function that is, in general, not differentiable. By using Glad and Polak's multiplier function [13] and a differentiable exact penalty function [6], Grippo and Lucidi [15], [16] also derived an unconstrained reformulation of (1.1) when $A = I$ and the feasible region $\{x : l \leq x \leq u\}$ is compact. The derived penalty function $P(x, \epsilon)$ is a complicated piecewise rational function and is only continuously differentiable in a compact neighborhood $\mathcal{D}$ of the feasible region. The penalty function $P(x, \epsilon)$ approaches ∞ as x moves toward the boundary of $\mathcal{D}$ because of barrier terms involved. The penalty parameter ϵ has to be determined by maximization of some linear and nonlinear functions. In contrast, our penalty function $\Psi(x)$ for (1.1) is a convex quadratic spline function when $A = I$. A parameter α involved in the definition of $\Psi(x)$ easily can be determined by the 2-norm of M. When M is positive definite, we actually reformulate (1.1) as the unconstrained minimization of a convex quadratic spline function $\Phi(w)$ in dual variables w. Again, a parameter α involved in the definition of $\Phi(w)$ easily can be determined by the 2-norm of $AM^{-1}A^T$. Note that (as pointed out by Di Pillo and Grippo), in practice, one could only choose a penalty parameter to get a differentiable exact penalty function of a constrained minimization problem with reference to some compact set [6]. However, for convex quadratic programs (1.1) with a positive definite matrix M or a nonsingular square matrix A, we can have differentiable exact penalty functions on either $\mathbb{R}^m$ or $\mathbb{R}^n$.

3. Unconstrained minimization of convex quadratic splines.

Due to our reformulations given in the previous section, it is natural to study the theory of unconstrained minimization of convex quadratic splines. Consider the following unconstrained minimization of a convex quadratic spline $f(w)$:

$$-\infty < f_{\min} := \min_{w \in \mathbb{R}^m} f(w). \tag{3.1}$$

Let W^* be the solution set of (3.1). That is, $W^* := \{w \in \mathbb{R}^m : f(w) = f_{\min}\}$. Here we establish two basic results (Lemmas 3.3 and 3.7) about descent methods for solving (3.1). Suppose that we generate a descent sequence $\{w^k\}$ (i.e., $f(w^k) \geq f(w^{k+1})$). The first basic result (Lemma 3.3) shows that if there exists a subsequence $\{k_j\}$ such that $\lim_{j\to\infty} \|f'(w^{k_j})\| = 0$, then $\lim_{k\to\infty} f(w^k) = f_{\min}$ (i.e., $\{w^k\}$ is a weakly convergent sequence). This provides a lot of flexibility to design a descent method that will generate a weakly convergent sequence. For example, we can generate a weakly convergent descent sequence $\{w^k\}$ freely as long as there is a subsequence $\{k_j\}$ such that $\|f'(w^{k_j})\| \leq \gamma(f(w^{k_j}) - f(w^{k_j+1}))$. A weakly convergent sequence actually allows us to identify (3.1) with a feasibility problem. To be more specific, let $\{W_i\}_1^s$ be a collection of polyhedral subsets of $\mathbb{R}^m$ such that $\mathbb{R}^m = \bigcup_{i=1}^s W_i$ and $f(w)$ is a quadratic function on each W_i. Such W_i's can be determined by the representation of $f(w)$ (cf. $\Psi(y)$ and $\Phi(w)$ in section 2). We say that W_i is a solution region if W_i contains a solution of (3.1). If w^k is in a solution region W_i, then (3.1) is equivalent to the following feasibility problem:

$$A^i w - b^i = 0, \quad w \in W_i, \tag{3.2}$$

where $f'(w) = A^i w - b^i$ on W_i. It turns out (cf. Lemma 3.7) that, for k large enough, w^k will always be in a solution region, provided that $\lim_{k\to\infty} f(w^k) = f_{\min}$. This allows us to design descent methods for solving (2.7) that terminate in finite iterations. One way to design such a method is to generate a good approximate solution w^{k+1} of the feasibility problem (3.2) if w^k is in W_i and, best of all, to generate a solution of (3.2) whenever it is solvable. The design of Algorithm 4.1 is motivated by the ideas illustrated in Lemmas 3.3 and 3.7. Not surprisingly, Lemmas 3.3 and 3.7 will be crucial in the proof of finite termination of Algorithm 4.1.

In order to derive that the gradients of iterates $\{x^k\}$ generated by a descent method converge to 0 (i.e., $\lim_{k\to\infty} \|f'(x^k)\| = 0$), one needs an estimate of $\|f'(x^k)\|$ in terms of $(f(x^k) - f(x^{k+1}))$. The following inequality (3.3) provides a means to do so. The inequality was implicitly used in proving Wolfe's weak convergence result on descent methods for any unconstrained minimization problem [45]. However, if $h(w)$ is not convex, then the same inequality requires an additional assumption $h(w) \geq h(w+tz) - \delta t z^T h'(w)$ with a fixed positive constant δ (cf. pp. 118–121 in [5]).

LEMMA 3.1. *Suppose that $h(w)$ is a convex function, its gradient $h'(w)$ is Lipschitz continuous, and $0 < \beta < 1$. Then, there exists a positive constant γ (depending only on h and β) such that*

$$\left(\frac{z^T h'(w)}{\|z\|}\right)^2 \leq \gamma(h(w) - h(w+tz)), \tag{3.3}$$

whenever $t > 0$ and $0 \geq z^T h'(w+tz) \geq \beta \cdot z^T h'(w)$.

Proof. Since $h(w+\theta z)$ is a convex function of θ, the derivative $g(\theta) := \frac{d}{d\theta} h(w+\theta z) = z^T h'(w+\theta z)$ is a nondecreasing function of θ. By the intermediate value theorem, there exists a positive constant $\hat{t} \leq t$ such that $g(\hat{t}) = \beta \cdot z^T h'(w)$. Since $g(\theta) \leq g(\hat{t}) \leq g(\tau) \leq 0$ for $0 \leq \theta \leq \hat{t} \leq \tau \leq t$,

$$h(w+tz) = h(w) + \int_0^t g(\theta)d\theta \leq h(w) + \int_0^{\hat{t}} g(\theta)d\theta \leq h(w) + g(\hat{t})\hat{t} = h(w) + \beta \cdot \hat{t} z^T h'(w). \tag{3.4}$$

Rewrite (3.4) as follows:

$$0 \leq -z^T h'(w) \leq \frac{1}{\beta \hat{t}}(h(w) - h(w+tz)). \tag{3.5}$$

Moreover, we have

$$-(1-\beta) z^T h'(w) = z^T h'(w+\hat{t}z) - z^T h'(w) \leq \lambda \hat{t} \|z\|^2, \tag{3.6}$$

where the first equality follows from the definition of $\hat{t}$ and the second inequality is the consequence of the Cauchy–Schwarz inequality and the Lipschitz continuity of h' (with the Lipschitz constant λ). Since $\hat{t} > 0$, by (3.5) and (3.6) we obtain

$$\begin{aligned}
\left(\frac{z^T h'(w)}{\|z\|}\right)^2 &= \frac{(-z^T h'(w))(-z^T h'(w))}{\|z\|^2} \\
&\leq \left(\frac{1}{\beta \hat{t}}(h(w) - h(w+tz))\right)\left(\frac{\lambda}{1-\beta}\hat{t}\|z\|^2\right)\frac{1}{\|z\|^2} \\
&= \frac{\beta\lambda}{1-\beta}(h(w) - h(w+tz)). \qquad \square
\end{aligned}$$

To prove the first basic lemma in this section, we also need the following error estimate for approximate solutions of a piecewise linear equation, which is a consequence of the upper Lipschitz continuity of a polyhedral mapping proved by Robinson [42].

LEMMA 3.2 (Robinson's theorem [42]). *Let $g(w)$ be a piecewise linear mapping and let $Y^* := \{w : g(w) = 0\} \neq \emptyset$. Then there exist positive constants ϵ and λ such that*

$$\operatorname{dist}(w, Y^*) := \inf_{w^* \in Y^*} \|w - w^*\| \leq \lambda \|g(w)\| \quad \textit{for } \|g(w)\| \leq \epsilon.$$

LEMMA 3.3. *Suppose that $f(w^{k+1}) \leq f(w^k)$ for $k = 0, 1, \ldots$. If there exists a subsequence $\{k_j\}$ such that $\lim_{j\to\infty} \|f'(w^{k_j})\| = 0$, then $\lim_{k\to\infty} f(w^k) = f_{\min}$.*

Proof. Let $W^* := \{w \in \mathbb{R}^m : f'(w) = 0\}$. Since the gradient $f'(w)$ of a quadratic spline is a piecewise linear mapping, $\lim_{j\to\infty} \operatorname{dist}(w^{k_j}, W^*) = 0$ by Lemma 3.2. Let $\hat{w}^{k_j} \in W^*$ be such that $\operatorname{dist}(w^{k_j}, W^*) = \|w^{k_j} - \hat{w}^{k_j}\|$. Since f is convex, $f(w) = f_{\min}$ if and only if $f'(w) = 0$. Thus, $f(\hat{w}^{k_j}) = f_{\min}$. By the mean value theorem, there exists $0 < \theta_j < 1$ such that

$$\begin{aligned} f(w^{k_j}) - f_{\min} &= f(w^{k_j}) - f(\hat{w}^{k_j}) \\ &= (w^{k_j} - \hat{w}^{k_j})^T f'(\hat{w}^{k_j} + \theta_j(w^{k_j} - \hat{w}^{k_j})) \\ &= (w^{k_j} - \hat{w}^{k_j})^T (f'(\hat{w}^{k_j} + \theta_j(w^{k_j} - \hat{w}^{k_j})) - f'(\hat{w}^{k_j})) \\ &\leq \lambda \|w^{k_j} - \hat{w}^{k_j}\|^2, \end{aligned} \tag{3.7}$$

where the last equality follows from $f'(\hat{w}^{k_j}) = 0$ and the last inequality is the consequence of the Cauchy–Schwartz inequality and the Lipschitz continuity of the piecewise linear mapping $f'(w)$ (with the Lipschitz constant λ). By (3.7), $\{f(w^k)\}$ has a subsequence converging to $f_{\min}$; hence, $\lim_{k\to\infty} f(w^k) = f_{\min}$. □

The next lemma shows one way to generate a weakly convergent descent sequence.

LEMMA 3.4. *Suppose that $f(w^{k+1}) \leq f(w^k)$ for $k = 0, 1, \ldots$, and $\mathcal{D}$ is a collection of finitely many positive definite matrices. If there are infinitely many k's such that*

$$(w^{k+1} - w^k)^T f'(w^{k+1}) = 0 \quad \textit{and} \quad w^{k+1} = w^k - t_k D^k f'(w^k) \ \textit{for } D^k \in \mathcal{D}, t_k > 0, \tag{3.8}$$

then $\lim_{k\to\infty} f(w^k) = f_{\min}$.

Proof. Note that the first equality in (3.8) is an exact line-search condition. Applying Lemma 3.1 with $w = w^k$, $t = t_k$, and $z = -D^k f'(w^k)$, we obtain that there exists a positive constant γ (depending only on f) such that if (3.8) holds, then

$$\left(\frac{(D^k f'(w^k))^T f'(w^k)}{\|D^k f'(w^k)\|} \right)^2 \leq \gamma (f(w^k) - f(w^{k+1})). \tag{3.9}$$

Since $\mathcal{D}$ contains only finitely many positive definite matrices, there exists a positive constant δ such that

$$\delta \|v\|^2 \leq v^T D v \quad \text{and} \quad \|Dv\| \leq \frac{1}{\delta} \|v\| \quad \text{for } D \in \mathcal{D}, v \in \mathbb{R}^m.$$

Thus,

$$\|D^k f'(w^k)\| \leq \frac{1}{\delta} \|f'(w^k)\| \quad \text{and} \quad (D^k f'(w^k))^T f'(w^k) = (f'(w^k))^T D^k f'(w^k) \geq \delta \|f'(w^k)\|^2. \tag{3.10}$$

It follows from (3.9) and (3.10) that if (3.8) holds, then

$$\|f'(w^k)\|^2 \le \frac{\gamma}{\delta^4}(f(w^k) - f(w^{k+1})). \tag{3.11}$$

Let $\{k_j\}$ be a subsequence such that (3.8) holds for $k = k_j$, $j = 1, 2, \ldots$. Since $\{f(w^k)\}$ is a nonincreasing sequence bounded below, it converges; $\lim_{k\to\infty}(f(w^k) - f(w^{k+1})) = 0$. By (3.11), $\lim_{j\to\infty}\|f'(w^{k_j})\| = 0$. By Lemma 3.3, $\lim_{k\to\infty} f(w^k) = f_{\min}$. □

In order to prove the second basic lemma, we need the following Frank–Wolfe theorem about the existence of a solution of a quadratic program and an error estimate of feasible solutions of a convex piecewise quadratic program by Li.

LEMMA 3.5 (Frank–Wolfe theorem [10]). *If a quadratic function $g(w)$ is bounded below on a nonempty polyhedron W in $\mathbb{R}^m$, then $g(w)$ attains its infimum on W. That is, if $\inf_{w\in W} g(w) > -\infty$, then there exists $w^* \in W$ such that $g(w^*) = \inf_{w\in W} g(w)$.*

LEMMA 3.6 (see Li [19]). *Let $g(w)$ be a convex quadratic spline on a polyhedron W in $\mathbb{R}^m$ and $Y^* := \{w \in W : g(w) = g_{\min}\}$ with $g_{\min} := \min_{w\in W} g(w) > -\infty$. Then there exists a positive constant λ such that*

$$\operatorname{dist}(w, Y^*) \le \lambda\left((g(w) - g_{\min}) + (g(w) - g_{\min})^{\frac{1}{2}}\right) \quad \textit{for } w \in W.$$

LEMMA 3.7. *Suppose that $\{W_i\}_1^s$ are a collection of polyhedral subsets of $\mathbb{R}^m$ such that $f(w)$ is a quadratic function on each W_i. If $\lim_{k\to\infty} f(w^k) = f_{\min}$, then there exists $k^* \ge 1$ such that $w^k \in W_i$ implies $W_i \cap W^* \ne \emptyset$ for $k \ge k^*$.*

Proof. Since $W^* \times W_i$ is a closed convex polyhedral set, by Lemma 3.5 there exists $(w^*, w^i) \in W^* \times W_i$ such that

$$\epsilon_i := \|w^* - w^i\| = \inf_{(z^*,z)\in W^*\times W_i} \|z - z^*\|. \tag{3.12}$$

Let $\epsilon := \min\{\epsilon_i : \epsilon_i > 0\} > 0$. Since $f(w)$ is a convex quadratic spline, by Lemma 3.6 there exists a positive constant λ such that

$$\operatorname{dist}(w, W^*) \le \lambda\left((f(w) - f_{\min}) + (f(w) - f_{\min})^{\frac{1}{2}}\right) \quad \text{for } w \in \mathbb{R}^m. \tag{3.13}$$

Since $\lim_{k\to\infty}(f(w^k) - f_{\min}) = 0$, there exists k^* such that

$$(f(w^k) - f_{\min}) + (f(w^k) - f_{\min})^{\frac{1}{2}} < \frac{\epsilon}{\lambda} \quad \text{for } k \ge k^*. \tag{3.14}$$

We claim that $W_i \cap W^* \ne \emptyset$ if $w^k \in W_i$ and $k \ge k^*$.

In fact, by (3.13) and (3.14),

$$\operatorname{dist}(w^k, W^*) \le \lambda\left((f(w^k) - f_{\min}) + (f(w^k) - f_{\min})^{\frac{1}{2}}\right) < \epsilon \quad \text{for } k \ge k^*.$$

Since $w^k \in W_i$, by the definition of ϵ we must have $\epsilon_i = 0$; i.e., $W_i \cap W^* \ne \emptyset$. □

4. An algorithm for strictly convex quadratic programs. In this section we propose a new algorithm, called QPspline, for solving (1.1) when M is nonsingular. The algorithm is based on the reformulation (2.5) of (1.1). The main features of QPspline are outlined in the introduction. The real advantage of QPspline is its flexibility. Even though we have very limited numerical tests on its performance, its potential seems to be greater than we expected. The algorithm is a mixture of an active-set method and a dual descent method which are tied together in a natural way, due to the reformulation 2.5 of (1.1). It is essentially an iterative method, yet it terminates in finite operations (in exact arithmetic).

Algorithm 4.1 (QPspline) is a Newton method with exact line search for unconstrained minimization of spline function $\Phi(w)$. Due to possible singularity of the Hessian of $\Phi(w)$, we have to make some technical modifications for the computation of a Newton direction. Lemma 4.2 shows that Algorithm 4.1 is a well-defined descent method for unconstrained minimization of $\Phi(w)$. The proof of finite termination of Algorithm 4.1 is quite complicated. Our proof is based on the fact that Algorithm 4.1 is implicitly an active-set method. Lemma 4.3 reveals a feature of Algorithm 4.1 as an active-set method. However, this feature appears only if a linear system for the computation of a Newton direction is consistent. Lemma 4.4 ensures the consistency of this linear system once the current iterate is in a polyhedral region containing a dual solution of (1.1). Theorem 4.5 combines these results with the convergence results for unconstrained minimization of a convex quadratic spline given in section 3 to establish the finite termination of Algorithm 4.1.

Algorithm 4.1 (QPspline). *Let* $B = \alpha I - AM^{-1}A^T$ *with* $\alpha > \|AM^{-1}A^T\|$, $x(w) = M^{-1}(A^T w + b)$, *and* $\varphi(w)$ *and* $\Phi(w)$ *be defined as in* (2.5) *and* (2.6), *respectively. For any* w, *define*

$$J_l \equiv J_l(w) := \{i : (Ax(w) - \alpha w)_i < l_i\},$$
$$J_u \equiv J_u(w) := \{i : (Ax(w) - \alpha w)_i > u_i\}. \tag{4.1}$$

For any index set J, D^J *denotes the diagonal matrix such that the* j*th diagonal entry is* 1 *for* $j \in J$ *and* 0 *otherwise.*

```
(0) Let $w$ be any given starting point.
(1) If $\varphi(w) = 0$, then $x := M^{-1}(A^T w + b)$ is the solution and stop; otherwise, let
    $a_i = l_i$ for $i \in J_l$ and $a_i = u_i$ for $i \in J_u$.
(2) Let $J$ be a subset of $S := J_l \cup J_u$ such that $A_J$ is row independent and $A_J$
    has the same rank as $A_S$.
(3) Let $\hat{w} \in \mathbb{R}^m$ be such that $\hat{w}_j = 0$ for $j \notin J$ and $\hat{w}_J = (A_J M^{-1} A_J^T)^{-1}(a_J - A_J M^{-1} b)$.
(4) If $A_S x(\hat{w}) = a_S$, then compute $\hat{t} > 0$ such that $(\hat{w}-w)^T B\varphi(w+\hat{t}(\hat{w}-w)) = 0$,
    set $t := \min\{\hat{t}, 1\}$, update $w := w + t(\hat{w} - w)$, and go to step (1).
(5) Compute the dual descent direction $z = -(\alpha I - D^J B)^{-1}\varphi(w)$.
(7) Find a stepsize $t > 0$ such that $z^T B\varphi(w + tz) = 0$.
(8) Update $w := w + tz$ and go to step (1).
```

Remark. The name QPspline for this algorithm was suggested by Michael Saunders to emphasize solving a **Q**uadratic **P**rogram by using a **spline** merit/penalty function.

The index set S is treated as the current active set with active constraints $A_S x = a_S$. Thus, when $A_S x = a_S$ is consistent, we solve the corresponding quadratic program with equality constraints $A_S x = a_S$. It turns out that the solution is $x(\hat{w})$ (cf. Lemma 4.3). Moreover, $(\hat{w} - w)$ is a descent direction for Φ at w (cf. Lemmas 4.3 and 4.2 (4)). However, we do not force the next iterate to be $\hat{w}$. Instead, we rely on the line search procedure to decide whether $\hat{w}$ is a good candidate as the next iterate. If $\hat{t} \geq 1$, then $\Phi(w + \theta(\hat{w} - w))$ is a monotone decreasing function for $0 \leq \theta \leq 1$, $t = 1$, and $w := \hat{w}$ is the next iterate; otherwise, we would rather use the line minimizer $w + \hat{t}(\hat{w} - w)$ as the next iterate. However, due to linear dependence of the rows of A_S, $A_S x = a_S$ might not be consistent. This tells us that the current iterate w is not in a solution region.

Therefore, instead of trying to find a solution in the current iteration, we reduce the value of the objective function Φ as much as possible. Since we do not know whether $(\hat{w} - w)$ is a descent direction or not, we generate a descent direction z (cf. Lemma 4.2 (4)), which is very easy to compute based on a factorization of $A_J M^{-1} A_J^T$ (cf. Lemma 4.2 (1)). Therefore, even though $A_S x(\hat{w}) \neq a_S$, we only need to deal with one matrix $A_J M^{-1} A_J^T$ for the linear systems involved. Also, due to the simple structure of $\Phi(w)$, we can use a linear time algorithm to solve the line search problem (cf. [27], [40]).

In the case that A is row independent, J is always the same as S and we can skip steps (2)–(4). This is a Newton method with line search proposed by the authors [27]. For $A = \nabla_r$ (the rth divided difference matrix), $M = I$ (the $n \times n$ identity matrix) and $n^r \leq 10^9$, the Newton method exhibited the finite termination feature. When $r = 2$, $l = 0$, and $u = \infty$, we were able to produce a fairly accurate solution of the convex regression problem with n up to 2000, even though ∇_r is ill conditioned with condition number about n^r [27].

The following lemma clarifies some technical aspects of Algorithm 4.1, such as the nonsingularity of $\alpha I - D^J B$, the descent direction z, and the relation between $A_J M^{-1} A_J^T$ and $\alpha I - D^J B$.

LEMMA 4.2. *Let J be a subset of $\{i\}_1^m$ and $J^c := \{i\}_1^m \setminus J$.*

1. *For any given w, $(\alpha I - D^J B)z = \varphi(w)$ if and only if $z_i = \frac{1}{\alpha}\varphi(w)_i$ for $i \notin J$ and*

$$(A_J M^{-1} A_J^T) z_J = \varphi(w)_J - \frac{1}{\alpha}(A_J M^{-1} A_{J^c}^T)\varphi(w)_{J^c}.$$

2. *The matrix $(\alpha I - D^J B)$ is a nonsingular matrix if and only if A_J is row independent.*
3. *The matrix $(\alpha B - BD^J B)$ is always positive semidefinite. If A_J is row independent, then $(\alpha B - BD^J B)$ and $(\alpha B - BD^J B)^{-1}$ are positive definite.*
4. *If $(\alpha I - D^J B)z = -\varphi(w)$, then either $z^T \Phi'(w) < 0$ or $\Phi'(w) = 0$.*

Proof. By the definition of D^J, we have $(\alpha I - D^J B)_J = A_J M^{-1} A^T$ and $(\alpha I - D^J B)_{J^c} = \alpha I_{J^c}$. Therefore, $(\alpha I - D^J B)z = v$ if and only if $z_i = \frac{1}{\alpha} v_i$ for $i \notin J$ and

$$A_J M^{-1} A^T z = v_J. \tag{4.2}$$

Substitute $z_{J^c} = \frac{1}{\alpha} v_{J^c}$ into (4.2); we then have

$$A_J M^{-1} A_J^T z_J = v_J - \frac{1}{\alpha} A_J M^{-1} A_{J^c}^T v_{J^c}. \tag{4.3}$$

Statement (1) follows from (4.3). Note that $(\alpha I - D^J B)$ is nonsingular if and only if the above system has a unique solution for any v, which is equivalent to the nonsingularity of $A_J M^{-1} A_J^T$. However, $A_J M^{-1} A_J^T$ is nonsingular if and only if A_J is row independent. This proves statement (2).

For any $v \in \mathbb{R}^m$,

$$v^T(\alpha B - BD^J B)v = \alpha v^T B v - v^T B D^J B v \geq \alpha v^T B v - v^T B B v = v^T(\alpha B - B^2)v \geq 0,$$

since $(\alpha B - B^2)$ is a positive semidefinite matrix (cf. the paragraph before Theorem 2.5). That is, $(\alpha B - BD^J B)$ is symmetric positive semidefinite. If A_J is row independent, then $B(\alpha I - D^J B)$ is also nonsingular (cf. Lemma 4.2 (2)) and $B(\alpha I - D^J B)$ is actually positive definite. Therefore, $(\alpha B - BD^J B)^{-1}$ is also positive definite. This proves statement (3).

Finally, by the definition of z, we have

$$\Phi'(w) = B\varphi(w) = -B(\alpha I - D^J B)z = -(\alpha B - BD^J B)z.$$

Since $(\alpha B - BD^J B)$ is positive semidefinite (cf. statement (2)), we have

$$z^T \Phi'(w) = -z^T(\alpha B - BD^J B)z \leq 0.$$

If $z^T\Phi'(w) = 0$, then $\Phi'(w) = -(\alpha B - BD^J B)z = 0$. □

The next lemma shows that, whenever $A_S x = a_S$ is consistent, $x(\hat{w})$ is actually the solution of the corresponding quadratic problem with equality constraints $A_S x = a_S$. Moreover, along with Lemma 4.2 (4), it proves that $(\hat{w}-w)$ is also a descent direction.

LEMMA 4.3. *For any w, let $\hat{w}$, S, and a_S be given as in Algorithm* 4.1*. Suppose that $A_S x = a_S$ is consistent. Then $(\alpha I - D^S B)(w - \hat{w}) = \varphi(w)$, and $x^* := M^{-1}(A^T\hat{w} + b)$ is the solution of the following strictly convex quadratic program with equality constraints:*

$$\min_x \ \frac{1}{2}x^T Mx - b^T x \quad \textit{subject to} \quad A_S x = a_S. \tag{4.4}$$

Proof. Since $A_S x = a_S$ is consistent, $M^{-1}(A_S^T y_S + b)$ is the solution of (4.4) if and only if

$$A_S M^{-1}(A_S^T y_S + b) = a_S. \tag{4.5}$$

Since $J \subset S$ and A_J has the same row rank as A_S, y_S is a solution of (4.5) if and only if

$$A_J M^{-1}(A_S^T y_S + b) = a_J. \tag{4.6}$$

By the definition of $\hat{w}$, $\hat{w}_S$ is a solution of (4.6); hence, $x^* := M^{-1}(A^T\hat{w} + b) = M^{-1}(A_S^T\hat{w}_S + b)$ is the solution of (4.4). Moreover,

$$A_S M^{-1}(A_S^T \hat{w}_S + b) = a_S. \tag{4.7}$$

Let $z := w - \hat{w}$. By the definition of S and $\varphi(w)$, $\varphi(w)_i = \alpha w_i$ for $i \notin S$ and $\varphi(w)_i = (A_S x(w) - a_S)_i$ for $i \in S$. Hence, $z_i = \frac{1}{\alpha}\varphi(w)_i$ for $i \notin S$. By (4.7),

$$\begin{aligned}(A_S M^{-1} A_S^T)z_S &= (A_S M^{-1} A_S^T)w_S - (A_S M^{-1} A_S^T)\hat{w}_S \\ &= (A_S M^{-1} A_S^T)w_S - a_S + (A_S M^{-1})b \\ &= A_S M^{-1}(A^T w + b) - a_S - (A_S M^{-1} A_{S^c}^T)w_{S^c} \\ &= \varphi(w)_S - \frac{1}{\alpha}(A_S M^{-1} A_{S^c}^T)\varphi(w)_{S^c}.\end{aligned}$$

By Lemma 4.2 (1), $(\alpha I - D^S B)z = \varphi(w)$. □

One might wonder why we wish to generate $\hat{w}$ such that $x(\hat{w})$ is the solution of (4.4). The answer is very simple: eventually, $x(\hat{w})$ is actually the solution of (1.1).

LEMMA 4.4. *If $\lim_{k\to\infty}\Phi(w^k) = \Phi_{\min}$, then there exists an integer $k^* \geq 1$ such that x^* is the solution of (1.1) for $k \geq k^*$ if and only if x^* is the solution of the following strictly convex quadratic program with equality constraints:*

$$\min \ \frac{1}{2}x^T Mx - b^T x \quad \textit{subject to} \quad A_{S^k} x = a^k, \tag{4.8}$$

where $a_i^k = l_i$ for $i \in J_l^k$, $a_i^k = u_i$ for $i \in J_u^k$, and $S^k := J_l^k \cup J_u^k$ with

$$J_l^k := \{i : (Ax(w^k) - \alpha w^k)_i < l_i\},$$
$$J_u^k := \{i : (Ax(w^k) - \alpha w^k)_i > u_i\}.$$

Proof. Let $S_{-1,i} := (-\infty, l_i]$, $S_{0,i} := [l_i, u_i]$, and $S_{1,i} := [u_i, \infty)$. For $\tau := (\tau_1, \ldots, \tau_m)$ with $\tau_i \in \{-1, 0, 1\}$, define

$$W_\tau := \{w \in \mathbb{R}^m : (Ax(w^k) - \alpha w^k)_i \in S_{\tau_i,i} \text{ for } 1 \le i \le s\}.$$

Then, $\varphi(w)$ is an affine mapping on W_τ for each τ. Recall that $\Phi'(w) = B\varphi(w)$ and $\Phi(w)$ is a convex quadratic function on each W_τ. By Lemma 3.7, there exists an integer k^* such that $W_\tau \cap W^* \ne \emptyset$ if $w^k \in W_\tau$ and $k \ge k^*$.

For $k \ge k^*$, let $\tau_i^k = -1$ if $i \in J_l^k$, $\tau_i^k = 1$ if $i \in J_u^k$, and $\tau_i^k = 0$ otherwise; then $w^k \in W_{\tau^k}$. Let $\hat{w}^k \in W_{\tau^k} \cap W^*$. Then, $x(\hat{w}^k) := M^{-1}(A^T\hat{w}^k + b)$ is the solution of (1.1) and $\varphi(\hat{w}^k) = 0$, which is equivalent to

$$Ax(\hat{w}^k) = (Ax(\hat{w}^k) - \alpha\hat{w}^k)_l^u. \tag{4.9}$$

Since $\hat{w}^k \in W_\tau$, $l_i \le (Ax(\hat{w}^k) - \alpha\hat{w}^k)_i \le u_i$ for $i \notin J_l^k \cup J_u^k$. By (4.9), we obtain $\hat{w}_i^k = 0$ for $i \notin (J_l^k \cup J_u^k)$. Moreover, if $i \in J_l^k$, then

$$(Ax(\hat{w}^k) - \alpha\hat{w}^k)_i \le l_i,$$

which implies $(Ax(\hat{w}^k))_i = l_i$. When $i \in J_u^k$, we have

$$(Ax(\hat{w}^k) - \alpha\hat{w}^k)_i \ge u_i,$$

which implies $(Ax(\hat{w}^k))_i = u_i$. Therefore, $A_{S^k}x(\hat{w}^k) = a^k$ and $x(\hat{w}^k) = M^{-1}(A_{S^k}^T\hat{w}_{S^k}^k + b)$. That is, $x(\hat{w}^k)$ is also the solution of (4.8). □

Now, we are ready to prove the finite termination of Algorithm 4.1.

THEOREM 4.5. *If M is positive definite and $l \le Ax \le u$ has a feasible solution, then Algorithm* 4.1 *produces the solution of* (1.1) *in finitely many operations (with exact arithmetic).*

Proof. By Lemmas 4.2 and 4.3, we know that $(\alpha I - D^J B)^{-1}\varphi(w)$ is well defined and z (or $(\hat{w} - w)$) is a descent direction for Φ at w. Therefore, Algorithm 4.1 is a descent method for solving (2.7).

Let t_k, J^k, w^k, z^k, S^k, a^k, and $\hat{w}^k$ denote t, J, w, z, S, a, and $\hat{w}$, respectively, produced by Algorithm 4.1 in the kth iteration.

We prove the finite termination of Algorithm 4.1 by contradiction. Here is an outline of the essential steps in our proof. The first step is to show that $\lim_{k\to\infty}\Phi(w^k) = \Phi_{\min}$. This follows from Lemma 3.4 if infinitely many iterates are generated by $w^{k+1} = w^k + t_k z^k$. The problem occurs if $w^{k+1} := w^k + t_k(\hat{w}^k - w^k)$ for k large enough. In this case, the restriction of step size t_k seems to be crucial, because it actually generates a bounded sequence $\{w^k\}$. The boundedness of $\{w^k\}$ allows us to prove $\lim_{k\to\infty}\|\Phi'(w^k)\| = 0$. By Lemma 3.3, we know that $\{w^k\}$ is a weakly convergent sequence (i.e., $\lim_{k\to\infty}\Phi(w^k) = \Phi_{\min}$). Therefore, by Lemmas 4.3 and 4.4, $x(\hat{w}^k)$ is the solution of (1.1) for k large enough. This indicates that Algorithm 4.1 should find a solution of (2.7) when k is large enough. The second step is to verify that $w^{k+1} := w^k + t_k(\hat{w}^k - w^k)$ does generate a solution of (2.7) when k is large enough.

Now assume the contrary, that Algorithm 4.1 generates an infinite sequence $\{w^k\}_1^\infty$.

Claim 1. $\lim_{k\to\infty} \Phi(w^k) = \Phi_{\min}$.

First, assume that there exists k_0 such that $w^{k+1} = w^k + t_k(\hat{w}^k - w^k)$ for $k \geq k_0$. Since there are only finitely many choices of a^k and J^k, there are finitely many distinct $\hat{w}^k$'s. Therefore, there exists a positive constant λ such that $\|\hat{w}^k\| \leq \lambda$. Hence,

$$\|w^{k+1}\| = \|(1-t_k)w^k + t_k\hat{w}^k\| \leq (1-t_k)\|w^k\| + t_k\lambda \leq \max\{\|w^k\|, \lambda\},$$

which implies

$$\|w^k\| \leq \max\{\|w^{k_0}\|, \lambda\} \quad \text{for } k \geq k_0. \tag{4.10}$$

If $(\hat{w}^k - w^k)^T\Phi'(w^{k+1}) \geq \frac{1}{2}(\hat{w}^k - w^k)^T\Phi'(w^k)$, by Lemma 3.1 there exists a positive constant γ such that

$$\left(\frac{(\hat{w}^k - w^k)^T\Phi'(w^k)}{\|\hat{w}^k - w^k\|}\right)^2 \leq \gamma(\Phi(w^k) - \Phi(w^{k+1})). \tag{4.11}$$

Since $\{\|\hat{w}^k - w^k\|\}$ is a bounded sequence (cf. (4.10)), there exists a positive constant κ such that $\|\hat{w}^k - w^k\| \leq \kappa$. This, along with (4.11), establishes the following estimate of $(w^k - \hat{w}^k)^T\Phi'(w^k)$:

$$(w^k - \hat{w}^k)^T\Phi'(w^k) \leq (\sqrt{\gamma}\kappa)\sqrt{\Phi(w^k) - \Phi(w^{k+1})}. \tag{4.12}$$

If $(\hat{w}^k - w^k)^T\Phi'(w^{k+1}) < \frac{1}{2}(\hat{w}^k - w^k)^T\Phi'(w^k) < 0$, then $t_k = 1$ and

$$g(\theta) := (\hat{w}^k - w^k)^T\Phi'(w^k + \theta(\hat{w}^k - w^k)) < \frac{1}{2}(\hat{w}^k - w^k)^T\Phi'(w^k) \quad \text{for } 0 \leq \theta \leq 1,$$

since $g(\theta)$ is a monotone function of θ. By the mean value theorem, there exists $0 < \theta_k < 1$ such that

$$\Phi(w^k) - \Phi(w^{k+1}) = -g(\theta_k) > -\frac{1}{2}(\hat{w}^k - w^k)^T\Phi'(w^k) = \frac{1}{2}(w^k - \hat{w}^k)^T\Phi'(w^k). \tag{4.13}$$

Since $(\alpha I - D^{S^k}B)(w^k - \hat{w}^k) = \varphi(w^k)$ (cf. Lemma 4.3) and $B\varphi(w^k) = \Phi'(w^k)$, we have

$$(\alpha B - BD^{S^k}B)(w^k - \hat{w}^k) = B\varphi(w^k) = \Phi'(w^k). \tag{4.14}$$

It follows from (4.12), (4.13), and (4.14) that for $k \geq k_0$,

$$\begin{aligned}&(w^k - \hat{w}^k)^T(\alpha B - BD^{S^k}B)(w^k - \hat{w}^k) \\ &\quad \leq (2 + \sqrt{\gamma}\kappa)\left(\sqrt{\Phi(w^k) - \Phi(w^{k+1})} + (\Phi(w^k) - \Phi(w^{k+1}))\right).\end{aligned}$$

Since $(\alpha B - BD^{S^k}B)$ is symmetric positive semidefinite,

$$y^T(\alpha B - BD^{S^k}B)y \geq \delta_k\|(\alpha B - BD^{S^k}B)y\|^2 \quad \text{for } y \in \mathbb{R}^m, \tag{4.15}$$

where $\delta_k \leq \|\alpha B - BD^{S^k}B\|^{-1}$ (cf. Lemma 3.1 in [30]). For easy reference, we include the proof here. For a symmetric positive semidefinite matrix Q, let $Q^{\frac{1}{2}}$ denote the square root of Q. Then,

$$\|Qy\|^2 = (Q^{\frac{1}{2}}y)^T Q(Q^{\frac{1}{2}}y) \leq \|Q\| \|Q^{\frac{1}{2}}y\|^2 = \|Q\| \cdot (y^T Qy).$$

Since there are only finitely many distinct S^k, we can choose $\delta_k \equiv \delta > 0$ such that (4.15) holds for all k. Thus, for $k \geq k_0$,

$$\begin{aligned} \|\Phi'(w^k)\|^2 &= \|(\alpha B - BD^{S^k}B)(w^k - \hat{w}^k)\|^2 \\ &\leq \frac{1}{\delta}(w^k - \hat{w}^k)^T(\alpha B - BD^{S^k}B)(w^k - \hat{w}^k) \\ &\leq \frac{1}{\delta}(2 + \sqrt{\gamma}\kappa)\left(\sqrt{\Phi(w^k) - \Phi(w^{k+1})} + (\Phi(w^k) - \Phi(w^{k+1}))\right). \end{aligned}$$

Since $\lim_{k\to\infty}(\Phi(w^k) - \Phi(w^{k+1})) = 0$, we obtain $\lim_{k\to\infty} \|\Phi'(w^k)\| = 0$. By Lemma 3.3, we have $\lim_{k\to\infty} \Phi(w^k) = \Phi_{\min}$.

If there is no k_0 such that $w^{k+1} = w^k + t_k(\hat{w}^k - w^k)$ for $k \geq k_0$, then there exists a subsequence $\{k_j\}$ such that $w^{k+1} = w^k + t_k z^k$ for $k = k_j$, $j = 1, 2, \ldots$. By the definition of $\Phi(w)$, $\Phi'(w) = B\varphi(w)$. By Lemma 4.2, $\alpha I - D^{J^k}B$ is nonsingular and $(\alpha B - BD^{J^k}B)^{-1}$ is positive definite. Since there are only finitely many different J^k's, $(w^{k+1} - w^k)^T\Phi'(w^{k+1}) = 0$, and $z^k = -(\alpha B - BD^{J^k}B)^{-1}\Phi'(w^k)$, it follows from Lemma 3.4 that $\lim_{k\to\infty} \Phi(w^k) = \Phi_{\min}$. This proves Claim 1.

By Claim 1 and Lemma 4.4, there exists an integer k^* such that x^* is the solution of (1.1) for $k \geq k^*$ if and only if x^* is the solution of the following strictly convex quadratic program with equality constraints:

$$\min_x \ \frac{1}{2}x^T Mx - b^T x \quad \text{subject to} \quad A_{S^k}x = a^k. \tag{4.16}$$

In particular, $A_{S^k}x = a^k$ is consistent. By Lemma 4.3, $A_{S^k}x(\hat{w}^k) = a^k$ and $w^{k+1} = w^k + t_k(\hat{w}^k - w^k)$ for $k \geq k^*$. Moreover, $x(\hat{w}^k)$ is the solution of (4.16) for $k \geq k^*$; hence, $x(\hat{w}^k) = x^*$ for $k \geq k^*$, where x^* is the solution of (1.1).

Now, let $k \geq k^*$. Since $w^{k+1} = (1 - t_k)w^k + t_k\hat{w}^k$, we obtain, for $i \notin S^k$, that

$$\begin{aligned} (Ax(w^{k+1}) - \alpha w^{k+1})_i &= (1 - t_k)(Ax(w^k) - \alpha w^k)_i + t_k(Ax(\hat{w}^k) - \alpha\hat{w}^k)_i \\ &= (1 - t_k)(Ax(w^k) - \alpha w^k)_i + t_k(Ax(\hat{w}^k))_i. \end{aligned} \tag{4.17}$$

By the definition of S^k, $l_i \leq (Ax(w^k) - \alpha w^k)_i \leq u_i$ for $i \notin S^k$. Since $l \leq Ax(\hat{w}^k) = Ax^* \leq u$ and $0 \leq t_k \leq 1$ (cf. step (4) in Algorithm 4.1) by (4.17) we have

$$l_i \leq (Ax(w^{k+1}) - \alpha w^{k+1})_i \leq u_i \quad \text{for } i \notin S^k.$$

Therefore, $S^{k+1} \subset S^k$ for $k \geq k^*$, and there exists k_0 such that $S^k = S^{k_0} =: S$ for $k \geq k_0$. For $i \in S$,

$$(a^{k+1})_i = (Ax(\hat{w}^{k+1}))_i = (Ax^*)_i = (Ax(\hat{w}^k))_i = (a^k)_i =: a_i.$$

Therefore, $((Ax(w^k) - \alpha w^k)_l^u)_i = a_i$ for $i \in S$ and $k \geq k_0$. Since $((Ax(w^k) - \alpha w^k)_l^u)_i = (Ax(w^k) - \alpha w^k)_i$ for $i \notin S$, we have, for $k \geq k_0$, that

$$\begin{aligned} \varphi(w^k)_i &= (\alpha w^k)_i && \text{for } i \notin S, \\ \varphi(w^k)_i &= (Ax(w^k))_i - a_i && \text{for } i \in S. \end{aligned} \tag{4.18}$$

Let $C = \alpha I - D^S B$ and a be the extension of a_S in $\mathbb{R}^m$ with $a_i = 0$ for $i \notin S$. Then, $\varphi(w^k) = Cw^k + a$ for $k \geq k_0$.

Now let $k = k_0$. If $t_k = 1$, $w^{k+1} = \hat{w}^k$. Since $\hat{w}_i = 0$ for $i \notin S$ and $(Ax(\hat{w}^k))_i = a_i$ for $i \in S$, $\varphi(w^{k+1}) = 0$ by (4.18), and Algorithm 4.1 should terminate after $(k+1)$ iterations. Therefore, $0 < t_k < 1$ and $(w^k - w^{k+1})^T B\varphi(w^{k+1}) = 0$. Thus,

$$\begin{aligned} t_k(w^k - \hat{w}^k)^T BC(w^k - \hat{w}^k) &= (w^k - w^{k+1})^T B(\varphi(w^k) + a - C\hat{w}^k) \\ &= (w^k - w^{k+1})^T B\varphi(w^k) \\ &= (w^k - w^{k+1})^T B(\varphi(w^k) - \varphi(w^{k+1})) \\ &= (w^k - w^{k+1})^T B((Cw^k + a) - (Cw^{k+1} + a)) \\ &= (w^k - w^{k+1})^T BC(w^k - w^{k+1}) \\ &= t_k^2(w^k - \hat{w}^k)^T BC(w^k - \hat{w}^k), \end{aligned}$$

where the second equality follows from $(C\hat{w}^k)_i = 0$ for $i \notin S$ and also $(C\hat{w}^k)_i = (Ax(\hat{w}^k))_i = a_i$ for $i \in S$; the third is due to the choice of t_k: $(\hat{w}^k - w^k)^T B\varphi(w^{k+1}) = 0$, and the remaining ones are easy consequences of $w^{k+1} = w^k + t_k(\hat{w}^k - w^k)$ and $\varphi(w^k) = Cw^k - a$. Since $0 < t_k < 1$, the above identity implies $(w^k - \hat{w}^k)^T BC(w^k - \hat{w}^k) = 0$. Since BC is symmetric positive semidefinite (cf. Lemma 4.2 (3)), $BC(w^k - \hat{w}^k) = 0$. By (4.14), $B\varphi(w^k) = 0$, and Algorithm 4.1 should again terminate after k iterations. This completes the proof of Theorem 4.5. □

Remark. Since there is no dual feasibility requirement, $x(\hat{w}^k) = x^*$ does not necessarily imply that $\hat{w}^k$ is a solution of (2.7) (i.e., $\varphi(\hat{w}^k) = 0$). Even when there exists $\hat{w}$ such that $\hat{w}_i = 0$ for $i \notin S^k$ and $\varphi(\hat{w}) = 0$, we do not know how to choose J^k such that $\varphi(\hat{w}^k) = 0$. Fortunately, the above proof shows that the algorithm has a mechanism to prevent the "wrong indices" from entering the current active set S^k. In order to understand this mechanism, let us explore more carefully what happens when $w^{k+1} = w^k + t_k(\hat{w}^k - w^k)$ and $x(\hat{w}^k) = x^*$. If there exists $\hat{w}$ such that $\hat{w}_i = 0$ for $i \notin S^k$ and $\varphi(\hat{w}) = 0$, then $A_{S^k}x(\hat{w}^k) = A_{S^k}x^* = a^k$. Therefore, $A_{S^k}x(\hat{w}^k) = a^k$ indicates that a suitable choice of J^k might yield a solution for the equation $\varphi(w) = 0$. However, due to redundancy in $A_{S^k}x = a^k$, the algorithm might choose J^k that produces $\hat{w}^k$ with $\varphi(w^{k+1}) \neq 0$. A natural recovery procedure is to drop some redundant constraints whose indices are in S^k. This is automatically done through the dual descent method. Note that the proof of Theorem 4.5 shows that $S^{k+1} \subset S^k$. If $S^k = S^{k+1}$ and $a^k = a^{k+1}$, the proof of Theorem 4.5 yields $\varphi(w^{k+1}) = 0$. Therefore, we have either $S^{k+1} \neq S^k$ or $a^k \neq a^{k+1}$. If $S^{k+1} \neq S^k$, then some constraints whose indices are in S^k will not be treated as the current active constraints in the next iteration. Therefore, there is less chance of choosing "wrong indices" in the next iteration. However, if $S^k = S^{k+1}$ but $a^{k+1} \neq a^k$, then the algorithm accomplishes nothing in terms of identifying active constraints. Our proof shows that this can only happen if w^k is still far away from the solution set. Since the descent method guarantees that $\lim_{k\to\infty} \operatorname{dist}(w^k, W^*) = 0$, we can not have $S^k = S^{k+1}$ and $a^{k+1} \neq a^k$ for k large enough. In summary, the algorithm first identifies a set of active constraints by reducing the distance from the current iterate to the solution set and then starts dropping redundant constraints until a dual solution is found.

5. Numerical results.

Reformulations of convex quadratic programs as problems of unconstrained minimization of convex quadratic splines allow one to develop new efficient algorithms for solving the original quadratic programs. In this section we provide numerical evidence for such a statement by comparing QPspline with QPROG (an active-set method) in the IMSL library [18]. The numerical experiments are done with respect to the following two classes of quadratic programs: strictly convex quadratic programs with simple bound constraints (which are generated as in [35]) and least-squares problems with linearly independent two-sided inequality constraints, which are generated by a variation of Moré and Toraldo's method [35]. The results for quadratic programs with simple bound constraints are included in Tables 1–3 and the results for least-squares problems are included in Tables 4–6. We compare two performance measures: accuracy (the maximum deviation from the exact solution) and efficiency (CPU time in seconds).

All numerical results were done in double precision on an IBM RS 6000/590 computer running IBM AIX Version 3.2 (UNIX System V) operating system.

We used a straightforward implementation of QPspline for solving the least-squares problems, where the systems of equations with matrices $A_J M^{-1} A_J^T$ were solved by a Cholesky factorization. The code is a slight modification of the code written for the algorithm for solving the convex regression problem (cf. [27]). For strictly convex quadratic programs with simple bound constraints, we used a straightforward implementation of the Newton method for finding the minimizer of $\Psi(y)$ with $(y = x)$, where the Newton direction was also computed by a Cholesky factorization. There is no special measure being taken to deal with possible ill-conditioning of linear systems when we solve for Newton directions. For quadratic problems with simple bound constraints we start with the unconstrained minimizer of the objective function, while we start with $w = 0$ for least-squares problems. In other words, we use the same initial guess as QPROG. We stop the algorithm when $\|y - (Ey + q)_l^u\|_\infty$ or $\|Ax(w) - (Ax(w) - \alpha w)_l^u\|_\infty$ is less than 10^{-10}.

Sometimes, we might get a negative stepsize t_k that could be the result of errors in the computation of either the Newton direction or the stepsize. If the computed stepsize is wrong but the Newton direction is accurate, which happens quite often when the current iterate is very close to the exact solution, then we could get the exact solution by replacing t_k by 1. This strategy has proven to be very effective in finding the exact solutions of our test problems (cf. [27]). However, if the computed Newton direction is actually an ascending direction due to a nearly singular matrix $A_J M^{-1} A_J^T$, then replacing t_k by 1 will produce an iterate that might be far away from the exact solution. As a consequence, the algorithm has to use more iterations to get an iterate that is close to the exact solution again. This is the reason why we see a surge in CPU time for solving some test problems.

For the randomly generated problems with 100 variables, our algorithm is significantly faster than QPROG and is almost as accurate as QPROG. Note that, for very ill-conditioned problems (with condition number 10^9–10^{12}), a solution produced by QPROG might have 2–3 more accurate decimal places than a solution found by QPspline. We believe that this is the effect of a numerical error caused by the Cholesky factorization of a very ill-conditioned positive definite matrix. However, the worst maximum deviation of our solutions from the exact optimal solutions is 10^{-3}, which is comparable with the worst maximum deviation of QPROG's solutions from the exact optimal solutions, 10^{-4}.

It is important to note that QPROG becomes slower in finding the solution when the number of active constraints increases from 10 to 90 (cf. Tables 1–3), while QPspline is not sensitive to the number of active constraints. In fact, in most cases, QPspline takes less time to find the solution when the number of active constraints is

Table 1

Number of Variables			Active Constraints		
100			10		
Accuracy		CPU Time			
Newton	IMSL	Newton	IMSL	Condition	Degeneracy
.57E−13	.16E−13	0.02	0.05	10^3	10^{-3}
.21E−12	.12E−13	0.02	0.17	10^3	10^{-6}
.12E−12	.16E−13	0.01	0.25	10^3	10^{-9}
.87E−13	.26E−13	0.02	0.06	10^3	10^{-12}
.20E−12	.83E−14	0.03	0.15	10^6	10^{-3}
.35E−13	.30E−14	0.02	0.25	10^6	10^{-6}
.64E−13	.46E−14	0.03	0.05	10^6	10^{-9}
.13E−12	.15E−13	0.03	0.20	10^6	10^{-12}
.54E−13	.53E−15	0.02	0.27	10^9	10^{-3}
.16E−12	.19E−13	0.05	0.07	10^9	10^{-6}
.17E−12	.53E−14	0.05	0.16	10^9	10^{-9}
.23E−13	.16E−14	0.02	0.25	10^9	10^{-12}
.86E−10	.42E−11	0.02	0.17	10^{12}	10^{-3}
.11E−09	.21E−10	0.03	0.23	10^{12}	10^{-6}
.26E−11	.27E−13	0.03	0.34	10^{12}	10^{-9}
.62E−10	.44E−11	0.04	0.15	10^{12}	10^{-12}

Table 2

Number of Variables			Active Constraints		
100			50		
Accuracy		CPU Time			
Newton	IMSL	Newton	IMSL	Condition	Degeneracy
.53E−10	.96E−12	0.04	0.21	10^3	10^{-3}
.51E−10	.30E−11	0.01	0.32	10^3	10^{-6}
.98E−10	.63E−11	0.15	0.14	10^3	10^{-9}
.98E−10	.14E−11	0.08	0.22	10^3	10^{-12}
.53E−10	.16E−11	0.03	0.31	10^6	10^{-3}
.26E−09	.87E−11	0.05	0.13	10^6	10^{-6}
.16E−09	.59E−11	0.07	0.24	10^6	10^{-9}
.18E−10	.98E−12	0.02	0.34	10^6	10^{-12}
.43E−07	.19E−08	0.02	0.22	10^9	10^{-3}
.13E−07	.50E−08	0.06	0.22	10^9	10^{-6}
.42E−07	.18E−10	0.05	0.50	10^9	10^{-9}
.76E−07	.11E−08	0.05	0.36	10^9	10^{-12}
.39E−07	.56E−08	0.05	0.27	10^{12}	10^{-3}
.10E−06	.52E−09	0.06	0.36	10^{12}	10^{-6}
.87E−07	.33E−08	0.05	0.32	10^{12}	10^{-9}
.19E−06	.23E−08	0.09	0.31	10^{12}	10^{-12}

90 than when it is 50 (cf. Tables 2 and 3).

In the remainder of this section, we describe how the test problems and the entries in the tables are generated. One can find some missing technical details in [35].

First consider the following strictly convex quadratic program with simple bound constraints:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}x^TMx - b^Tx \quad \text{subject to} \quad l \le x \le u. \tag{5.1}$$

Our test problems are generated in the same way as described in [35]. The test results are included in Tables 1–3. (See Tables 10–12 in [35] for numerical results of Moré and Toraldo's active-set method on an Alliant FX/8.)

The positive definite matrix $M = YDY$, where Y is a randomly generated orthog-

Table 3

Number of Variables	Active Constraints
100	90

Accuracy		CPU Time			
Newton	IMSL	Newton	IMSL	Condition	Degeneracy
.10E−09	.26E−11	0.05	0.37	10^3	10^{-3}
.42E−07	.95E−08	0.02	0.21	10^3	10^{-6}
.14E−06	.57E−08	0.08	0.40	10^3	10^{-9}
.36E−09	.20E−10	0.05	0.39	10^3	10^{-12}
.30E−04	.34E−05	0.01	0.38	10^6	10^{-3}
.56E−04	.11E−05	0.08	0.37	10^6	10^{-6}
.38E−05	.74E−08	0.05	0.34	10^6	10^{-9}
.92E−04	.53E−05	0.08	0.18	10^6	10^{-12}
.14E−03	.22E−05	0.09	0.34	10^9	10^{-3}
.14E−08	.46E−10	0.07	0.56	10^9	10^{-6}
.94E−04	.33E−06	0.10	0.36	10^9	10^{-9}
.76E−04	.15E−05	0.12	0.33	10^9	10^{-12}
.45E−04	.11E−05	0.05	0.58	10^{12}	10^{-3}
.61E−04	.19E−05	0.03	0.26	10^{12}	10^{-6}
.48E−04	.12E−04	0.10	0.39	10^{12}	10^{-9}
.91E−05	.46E−07	0.04	0.35	10^{12}	10^{-12}

onal Householder matrix and the matrix D is a diagonal matrix whose ith component d_i is defined by

$$\log d_i = \left(\frac{i-1}{n-1}\right) \cdot ncond, \quad i = 1, \ldots, n. \tag{5.2}$$

Note that the condition number of M is 10^{ncond}, which is listed in the tables under "Condition." The exact solution x^* is generated with components randomly in the interval $(-1, 1)$. For a given number nax of active constraints, we randomly generate a subset J of $\{1, \ldots, n\}$ with nax indices.

For the active set J, we use a parameter $ndeg$ to generate the Lagrange multiplier y:

$$|y_i| = 10^{-\mu_i \cdot ndeg}, \quad i \in J,$$

where μ_i is randomly generated in the interval (0,1). We list 10^{-ndeg} in the tables under "Degeneracy," which shows the amount of "numerical degeneracy" of a problem. For all the test problems, the number of variables $n = 100$.

With randomly generated M, x^*, y and the active set J, we define $b = Mx^* - y$ and

$$l_i = -1, \quad u_i = 1, \quad y_i = 0, \quad i \notin J$$

and

$$l_i = x_i^*, \quad u_i = 1, \quad y_i > 0$$

or

$$l_i = -1, \quad u_i = x_i^*, \quad y_i < 0.$$

The accuracy is measured by the ℓ_∞ norm of $\bar{x} - x^*$, $\|\bar{x} - x^*\|_\infty$, where $\bar{x}$ is a solution generated either by QPspline or QPROG in the IMSL library.

Table 4

Number of Variables		Condition Number		Degeneracy	
100		10^3		10^{-3}	
Accuracy		CPU Time		Constraints	
Newton	IMSL	Newton	IMSL	Total	Active
.11E−13	.66E−14	0.00	0.04	10	5
.24E−13	.70E−14	0.00	0.04	10	10
.84E−13	.24E−13	0.01	0.07	50	25
.42E−13	.28E−13	0.03	0.13	50	50
.59E−13	.44E−14	0.05	0.14	90	45
.11E−12	.17E−13	0.11	0.23	90	90

Table 5

Number of Variables		Condition Number		Degeneracy	
100		10^6		10^{-6}	
Accuracy		CPU Time		Constraints	
Newton	IMSL	Newton	IMSL	Total	Active
.11E−07	.17E−11	0.01	0.03	10	5
.11E−10	.99E−11	0.00	0.04	10	10
.16E−05	.64E−11	0.01	0.08	50	25
.51E−10	.11E−10	0.02	0.13	50	50
.45E−06	.16E−10	0.03	0.15	90	45
.41E−06	.81E−11	0.20	0.23	90	90

The next set of test problems is the following least-squares problem with two-sided inequality constraints:

$$\min_{x \in \mathbb{R}^n} \frac{1}{2}\|x - b\|_2^2 \quad \text{subject to} \quad l \le Ax \le u,$$

where A is an $m \times n$ matrix with rank m. We use a variation of Moré and Toraldo's method to generate test problems. As before, we choose $n = 100$, which is the number of variables listed in Tables 4–6. We choose $m = 10, 50$, and 90, which are listed under "Total Constraints" in Tables 4–6.

For given m and n, we randomly generate two orthogonal Householder matrices U and V of dimensions $m \times m$ and $n \times n$, respectively, as in [35]. Then we randomly generate an $m \times n$ matrix D whose entries are zeros (except for the diagonal entries that are generated by (5.2)). The constraint matrix $A := UDV$. Note that the condition number of A is 10^{ncond}, which is listed in the tables under "Condition Number."

Similarly, for a given number nax, we randomly generate a subset J of $\{1, \dots, m\}$ with nax indices. The exact solution x^* and its Lagrange multiplier y are generated as before. Here, we choose $ndeg \equiv ncond$. The amount of degeneracy, 10^{-ncond}, is listed under "Degeneracy."

Then we define $d = x^* + A^T y$ and

$$l_i = -2|(Ax^*)_i|, \quad u_i = 2|(Ax^*)_i|, \quad y_i = 0, \quad i \notin J,$$

and

$$l_i = (Ax^*)_i, \quad u_i = 2|(Ax^*)_i|, \qquad y_i > 0,$$

or

$$l_i = -2|(Ax^*)_i|, \quad u_i = (Ax^*)_i, \qquad y_i < 0.$$

Table 6

Number of Variables	Condition Number	Degeneracy
100	10^9	10^{-9}

Accuracy		CPU Time		Constraints	
Newton	IMSL	Newton	IMSL	Total	Active
.79E−10	.16E−09	0.00	0.03	10	5
.77E−10	.16E−09	0.01	0.04	10	10
.66E−06	.11E−09	0.01	0.07	50	25
.34E−07	.68E−09	0.01	0.16	50	50
.13E−06	.14E−08	0.03	0.15	90	45
.10E−06	.32E−08	0.06	0.23	90	90

The accuracy is measured by $\|\bar{x} - x^*\|_\infty$ as before.

6. Comments. Our main purpose is to establish simple and practical unconstrained reformulations of convex quadratic programs. In [27], we proposed a Newton method with line search for solving strictly convex quadratic programs with linearly independent constraints. The Newton method is an iterative method but terminates in a finite number of iterations. In this case, one would expect that the Newton method finds the unique minimizer of the strictly convex quadratic spline $\Phi(w)$ in a finite number of iterations, since the Newton method automatically identifies the unique minimizer of $\Phi(w)$ in the next iteration once the current iterate is in a solution region [27]. Without the assumption of linear independence of constraints, $\Phi(w)$ is not strictly convex and the Hessian of $\Phi(w)$ (if it exists) might be singular. A fundamental question related to unconstrained minimization of $\Phi(w)$ is whether or not one can design a finite algorithm to find a minimizer of $\Phi(w)$. The main purpose of QPspline is to provide a positive answer to this question. Based on sections 3 and 4, one might be able to design a descent method that finds a minimizer of any convex quadratic spline, which is bounded below on $\mathbb{R}^m$, in a finite number of iterations.

The proposed algorithm, QPspline, has a flavor of interior-point methods for solving linear programs, which have two phases: (1) reduction of a merit function (or a potential function) and (2) projection of the current iterate to the nearest vertex of the feasible region. Without steps (3) and (4), QPspline is a descent method that reduces the value of the merit function $\Phi(w)$. Steps (3) and (4) are the projection process to find a solution of (1.1). As mentioned before, we can find the descent direction and the projection by factorizing one nonsingular matrix $A_J M^{-1} A_J^T$.

The most significant feature of QPspline is that there is no requirement on feasibility. Note that QPspline is based on the dual unconstrained reformulation of (1.1) and that there is no primal or dual feasibility requirement. This has a great advantage in practical implementations of QPspline. For example, we can use any algorithm for finding a "good" approximate solution of (2.5) and use the approximate solution as the starting point of QPspline. In this way, we can first find a cheap approximate solution near the solution region and then use QPspline to obtain an exact solution in a few iterations. In particular, in the early stages of the computations we could skip steps (2) through (5) and simply compute the descent direction $z = -\varphi(w)$ to get the next iterate. Once the current w is "close" enough to the solution region, we can start to use the Newton directions to get an exact solution.

Note that it is very easy to design linearly convergent descent algorithms to find a minimizer of the convex quadratic spline function $\Phi(w)$, even if the set of all minimizers of $\Phi(w)$ is unbounded [21]. We can also use conjugate-gradient methods to find an approximate solution for the minimization of $\Phi(w)$ [22]. Another simple approach is to use the proximal-point algorithm [43] to find an approximate minimizer of the convex quadratic spline $\Phi(w)$. Here, for each subproblem, we only need to find the unique minimizer of a strictly convex quadratic spline, which easily can be computed by a Newton method [27] or a conjugate-gradient method [22].

Given the absence of any feasibility requirement, QPspline is ideal for solving a sequence of closely related strictly convex quadratic programs where a solution of the current quadratic program can be used as a good initial guess of the next one (cf. [25]). For example, suppose that we have two quadratic programs (1.1) with $M = M^i, A = A^i, l = l^i$, and $u = u^i$ for $i = 1, 2$. Let x^i and y^i be primal and dual solutions, respectively. If M^2, A^2, l^2, u^2 are very close to M^1, A^1, l^1, u^1, respectively, then we could expect that (x^2, y^2) is very close to (x^1, y^1). Therefore, if we start QPspline at y^1, then the algorithm should find y^2 in a few iterations. Note that all classical active-set methods require either primal or dual feasibility and, in general, one cannot directly use x^1 or y^1 as a starting point to find x^2 or y^2. This makes QPspline an ideal subroutine for sequential programming techniques (cf. [9]).

The unconstrained reformulations of quadratic programs given in this paper are closely related to augmented Lagrangian functions. Through augmented Lagrangian functions, we can get similar unconstrained reformulations for more general quadratic programs (cf. [23], [24]).

Our numerical results (cf. Tables 1–6) indicate that Newton methods for solving quadratic programs through their unconstrained reformulations are faster than QPROG (an active-set method) when applied to the two classes of randomly generated test problems in section 5. (Note that only for the third test problem listed in Table 2 is QPROG a little faster than QPspline.) However, the current implementation of QPspline is not as accurate as QPROG when the computation of a Newton direction involves a nearly singular matrix $A_J M^{-1} A_J^T$. The key issue of improving the performance of QPspline seems to be an intelligent choice of J. One simple remedy is to adopt Goldfarb and Idnani's strategy of selecting indices of "most violated constraints" among S with some singularity detection mechanism in the computation of a Newton direction. We should continue to study strategies for computing a Newton direction stably and efficiently. For large-scale problems especially, it is undesirable to solve a system of equations with matrix $A_J M^{-1} A_J^T$ by a direct matrix factorization in each iteration. An important implementation issue is whether or not matrix updating techniques can be used to find the Newton direction that requires the solution of systems of equations with matrices $A_J M^{-1} A_J^T$.

To conclude, it should be observed that QPspline has a flaw: lack of finite termination when the quadratic programming problem has no feasible solution. We do not know whether or not it is possible to incorporate a strategy into Algorithm 4.1 for the detection of infeasible problems.

Acknowledgment. The authors would like to thank N. Gould and M. Saunders for a thorough reading of the manuscript and many helpful comments that improved the presentation.

REFERENCES

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

[2] T. F. Coleman and L. A. Hulbert, *A globally and superlinearly convergent algorithm for convex quadratic programs with simple bounds*, SIAM J. Optim., 3 (1993), pp. 298–321.

[3] 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.

[4] I. C. Demetriou and M. J. D. Powell, *The minimum sum of squares change to univariate data that gives convexity*, IMA J. Numer. Anal., 11 (1991), pp. 433–448.

[5] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1983.

[6] G. Di Pillo and L. Grippo, *Exact penalty functions in constrained optimization*, SIAM J. Control Optim., 27 (1989), pp. 1333–1360.

[7] B. C. Eaves, *On the basic theorem of complementarity*, Math. Programming, 1 (1971), pp. 68–75.

[8] A. S. El-Bakry, R. A. Tapia, and Y. Zhang, *A study of indicators for identifying zero variables in interior-point methods*, SIAM Rev., 36 (1994), pp. 45–72.

[9] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley & Sons, Inc., New York, 1968.

[10] M. Frank and P. Wolfe, *An algorithm for quadratic programming*, Naval Res. Logist., 3 (1956), pp. 95–110.

[11] A. Friedlander and J. M. Martínez, *On the maximization of a concave quadratic function with box constraints*, SIAM J. Optim., 4 (1994), pp. 177–192.

[12] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, New York, 1981.

[13] T. Glad and E. Polak, *A multiplier method with automatic limitation of penalty growth*, Math. Programming, 17 (1979), pp. 140–155.

[14] D. Goldfarb and A. Idnani, *A numerically stable dual method for solving strictly convex quadratic programs*, Math. Programming, 27 (1983), pp. 1–33.

[15] L. Grippo and S. Lucidi, *On the solution of a class of quadratic programs using a differentiable exact penalty function*, in System Modelling and Optimization, H. J. Sebastian and K. Tammer, eds., Leipzig, 1989, pp. 764–773, and Lecture Notes in Control and Information Science 143, Springer, Berlin, 1990.

[16] L. Grippo and S. Lucidi, *A differentiable exact penalty function for bound constrained quadratic programming problems*, Optimization, 22 (1991), pp. 557–578.

[17] P. T. Harker and J.-S. Pang, *Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications*, Math. Programming, 48 (1990), pp. 161–220.

[18] *IMSL* 32*-Bit Fortran Numerical Libraries*, Visual Numerics, Inc., Houston, TX, 1994.

[19] W. Li, *Error bounds for piecewise convex quadratic programs and applications*, SIAM J. Control Optim., 33 (1995), pp. 1510–1529.

[20] W. Li, *Remarks on convergence of the matrix splitting algorithm for the symmetric linear complementarity problem*, SIAM J. Optim., 3 (1993), pp. 155–163.

[21] W. Li, *Linearly convergent descent methods for unconstrained minimization of a convex quadratic spline*, J. Optim. Theory Appl., 86 (1995), pp. 145–172.

[22] W. Li, *A conjugate gradient method for the unconstrained minimization of strictly convex quadratic splines*, Math. Programming, 72 (1996), pp. 17–32.

[23] W. Li, *Differentiable piecewise quadratic exact penalty functions for quadratic programs with simple bound constraints*, SIAM J. Optim., 6 (1996), pp. 299–315.

[24] W. Li, *Differentiable Piecewise Quadratic Exact Penalty Functions for Convex Quadratic Programs with Linearly Independent Two-Sided Inequality Constraints*, Preprint, Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA, June 1995.

[25] W. Li, D. Naik, and J. Swetits, *A data smoothing technique for piecewise convex/concave curves*, SIAM J. Sci. Comput., 17 (1996), pp. 517–537.

[26] W. Li, P. Pardalos, and C. G. Han, *Gauss-Seidel method for least distance problems*, J. Optim. Theory Appl., 75 (1992), pp. 487–500.

[27] W. Li and J. Swetits, *A Newton method for convex regression, data smoothing, and quadratic programming with bounded constraints*, SIAM J. Optim., 3 (1993), pp. 466–488.

[28] Y. Y. Lin and J. S. Pang, *Iterative methods for large quadratic programs: A survey*, SIAM J. Control Optim., 25 (1987), pp. 383–411.

[29] Z.-Q. Luo, *Convergence analysis of primal-dual interior point algorithms for convex quadratic programs*, in Recent Trends in Optimization Theory and Applications, World Sci. Ser. Appl. Anal. 5, World Sci. Publishing, River Edge, NJ, 1995, pp. 255–270.

[30] X.-D. Luo and Z.-Q. Luo, *Extension of Hoffman's error bound to polynomial systems*, SIAM J. Optim., 4 (1994), pp. 383–392.

[31] Z.-Q. Luo and P. Tseng, *Error bound and convergence analysis of matrix splitting algorithms*

for the affine variational inequality problem, SIAM J. Optim., 2 (1992), pp. 43–54.

[32] Z.-Q. Luo and P. Tseng, *On the convergence of a matrix splitting algorithm for the symmetric monotone linear complementarity problem*, SIAM J. Control Optim., 29 (1991), pp. 1037–1060.

[33] O. L. Mangasarian, *Solution of symmetric linear complementarity problems by iterative methods*, J. Optim. Theory Appl., 22 (1977), pp. 465–485.

[34] O. L. Mangasarian, *Convergence of iterates of an inexact matrix splitting algorithm for the symmetric monotone linear complementarity problem*, SIAM J. Optim., 1 (1991), pp. 114–122.

[35] J. J. Moré and G. Toraldo, *On the solution of large quadratic programming problems with bound constraints*, SIAM J. Optim., 1 (1991), pp. 93–113.

[36] J. J. Moré and S. J. Wright, *Optimization Software Guide*, SIAM, Philadelphia, PA, 1993.

[37] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM, Philadelphia, PA, 1994.

[38] J. M. Ortega and W. C. Rheinboldt, *Iterative Solutions of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[39] J. S. Pang, *Methods for quadratic programming: A survey*, Computers Chem. Engineering, 7 (1983), pp. 583–594.

[40] P. M. Pardalos and N. Kovoor, *An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds*, Math. Programming, 46 (1990), pp. 321–328.

[41] M. J. D. Powell, *On the quadratic programming algorithm of Goldfarb and Idnani*, Math. Programming Study, 25 (1985), pp. 46–61.

[42] S. M. Robinson, *Some continuity properties of polyhedral multifunctions*, Math. Programming Study, 14 (1981), pp. 206–214.

[43] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.

[44] R. J. Vanderbei and T. J. Carpenter, *Symmetric indefinite systems for interior point methods*, Math. Programming, 58 (1993), pp. 1–32.

[45] P. Wolfe, *Convergence conditions for ascent methods*, SIAM Rev., 11 (1969), pp. 226–235.

[46] E. K. Yang and J. W. Tolle, *A class of methods for solving large, convex quadratic program subject to box constraints*, Math. Programming, 51 (1991), pp. 223–228.

SIAM J. Optim.
Vol. 7, No. 3, pp. 620–640, August 1997

AN INFEASIBLE-INTERIOR-POINT METHOD FOR LINEAR COMPLEMENTARITY PROBLEMS*

EVANGELIA M. SIMANTIRAKI† AND DAVID F. SHANNO‡

Abstract. In this work we present an infeasible-interior-point algorithm which is based on a method for the general nonlinear programming problem to solve linear complementarity problems. For this algorithm, we prove global convergence from any strictly positive starting point, under minor assumptions. Numerical results are reported which demonstrate very good computational performance on large-scale linear complementarity problems.

Key words. linear complementarity, interior-point methods

AMS subject classifications. 65K05, 90C33

PII. S1052623495282882

1. Introduction.

The linear complementarity problem (LCP) determines a vector pair (x, z) satisfying

$$\begin{aligned} Mx - c &= z, \\ x^T z &= 0, \\ (x, z) &\geq \mathbf{0}, \end{aligned} \tag{1}$$

where $x, z, c \in \Re^n$ and $M \in \Re^n \times \Re^n$. LCPs arise in many areas, such as quadratic programming, bimatrix games, variational inequalities, and economic equilibria problems, and they have been the subject of much research interest. A number of direct as well as iterative methods have been proposed for their solution. The book by Cottle, Pang, and Stone [1] is a good reference for pivoting methods developed to solve LCPs. Another important class of methods used to tackle LCPs are the interior-point and infeasible-interior-point methods, which were first designed to solve linear programs (see [9], [14], [19]). Most of these methods were developed to solve monotone LCPs, i.e., problems in which the matrix M is positive semidefinite (see, for example, [25], [27]). However, much recent research has been devoted to interior-point methods for nonmonotone LCPs (see [10], [13], [18], [20], [21], [26]).

The method presented in this paper is an infeasible-interior-point method developed for solving the general nonmonotone LCP, and it is a modification of a method devised by El-Bakry et al. [4] to solve the general nonlinear programming problem.

Before presenting the algorithm, it is instructive to see how the logarithmic-barrier method can be applied to problem (1) and provide the basis for the algorithm.

Problem (1) can be formulated as the minimization problem

$$\text{minimize } x^T z$$

* Received by the editors March 10, 1995; accepted for publication (in revised form) February 24, 1996. 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 right. This research was sponsored by the Air Force Office of Scientific Research, Air Force System Command under grant F49620-95-1-0110.

http://www.siam.org/journals/siopt/7-3/28288.html

† Rutcor and Graduate School of Management, Rutgers University, New Brunswick, NJ 07102 (vms@rutcor.rutgers.edu).

‡ Rutgers Center for Operations Research, Rutgers University, New Brunswick, NJ 08903 (shanno@tucker.rutgers.edu).

$$\begin{aligned} \text{subject to } & Mx - z = c, \\ & (x,z) \geq \mathbf{0}. \end{aligned} \tag{2}$$

Applying the logarithmic-barrier method to (2) yields

$$\begin{aligned} \text{minimize } & x^T z - \mu \sum \log(x_i) - \mu \sum \log(z_i) \\ \text{subject to } & Mx - z = c, \\ & (x,z) \geq \mathbf{0}, \end{aligned} \tag{3}$$

and if we denote by $\mathcal{L}(x, z, \lambda, \mu)$ the Lagrangian for (3), we have that

$$\mathcal{L}(x, z, \lambda, \mu) = x^T z - \mu \sum \log(x_i) - \mu \sum \log(z_i) - \lambda^T (Mx - z - c). \tag{4}$$

The first-order conditions for (4) yield the system of nonlinear equations

$$z - \mu X^{-1} e - M^T \lambda = \mathbf{0}, \tag{5}$$

$$x - \mu Z^{-1} e + \lambda = \mathbf{0}, \tag{6}$$

$$Mx - z = c, \tag{7}$$

which, with some manipulation and assuming that the matrix $XM^T + Z$ is nonsingular, reduces to the system

$$F(x,z) = \begin{bmatrix} Mx - z - c \\ XZe \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mu e \end{bmatrix}, \tag{8}$$

where $X = \text{diag}(x_i)$, $Z = \text{diag}(z_i)$, and e is a vector of all ones. Sufficient conditions for the matrix $XM^T + Z$ to be nonsingular can be found in [1].

The paper is organized as follows. In section 2 we give a thorough description of the algorithm. In section 3 we prove global convergence of our algorithm under only two assumptions, namely, that the matrix $XM + Z$ remains nonsingular at every iteration and the matrix M satisfies that if $|x_i^k| \to +\infty$ for $i \in J \subset \{1, 2, \ldots, n\}$, then there exists a $j \in J$ such that $|[Mx^k]_j| \to +\infty$ also. The second assumption is used to guarantee that the iterates remain bounded. In section 4 we present numerical results on several problems found in the literature. Some randomly generated single commodity spatial equilibrium problems were also considered. Finally, section 5 contains some concluding remarks and comments.

Throughout this paper, subscripts were used to denote iterations on scalars and superscripts to denote iterations on vectors and matrices. We write e for the vector of all ones, $\mathbf{0}$ for the vector of all zeros, and I for the identity matrix of suitable dimension. Unless otherwise stated, the symbol $\|\cdot\|$ denotes the Euclidean norm of a vector.

2. The algorithm.

The algorithm moves from the current estimate $v^k = (x^k, z^k)$ to the solution of (1) to a new estimate $v^{k+1} = (x^{k+1}, z^{k+1})$ by

$$\begin{aligned} x^{k+1} &= x^k + \alpha_k \Delta x^k, \\ z^{k+1} &= z^k + \alpha_k \Delta z^k, \end{aligned} \tag{9}$$

where Δx^k, Δz^k, solve the system

$$M\Delta x^k - \Delta z^k = c - Mx^k + z^k, \tag{10}$$

$$Z^k\Delta x^k + X^k\Delta z^k = \mu_k e - X^k Z^k e. \tag{11}$$

Multiplying (10) by X^k and adding it to (11) yields

$$\begin{aligned} \Delta x^k &= (X^k M + Z^k)^{-1}(X^k c - X^k M x^k + \mu_k e), \\ \Delta z^k &= M\Delta x^k + Mx^k - z^k - c. \end{aligned} \tag{12}$$

It is easy to check that system (12) is the result of applying one Newton step to the system of equations (8), and thus the method belongs to the general framework of centered and damped Newton methods. For any X^k and Z^k with strictly positive elements, if the matrix $(X^k M + Z^k)$ were to be factored it would have the same sparsity pattern as M, assuming that the diagonal entries of M are nonzero. Therefore, sparse LU factorizations can be utilized when M is sparse.

To start the algorithm, a strictly positive starting point (x^0, z^0) is required. Then, by controlling the step length α_k, the algorithm generates strictly positive iterates in every step.

The algorithm moves from iterate to iterate seeking to minimize the merit function

$$\phi(v) = \phi(x, z) = \{\ \|XZe\|^2 + \|Mx - z - c\|^2\ \}^{1/2} = \|F(x, z)\| \tag{13}$$

and terminates when $\phi(x, z) \leq \epsilon$ for some predetermined ϵ.

In order to fully describe a step between successive iterates of the algorithm, we need to say how μ_k and α_k are selected. Regarding μ_k, we adopted a typical selection (see, for example, [25] and [27]), which has proven to be good for both practical and theoretical purposes, namely,

$$\mu_k = \sigma_k \frac{{x^k}^T z^k}{n}, \quad \sigma_k \in (0, 1). \tag{14}$$

To specify the selection of α_k, we introduce the following quantities:

$$\tau_1 = \frac{\min\ x_i^0 z_i^0}{\frac{{x^0}^T z^0}{n}}, \tag{15}$$

$$\tau_2 = \frac{{x^0}^T z^0}{\|Mx^0 - z^0 - c\|}, \tag{16}$$

and the functions (see [4])

$$f^I(\alpha) = \min\ {x_i}^{k+1} {z_i}^{k+1} - \gamma\tau_1 \frac{{x^{k+1}}^T z^{k+1}}{n}, \tag{17}$$

$$f^{II}(\alpha) = {x^{k+1}}^T z^{k+1} - \gamma\tau_2 \|Mx^{k+1} - z^{k+1} - c\|, \tag{18}$$

where $\gamma \in (0, 1)$. First, we find $\hat{\alpha}_k$ to be the largest number $\in (0, 1]$ for which

$$f^I(\alpha) \geq 0 \quad \text{and} \quad f^{II}(\alpha) \geq 0 \quad \forall \alpha \in (0, \hat{\alpha}_k] \subset (0, 1] \tag{19}$$

and then perform backtracking on the merit function to find α_k. Specifically,

$$\alpha_k = \rho^t \hat{\alpha}_k,$$

where t is the smallest nonnegative integer such that α_k satisfies

$$\phi(v^{k+1}) \leq \phi(v^k) + \alpha_k \beta \nabla\phi(v^k)^T \Delta v^k \,, \tag{20}$$

where $\beta \in (0, 1/2]$ and $\rho \in (0,1)$.

Clearly, the way $\hat{\alpha_k}$ is selected in (19) guarantees that $(x^{k+1}, z^{k+1}) > 0$, and therefore no further conditions on α_k need be imposed. For a proof of the existence of $\hat{\alpha_k}$, see [27] and [28].

It is not hard to show (see [27]) that the following is true:

$$Mx^{k+1} - z^{k+1} - c = (1-\alpha_k)(Mx^k - z^k - c) = \nu_{k+1}(Mx^0 - z^0 - c), \tag{21}$$

where $\nu_0 = 1$ and

$$\nu_{k+1} = (1-\alpha_k)\nu_k = \prod_{j=0}^{k}(1-\alpha_j) > 0.$$

Hence, condition $f^{II}(\alpha) \geq 0$ is equivalent to

$$x^{k+1^T} z^{k+1} \geq \gamma(1-\alpha_k)\nu_k x^{0^T} z^0,$$

which is the same condition as the one used by Zhang in [27], but Zhang requires $\gamma = 1$. The function in (17) is a piecewise quadratic, and the resulting condition is commonly used in interior-point methods as a centering condition that prevents the iterates from approaching zero prematurely. Since γ can be chosen to be very small, our requirement for centrality is mild. The function in (18) is a quadratic, and it is used to ensure that feasibility is given a higher priority than complementarity (see also [25] and [27]). The complete algorithm is presented in Figure 1.

3. Proof of convergence. In this section we prove that the method presented in this paper is globally convergent. Specifically, starting from any strictly positive point (x^0, z^0), the algorithm will converge to a solution of the LCP provided that a solution exists and under the assumptions that the matrix $X^kM + Z^k$ which is used in the calculation of the new pair of iterates (x^{k+1}, z^{k+1}) stays nonsingular at every step and that the matrix M satisfies that if $|x_i^k| \to +\infty$ for $i \in J \subset \{1, 2, \ldots, n\}$, then there exists a $j \in J$ such that $|[Mx^k]_j| \to +\infty$ also. This latter assumption is used to guarantee that the iterates remain bounded. Computational experience with infeasible LCPs revealed that in such instances the matrix X^kM+Z^k became singular and, consequently, $\|\Delta v^k\|_\infty$ became arbitrarily large, the sequence of steplengths $\{a_k\}$ tended to zero rapidly, and the algorithm halted.

The proof of convergence of the algorithm is a modification of the proof given by El-Bakry et al. in [4] for the general nonlinear programming problem. We first show that the perturbed Newton direction Δv^k generated by the algorithm is a descent direction for the merit function defined in (13). As was pointed out earlier (see (8), (10), and (11)), the search direction satisfies

$$\Delta v^k = F'(v^k)^{-1}[-F(v^k) + \mu_k \hat{e}] \,, \tag{22}$$

where $\hat{e}$ is the vector $(\mathbf{0}, e)^T$. The gradient of the merit function is

$$\nabla\phi(v^k) = \frac{1}{\|F(v^k)\|} F'(v^k)^T F(v^k).$$

```
Let $k = 0$
Choose $v^0 = (x^0, z^0) > 0,\ \rho \in (0,1),\ \beta \in (0, 1/2], \gamma \in (0,1)$,
and compute $\phi(x^0, z^0)$.

1.    If $\phi^k \leq \epsilon$  STOP

          $(x^k, z^k)$ is an approximate solution to the LCP.

2.    Choose $\sigma_k \in (0,1)$, let $\mu_k = \sigma_k \frac{{x^k}^T z^k}{n}$, and find

          $\Delta v^k = (\Delta x^k\ , \ \Delta z^k)$ from  (12).

3.    Compute $\alpha_k$ as described in (19) and (20).

          If $\alpha_k \leq 10^{-10}$  STOP

4.    Find $(x^{k+1}, z^{k+1})$ by

          $x^{k+1} = x^k + \alpha_k \Delta x^k\,,$
          $z^{k+1} = z^k + \alpha_k \Delta z^k\,.$

5. Set $k = k + 1$ and return to 1.
```

FIG. 1. *Infeasible-interior-point method.*

Hence,

$$\begin{aligned}
\nabla\phi(v^k)^T \Delta v^k &= \frac{1}{\|F(v^k)\|} F(v^k)^T F'(v^k) F'(v^k)^{-1} [-F(v^k) + \mu_k \hat{e}] \\
&= \frac{1}{\|F(v^k)\|} (-\|F(v^k)\|^2 + \mu_k {x^k}^T z^k) \\
&= -\left(\|F(v^k)\| - \mu_k \frac{{x^k}^T z^k}{\|F(v^k)\|} \right) \\
&= -\left(\phi(v^k) - \mu_k \frac{{x^k}^T z^k}{\phi(v^k)} \right).
\end{aligned}$$

Since $\mu_k = \sigma_k \frac{{x^k}^T z^k}{n}$,

$$\begin{aligned}
\mu_k {x^k}^T z^k = \sigma_k \frac{({x^k}^T z^k)^2}{n} &= \sigma_k \left(\frac{\|X^k Z^k e\|_1}{\sqrt{n}} \right)^2 \\
&\leq \sigma_k \|X^k Z^k e\|^2 \leq \sigma_k \|F(v^k)\|^2 = \sigma_k \phi(v^k)^2.
\end{aligned}$$

Therefore,

$$\nabla\phi(v^k)^T \Delta v^k \leq -\phi(v^k)(1 - \sigma_k) \leq 0.$$

Moreover, from the backtracking line search performed on the merit function (see (20)), we have that

$$\phi(v^{k+1}) \leq [1 - \alpha_k \beta (1 - \sigma_k)]\, \phi(v^k), \tag{23}$$

where $\beta \in (0, 1/2]$.

The sequence $\{\phi(v_k)\}$ is monotone nonincreasing and bounded from below; thus it is convergent. Furthermore, (23) asserts that $\{\phi(v^k)\}$ converges to zero Q-linearly if $\{\alpha_k\}$ is bounded away from zero and $\{\sigma_k\}$ is bounded away from one.

Following the notation used by El-Bakry et al. in [4], let us define for any given $\epsilon > 0$, and for a fixed $\gamma \in (0,1)$, the set

$$\Omega(\epsilon) \equiv \left\{ (x,z) : \epsilon \leq \phi(x,z) \leq \phi_0\, , \; \frac{\min\; x_i z_i}{x^T z/n} \geq \gamma \tau_1\, , \; \frac{x^T z}{\|Mx - z - c\|} \geq \gamma \tau_2 \right\}.$$

For this set the following observations are in order:

1. $\Omega(\epsilon)$ is a closed set.
2. In $\Omega(\epsilon)$ where $\epsilon > 0$, all components of XZe are bounded above and away from zero, and, consequently, $x^T z$ is bounded above and away from zero.
3. The sequence $\{(x^k, z^k)\}$ generated by the algorithm satisfies $\{(x^k, z^k)\} \subset \Omega(0)$.

To prove the convergence of the algorithm, we will make the following assumptions.

Assumption 1. The matrix $(X^k M + Z^k)$ is invertible for any pair of iterates (x^k, z^k) such that $(x^k, z^k) \in \Omega(\epsilon)$, with $\epsilon > 0$.

Note that in section 1, when we applied the logarithmic-barrier method to obtain system 8, we assumed that the matrix $XM^T + Z$ is nonsingular. However, it is obvious that $\det(XM^T + Z) \neq 0$ is equivalent to $\det\,(XM + Z) \neq 0$, so Assumption 1 is also sufficient for the purposes of section 1.

Remark. The common assumption of many authors, namely, the positive semidefiniteness of the matrix M, guarantees that our assumption will be true. However, we do not require M to be positive semidefinite. Our algorithm has performed well on LCPs with nonsymmetric indefinite matrices. The next lemma actually shows that assuming that $XM + Z$ is nonsingular is equivalent to requiring that M belong in the P_0-class of matrices, which is larger than the class of positive semidefinite matrices. By definition, a matrix M belongs in the P_0-class if and only if all its principal minors are nonnegative.

LEMMA 1. *The matrix $XM + Z$ is nonsingular for any diagonal matrices X and Z with strictly positive elements if and only if M is a P_0-matrix.*

Proof. It is easy to show that for any strictly positive X and Z the following is true:

$$(XM + Z) \text{ nonsingular } \Leftrightarrow \begin{pmatrix} -M & I \\ Z & X \end{pmatrix} \text{ nonsingular }. \tag{24}$$

Moreover, Kojima et al. have shown in [10] that it also holds that

$$\begin{pmatrix} -M & I \\ Z & X \end{pmatrix} \text{ nonsingular } \Leftrightarrow M \in P_0. \tag{25}$$

Hence, from (24) and (25) it directly follows that

$$(XM + Z) \text{ nonsingular } \Leftrightarrow M \in P_0.$$

The P_0-class includes many important matrices such as positive semidefinite matrices, P-matrices, P_*-matrices, etc. (see [10]) that give rise to interesting and difficult-to-solve complementarity problems. Moreover, for the LCPs with P_0-matrices, it is not true that if the problem is strictly feasible then it has a solution. This holds for LCPs with P_*-matrices. An example of an infeasible P_0-matrix LCP is studied in section 4.

We wish to emphasize here that our algorithm only requires that the matrix $X^kM + Z^k$ be nonsingular for X^k, Z^k diagonal matrices containing the iterates $\{(x^k, z^k)\}$, and it was actually successfully tested on matrices with no special structure of any kind.

In addition to Assumption 1, we need to make the following assumption in order to guarantee that the iterates generated by the algorithm remain bounded.

Assumption 2. Let $J = \{i : |x_i^k| \to +\infty \text{ as } k \to +\infty\}$, $J \subset \{1, 2, \ldots, n\}$. Then there exists $j \in J$ such that $|[M(x^k)]_j| \to +\infty$.

Assumption 2 is sufficient to guarantee that the sequence $\{x^k\}$ generated by the algorithm is bounded, and, consequently, the merit function $\phi(x, z)$ has bounded level sets. This will become clear by the following two lemmas.

Lemma 2. *If $\|x^k\|_\infty \le \omega_1$ for $\omega_1 > 0$ sufficiently large, then $\exists\ \omega_2 > 0 \ni \|z^k\|_\infty \le \omega_2$. Consequently, $\exists\ \omega > 0$ such that*

$$\|(x^k, z^k)\|_\infty \le \omega.$$

Proof. The boundedness of $\{x^k\}$ implies that there exists $K_1 \ge 0$ such that the sequence $\{\|Mx^k - c\|\}$ is bounded above by K_1. Then,

$$\|z^k\| \le \|Mx^k - c - z^k\| + \|Mx^k - c\| \le \phi_0 + K_1 = \omega_2.$$

Moreover, $\|(x^k, z^k)\|_\infty \le \max\{\omega_1, \omega_2\} = \omega$.

Lemma 3. *The set*

$$S \equiv \{(x, z) : \epsilon \le \phi(x, z) \le \phi_0\} = \{(x, z) : \epsilon \le \{\|XZe\|^2 + \|Mx - c - z\|^2\}^{1/2} \le \phi_0\}$$

is bounded. Moreover, the generated sequence of iterates $\{(x^k, z^k)\}$ is bounded and the set $\Omega(\epsilon)$ is bounded.

Proof. From Lemma 2 it is clear that we only need to show that the sequence $\{x^k\}$ remains bounded. Assume on the contrary that $\|x^k\| \to \infty$ and let $J = \{i : \{x_i^k\} \text{ is unbounded}\}$. From Assumption 2 we have that $\exists\ i \in J$ such that $|[M(x^k)]_i| \to +\infty$. Furthermore, for this i we must have that $z_i^k \to 0$, since otherwise $x_i^k z_i^k \to \infty$ and $\|X^kZ^ke\|^2 \to \infty$. But if $x_i^k \to +\infty$ and $z_i^k \to 0$, then $\|[Mx^k]_i - c_i - z_i^k\| \to +\infty$ and $\|Mx^k - c - z^k\|^{\ 2} \to +\infty$. Hence, $\{x^k\}$ remains bounded, and from Lemma 2 the sequence $\{(x^k, z^k)\}$ is bounded.

Another condition sufficient to guaranteeing that the generated sequence of iterates is bounded is the assumption that the matrix M is a P-matrix; i.e., all its principal minors are positive. For a P-matrix it holds that there exists a constant $\gamma(M) > 0$ such that

$$\max_{1 \le i \le n} y_i[My]_i \ge c\|y\|^2 \ \forall x, y \in \mathbf{R}^n.$$

Using this property, it can be shown that Assumption 2 is satisfied (see [8]). Imposing such a condition on the matrix M is quite restrictive. It is known that an LCP with a P-matrix has a unique solution for every $c \in \mathbf{R}^n$. Note, however, that this condition is only sufficient to guarantee the boundedness and is not necessary. The algorithm was actually successfully tested on problems with matrices that do not have the P-matrix property.

Alternatively, we can assume that M is positive semidefinite. Actually, positive semidefiniteness of the matrix could be replaced for the purpose of boundedness by the weaker assumption

$$(x-\hat{x})^T(z-\hat{z}) \geq 0 \; \forall (x,z) \, \text{and} \, \forall (\hat{x},\hat{y}) \text{ such that } Mx-c=z \text{ and } M\hat{x}-c=\hat{z}.$$

The sequence $\{(x^k,z^k)\}$ can be proven to be bounded again in this case (see [28]).

To prove convergence, we first show that the direction Δv^k generated by the algorithm is uniformly bounded over the set $\Omega(\epsilon)$.

LEMMA 4. *If $\{v^k\} = \{(x^k,z^k)\} \in \Omega(\epsilon)$ with $\epsilon > 0$, then $\{[F'(v^k)]^{-1}\}$ is bounded and, furthermore, the Newton direction Δv^k is uniformly bounded over the set $\Omega(\epsilon)$.*

Proof. $F(v^k) = F(x^k,z^k)$ is given by

$$F(x,z) = \begin{bmatrix} Mx-z-c \\ XZe \end{bmatrix}$$

and, hence,

$$F'(x,z) = \begin{bmatrix} M & -I \\ Z & X \end{bmatrix}.$$

It then easily can be verified that

$$\begin{aligned} [F'(v^k)]^{-1} &= [F'(x^k,z^k)]^{-1} \\ &= \begin{bmatrix} (Z^k+X^kM)^{-1}X^k & (Z^k+X^kM)^{-1} \\ M(Z^k+X^kM)^{-1}X^k-I & M(Z^k+X^kM)^{-1} \end{bmatrix}. \end{aligned} \tag{26}$$

From the assumption on the matrix X^kM+Z^k and the fact that the iterates $\{(x^k,z^k)\}$ remain bounded in $\Omega(\epsilon)$, it follows that $[F'(v^k)]^{-1}$ is well defined and bounded over the set $\Omega(\epsilon)$, since every component involved in the right-hand side of (26) is well defined and bounded. Consequently, the search direction Δv^k, defined by (22), is a continuous function of the location $v^k = (x^k,z^k)$ and therefore also will be bounded in $\Omega(\epsilon)$.

In the next lemma we show that as long as the sequence v^k satisfies $v^k \in \Omega(\epsilon)$ for any given $\epsilon > 0$, then the sequence of steplengths $\{\hat{\alpha}_k\}$ defined by (19) is bounded away from zero.

LEMMA 5. *If $\{(x^k,z^k)\} \subset \Omega(\epsilon)$ and $\{\sigma_k\}$ is bounded away from zero, then $\{\hat{\alpha}_k\}$ is bounded away from* 0.

Proof. Let

$$\alpha^i = \max_{\alpha \in [0,1]} \{\alpha : f^i(\acute{\alpha}) \geq 0 \text{ for all } \acute{\alpha} \leq \alpha\}, \; i = I, II. \tag{27}$$

Then, clearly,

$$\hat{\alpha}_k = \min\{\alpha^I, \alpha^{II}\},$$

and to prove the lemma, it suffices to show that both α^I and α^{II} are bounded away from zero. We have

$$\begin{aligned} f^I(\alpha) &= \min({x_i}^k + \alpha \Delta {x_i}^k)({z_i}^k + \alpha \Delta {z_i}^k) - \gamma\tau_1 \frac{(x^k + \alpha\Delta x^k)^T (z^k + \alpha \Delta z^k)}{n} \\ &= (1-\alpha) \underbrace{\left({x_i}^k {z_i}^k - \gamma\tau_1 \frac{{x^k}^T z^k}{n} \right)}_{\geq 0} + (1-\gamma\tau_1)\mu_k \alpha \\ &+ \left(\Delta {x_i}^k \Delta {z_i}^k - \gamma\tau_1 \frac{\Delta {x^k}^T \Delta z^k}{n} \right) \alpha^2 \\ &\geq (1-\gamma\tau_1)\mu_k\alpha - \left\| \Delta {x_i}^k \Delta {z_i}^k - \gamma\tau_1 \frac{\Delta {x^k}^T \Delta z^k}{n} \right\| \alpha^2 \\ &\geq (1-\gamma\tau_1)\mu_k\alpha - B_1\alpha^2, \end{aligned}$$

where B_1 is a positive constant that satisfies

$$\left\| \Delta {x_i}^k \Delta {z_i}^k - \gamma\tau_1 \frac{\Delta {x^k}^T \Delta z^k}{n} \right\| \leq B_1.$$

Such a constant will exist, since $\Delta x^k, \Delta z^k$ are bounded in $\Omega(\epsilon)$ (see Lemma 4). Thus, from the definition of α^I (see (27)), we clearly have

$$\alpha^I \geq \frac{(1-\gamma\tau_1)\mu_k}{B_1}. \tag{28}$$

Similarly,

$$\begin{aligned} f^{II}(\alpha) &= (x^k + \alpha\Delta x^k)^T (z^k + \alpha\Delta z^k) - \gamma\tau_2 \| M(x^k + \alpha\Delta x^k) - (z^k + \alpha\Delta z^k) - c \| \\ &= (x^k + \alpha\Delta x^k)^T (z^k + \alpha\Delta z^k) - \gamma\tau_2 (1-\alpha) \| Mx^k - z^k - c\| \\ &= (1-\alpha) \underbrace{({x^k}^T z^k - \gamma\tau_2 \|Mx^k - z^k - c\|)}_{\geq 0} + n\mu_k\alpha + \Delta {x^k}^T \Delta z^k \alpha^2 \\ &\geq n\mu_k\alpha - \|\Delta {x^k}^T \Delta z^k\| \alpha^2 \\ &\geq n\mu_k\alpha - B_2\alpha^2, \end{aligned}$$

where B_2 is a positive constant such that

$$\|\Delta {x^k}^T \Delta z^k\| \leq B_2.$$

Using (27), we conclude that

$$\alpha^{II} \geq \frac{n\mu_k}{B_2}. \tag{29}$$

If σ_k is bounded away from zero, then $\mu_k = \sigma_k \frac{{x^k}^T z^k}{n}$ is bounded below in $\Omega(\epsilon)$, and it follows from (28) and (29) that α^I, α^{II} are both bounded below and, moreover, that the sequence $\hat{\alpha_k}$, $k = 1, \ldots$ is bounded below in $\Omega(\epsilon)$.

We can now prove the main convergence result, namely, that the sequence generated by the algorithm is globally convergent if Assumptions 1 and 2 hold.

THEOREM 1. *Let the sequence* $\{v^k\}$ *be generated by the algorithm displayed in Figure* 1. *Then for any* $\epsilon > 0$ *and* $\{\sigma_k\} \subset (0,1)$ *bounded away from zero and one,*

$$\exists\, k^* \ \ni \ \ \phi(v^k) \le \epsilon \ \ \forall k > k^*, \tag{30}$$

i.e., $\phi(v^k)$ *converges to zero.*

Proof. We have already seen that the sequence $\{\phi(v^k)\}$ is convergent. To prove the theorem, let us suppose that (30) is not true; i.e.,

$$\exists\, \hat\epsilon > 0 \ \ni \ \phi(v^k) > \hat\epsilon \ \ \forall k. \tag{31}$$

Then, $\{v^k\} \subset \Omega(\hat\epsilon)$.

If in infinitely many iterations $\alpha_k = \hat\alpha_k$, i.e., backtracking is not invoked, then from the inequality

$$\frac{\phi(v^{k+1})}{\phi(v^k)} \le 1 - \hat\alpha_k \beta (1 - \sigma_k)$$

and since $\hat\alpha_k$ are bounded away from zero (see Lemma 5), it follows that the corresponding subsequence converges to zero Q-linearly, which contradicts (31). Now assume that $\alpha_k < \hat\alpha_k$ in infinitely many iterations. In this case, the backtracking line search used in the algorithm produces a subsequence for which

$$\frac{\nabla\phi(v^k)^T \Delta v^k}{\|\Delta v^k\|} = \frac{-\left(\phi(v^k) - \mu_k \dfrac{{x^k}^T z^k}{\phi(v^k)}\right)}{\|\Delta v^k\|} \to 0\,. \tag{32}$$

A proof of this result can be found in [3].

Since $\|\Delta v^k\|$ is bounded above in $\Omega(\hat\epsilon)$ (see Lemma 4), (32) implies

$$\phi(v^k) - \mu_k \frac{{x^k}^T z^k}{\phi(v^k)} \to 0\,.$$

However,

$$(1 - \sigma_k)\phi(v^k) \le \phi(v^k) - \mu_k \frac{{x^k}^T z^k}{\phi(v^k)}\,,$$

therefore, it must hold that $\phi(v^k) \to 0$ because σ_k is bounded away from one. This again contradicts (31). Thus, (30) must be true, and this proves the theorem.

To establish the convergence result presented in this section we need to guarantee that the sequence of iterates $\{x^k\}$ remains bounded. Assumption 2 was used for this purpose. If we drop the boundedness requirement on the iterates, it is easy to modify the theorem and show that in this case the sequence of iterates either converges to a solution of the LCP or becomes unbounded, i.e., $\|x^k\| \to +\infty$ as $k \to +\infty$.

Kojima, Noma, and Yoshise [11] have actually proved that for the general nonlinear monotone complementarity problem and for a generic interior-point method satisfying certain conditions, three cases may occur. Either the sequence of iterates will converge to a solution of the complementarity problem in a finite number of steps, or it will converge to an approximately feasible point that is not complementary and from which a solution of the complementarity problem can be found in a finite number of steps, or, finally, there exists a region which contains no solution of the complementarity problem. Their theorem follows. Note that we only state the result here. The conditions that the algorithmic mapping should satisfy can be found in the above reference and are all met by our algorithm.

Let ϵ be any small positive number and $\mathcal{M}$ be any positive number. Then there exists a finite number p such that one of the following holds:

- ${x^p}^T z^p < \epsilon$ and $\|z^p - Mx^p + c\| < \epsilon$.
- $z^0 - Mx^0 + c \neq 0$ and $\|z^p - Mx^p + c\| < \epsilon$.
- $z^0 - Mx^0 + c \neq 0$, $\|z^p - Mx^p + c\| > \epsilon$, and $\nu_p r^T x^p - (x^P)^T z^p \geq \nu_p \mathcal{M}$.
  In this case the region $T(\mathcal{M}) = \{v = (x, z) \in \mathbf{R}^n_+ \times \mathbf{R}^n_+ : r^T x < \mathcal{M}\}$ contains no solution of the LCP.

Here $r = z^0 - Mx^0 + c$, and

$$\nu_0 = \begin{cases} 1 & \text{if } r \neq 0, \\ 0 & \text{if } r = 0, \end{cases}$$
$$\nu_{k+1} = (1 - a_k)\nu_k, \quad k = 1, 2, \ldots.$$

Note that from relation (21) we have that

$$(z^k - Mx^k + c) = \nu_k(z^0 - Mx^0 + c) = \nu_k r.$$

As Kojima, Noma, and Yoshise pointed out in [11], any given bounded subset of $\mathbf{R}^2_{++}$ is contained by the set $T(\mathcal{M})$ if $\mathcal{M}$ is sufficiently large.

This result is actually consistent with our theorem. As we have already noted, under the positive semidefiniteness assumption it can be shown that both Assumptions 1 and 2 hold, and thus the algorithm will converge provided that a solution exists.

4. Computational experience.

All the experiments presented in this section were performed on a SPARCstation 5 with 32 Mbytes RAM, and the codes were written in FORTRAN 77.

As we pointed out in section 2, when M is sparse, sparse LU factorization methods can be utilized. For the examples demonstrated in this section, the package Y12 from NETLIB was used interchangeably with the package UMFPACK [2] for nonsymmetric matrices.

4.1. Selection of parameters.

In choosing the algorithmic parameters, we followed the suggestions of El-Bakry et al. [4]. We select σ_k as

$$\sigma_k = \min\{\eta_1, \eta_2 {x^k}^T z^k\},$$

where $\eta_1 = .02$ and $\eta_2 = 1$. Clearly, σ_k is bounded away from one and in $\Omega(\epsilon), \epsilon > 0$, it is also bounded away from zero.

In [4] it was suggested that $\gamma \geq 1/2$, but our experience revealed that the algorithm performed better with a smaller γ. Theoretically, γ can be chosen to be very small as long as it is independent of n. In the numerical examples presented in this section, we took $\gamma = 6 \times 10^{-5}$. For the backtracking line search we used $\rho = 0.5$ and $\beta = 10^{-4}$.

The stopping criterion was

$$\phi(x^k, z^k) \leq \epsilon = 10^{-8}.$$

To start the algorithm, we used (x^0, z^0) with elements

$$x_i{}^0 = z_i{}^0 = \beta \ \ \forall i = 1, \ldots, n,$$

where $\beta > 0$ is arbitrarily chosen. For the purpose of comparison, results are provided with $\beta = 1, 10, 100$, and 1000 for most test problems. Although this choice of (x^0, z^0) may seem simplistic, it proved to work well in practice.

A strategy suggested by Fernandes, Júdice, and Patrício in [5] was also implemented and tested on some of the problems. With this strategy, the starting point is found as

$$\begin{aligned} x_i{}^0 &= \beta, \\ z_i{}^0 &= \|Mx - c\|_\infty, \ \ i = 1, \ldots, n, \end{aligned} \tag{33}$$

where β is a small integer $(1 \leq \beta \leq 5)$.

The differences observed when applying the method on the same problem with various starting points suggest that the selection of the starting point is important for the algorithm, and a general algorithm remains for further study.

4.2. Numerical results. The algorithm was tested on a number of problems found in the literature including both small- and large-scale LCPs. Some results with randomly generated single commodity spatial economic equilibrium problems are also provided. In all instances the algorithm performed extremely well and proved to be insensitive to the dimension of the problem. Two small infeasible problems are also included to demonstrate the behavior of the algorithm in such cases.

The first two problems were taken from Hock and Schittkowski [7] and were cited by Shanno in [22]. The matrix M, the vector c, and the solution x^* for each one of these problems are given next.

Problem I.

$$M = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 0 & 1 \\ 1 & 0 & 1 & 2 \\ -1 & -1 & -2 & 0 \end{pmatrix},$$

$$c = (8,\ 6,\ 4,\ -3)^T.$$

The solution of this LCP is $x^* = (2.5,\ 0.5,\ 0,\ 2.5)$.

Problem II.

$$M = \begin{pmatrix} 1 & 0 & -0.5 & 0 & 1 & 3 & 0 \\ 0 & 0.5 & 0 & 0 & 2 & 1 & -1 \\ -0.5 & 0 & 1 & 0.5 & 1 & 2 & -4 \\ 0 & 0 & 0.5 & 0.5 & 1 & -1 & 0 \\ -1 & -2 & -1 & -1 & 0 & 0 & 0 \\ -3 & -1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$c = (1,\ 3,\ -1,\ 1,\ -5,\ -4,\ 1.5)^T.$$

The solution of this LCP is $x^* = (0.09,\ 2.36,\ 0,\ 0.18,\ 0.9,\ 0,\ 0)$.

Problem III. This problem was taken from [15] and also appeared in [22]. Its matrix M and vector c are

$$M = \begin{pmatrix} 0 & 0 & 2 & 2 & 1 \\ 0 & 0 & 1 & 2 & 2 \\ 1 & 2 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \\ 2 & 3 & 0 & 0 & 0 \end{pmatrix},$$

$$c = (1,\ 1,\ 1,\ 1,\ 1)^T.$$

The solution of this LCP satisfies $x^* = (\alpha, \beta, 0, 0.5, 0)$, where indeterminacy gives different α and β (satisfying $3\alpha + \beta - 1 = 0$) for different starting points.

The results for these three problems with four different starting points are presented in Table 1.

Table 1

Problems I, II, III: *Iterations of the algorithm.*

Starting point	Problem I	Problem II	Problem III
${x_i}^0 = {z_i}^0 = 1$	7	10	6
${x_i}^0 = {z_i}^0 = 10$	8	12	9
${x_i}^0 = {z_i}^0 = 100$	11	15	13
${x_i}^0 = {z_i}^0 = 1000$	12	15	12

Problem IV. This test problem was taken from [6] and has also been cited in [22] and [5]. The matrix M of the problem satisfies $M = LL^T$, where L is a dense lower triangular matrix with diagonal elements equal to one and off-diagonal elements equal to two. Thus, M is as follows:

$$M = \begin{pmatrix} 1 & 2 & 2 & \cdots & 2 \\ 2 & 5 & 6 & \cdots & 6 \\ 2 & 6 & 9 & \cdots & 10 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 2 & 6 & 10 & \cdots & 4(n-1)+1 \end{pmatrix},$$

$$c = (1, \ldots, 1)^T.$$

The solution of the problem is $x^* = (1,\ 0,\ \ldots,\ 0)$.

Although it is difficult to compare different algorithms fairly, our algorithm performed better, in terms of the number of iterations it took to converge, than the predictor–corrector algorithm implemented and tested on the same problem in [5]. Table 2 illustrates the performance of the algorithm for four values of the dimension n with seven different starting points.

Clearly, the performance of the method does not appear to depend on the dimension of the LCP. Fernandes, Júdice, and Patrício pointed out in [5] that their computational experiments revealed that block pivoting algorithms do not share this

TABLE 2
Problem IV*: Iterations of the algorithm.*

	Dimension			
Starting point	50	100	200	300
${x_i}^0 = {z_i}^0 = 1$	5	5	5	5
${x_i}^0 = {z_i}^0 = 10$	10	10	10	10
${x_i}^0 = {z_i}^0 = 100$	13	14	14	14
${x_i}^0 = {z_i}^0 = 1000$	17	17	18	18
${x_i}^0 = 1, {z_i}^0 = \|Mx^0 - c\|_\infty$	19	19	19	19
${x_i}^0 = 4, {z_i}^0 = \|Mx^0 - c\|_\infty$	21	21	21	21
${x_i}^0 = 5, {z_i}^0 = \|Mx^0 - c\|_\infty$	22	22	22	22

property. On all problems tested, however, it appears to be a property of this algorithm.

Problem V. This problem was taken from [1]. M is the upper triangular matrix

$$M = \begin{pmatrix} 1 & 2 & 2 & \cdots & 2 \\ 0 & 1 & 2 & \cdots & 2 \\ 0 & 0 & 1 & \cdots & 2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 2 \end{pmatrix},$$

$$c = (1, \ldots, 1)^T.$$

The solution of the problem is $x^* = (0, 0, \ldots, 1)$.

We tested the algorithm with four different starting points. The results in number of iterations for three different values of the dimension n are illustrated in Table 3.

TABLE 3
Problem V*: Iterations of the algorithm.*

	Dimension		
Starting point	10	100	200
${x_i}^0 = {z_i}^0 = 1$	4	4	4
${x_i}^0 = {z_i}^0 = 10$	9	9	9
${x_i}^0 = {z_i}^0 = 100$	12	12	12
${x_i}^0 = {z_i}^0 = 1000$	15	15	15

Problem VI. Here we considered two LCPs taken from [1]. The matrix M for both of the LCPs is simply the transpose of the matrix we had in the previous problem, but the vectors c are quite different this time.

$$M = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 2 & 1 & 0 & \cdots & 0 \\ 2 & 2 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 2 & 2 & 2 & \cdots & 1 \end{pmatrix}.$$

For the first problem we have that the vector c has elements

$$c_i = \sum_{j=i}^{n} 2^j,$$

and the solution of the problem is

$$x^* = \left(\sum_{j=1}^{n} 2^j,\ 0,\ \ldots,\ 0 \right),$$

while in the second problem c satisfies

$$c_i = \sum_{j=n+1-i}^{n} 2^j,$$

and the solution of the resulting LCP is

$$x^* = (2^n,\ 0,\ \ldots,\ 0)\,.$$

Scaling was required for these problems because of the large values of the elements of c. Therefore, for these two experiments we took $x_i{}^0 = z_i{}^0 = \max\,(1, \frac{\|c\|_2}{n})$.

Table 4 displays the performance of the algorithm on both problems for $n = 20$.

TABLE 4
Problem VI: *Iterations of the algorithm.*

c	Iterations
$c_i = \sum_{j=i}^{n} 2^j$	15
$c_i = \sum_{j=n+1-i}^{n} 2^j$	19

Problem VII. This problem was taken from [5]. Both degenerate and nondegenerate problems were constructed sharing the same pentadiagonal matrix M with elements

$$\begin{aligned} m_{i,i} &= 6, \\ m_{i,i-1} &= m_{i-1,i} = -4, \\ m_{i,i-2} &= m_{i-2,i} = 1. \end{aligned} \tag{34}$$

For the nondegenerate problem, the vector c was constructed as in [5] in such a way that the solution x^* of the resulting LCP satisfies

$$\begin{aligned} x_i{}^* &= 1,\ z_i{}^* = 0, i = 1, \ldots, \frac{n}{2}, \\ x_i{}^* &= 0,\ z_i{}^* = 1, i = \frac{n}{2} + 1, \ldots, n. \end{aligned}$$

The degenerate problem was constructed so that its solution satisfies

$$\begin{aligned} x_i{}^* &= z_i{}^* = 0,\ i = 1, \ldots, \frac{n}{8}, \\ x_i{}^* &= 1,\ z_i{}^* = 0,\ i = \frac{n}{8} + 1, \ldots, \frac{n}{2}, \\ x_i{}^* &= 0,\ z_i{}^* = 1,\ i = \frac{n}{2} + 1, \ldots, n. \end{aligned}$$

Table 5 displays the performance of our algorithm on both types of problems for four different values of n. For the starting point we used $x_i{}^0 = 4$ and $z_i{}^0 =$

$\|Mx - c\|_\infty$. Results for $n = 500$ using a number of other starting points are also provided in Table 6.

This experience suggests that degeneracy has some but not great influence on our method. Compared to the results reported in [5] on the nondegenerate problem, our algorithm took fewer iterations to converge.

TABLE 5
Problem VII: *Iterations of the algorithm.*

Dimension	nondegenerate	degenerate
100	10	17
200	11	18
500	11	18
1000	11	18

TABLE 6
Problem VII: *Iterations of the algorithm ($n = 500$).*

Starting point	nondegenerate	degenerate
${x_i}^0 = {z_i}^0 = 1$	6	16
${x_i}^0 = {z_i}^0 = 10$	14	18
${x_i}^0 = {z_i}^0 = 100$	21	24
${x_i}^0 = {z_i}^0 = 1000$	28	32

Problem VIII. This problem was also taken from [5] and is based on the same pentadiagonal matrix given by (34). Nondegenerate LCPs were generated as before with matrices of the form

$$M + \lambda_k I,$$

where $\lambda_k,\ k = 1, \ldots, 5$ is a set of positive real numbers such that $M + \lambda_1 I$ is strictly diagonally dominant and

$$\lambda_{k+1} = \frac{\lambda_k}{10^t},\ k = 1, \ldots, 4\ ,\ \lambda_1 = 10$$

for $t = 2$.

The performance of the algorithm with starting point ${x_i}^0 = 4$ and ${z_i}^0 = \|Mx - c\|_\infty$ is illustrated in Table 7.

TABLE 7
Problem VIII: *Iterations of the algorithm.*

λ_k	Iterations
10	11
$1.0E - 01$	9
$1.0E - 03$	10
$1.0E - 05$	11
$1.0E - 07$	11

Problem IX. Next, we tested our algorithm on large-scale LCPs with matrices that arise in the solution of the Laplace equation

$$\frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} = 0$$

on the rectangle $[0,b]\times[0,d]$, with b and d positive real numbers, by finite differences. This problem was taken from [5], and as it is described there, M is an $(m-1)(t-1)\times(m-1)(t-1)$ matrix of the form

$$M=\begin{pmatrix} B & -I_{m-1} & 0 & \cdots & 0 \\ -I_{m-1} & B & -I_{m-1} & \cdots & 0 \\ 0 & -I_{m-1} & B & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & B \end{pmatrix},$$

where B is an $(m-1)\times(m-1)$ tridiagonal matrix with elements

$$b_{i,i}=2\left[\left(\frac{bt}{dm}\right)^2+1\right], \qquad b_{i-1,i}=b_{i,i+1}=-\left(\frac{bt}{dm}\right)^2,$$

and I_{m-1} is the identity matrix of dimension $(m-1)$. The vector c was constructed so that the resulting LCPs have nondegenerate solutions with elements

$$x_i{}^*=1,\ z_i{}^*=0,\ i=1,\dots,\frac{n}{2},$$
$$x_i{}^*=0,\ z_i{}^*=1,\ i=\frac{n}{2}+1,\dots,n.$$

Table 8 displays the performance of the algorithm for four dimensions n, and four different starting points. These results confirm once more that increasing the dimension of the problem does not have a negative effect on the performance of the algorithm.

TABLE 8
Problem IX: *Iterations of the algorithm.*

b	d	m	t	Dimension	$x_i{}^0=1$ $z_i{}^0=1$	$x_i{}^0=10$ $z_i{}^0=10$	$x_i{}^0=100$ $z_i{}^0=100$	$x_i{}^0=4$ $z_i{}^0=\|Mx^0-c\|_\infty$
100	1	101	31	3000	5	10	15	13
1	100	101	31	3000	5	10	15	13
1	1	101	31	3000	5	10	15	13
1	1	401	16	6000	5	10	15	13
1	1	501	21	10000	5	10	15	13
1	1	501	31	15000	5	10	15	13

Problem X. This is a large-scale single commodity spatial equilibrium problem whose formulation as an LCP is fully described in [16] and [17]. The problem is to find the desired equilibrium prices p_i, the net import y_i, and the trade flow x_{ij} from region i to region j which satisfy for all regions $i,j=1,\dots,N$ the following system of equations:

$$\alpha_i-b_iy_i=p_i,$$
$$\sum_{j=1}^{n}x_{ji}-\sum_{j=1}^{n}x_{ij}=y_i,$$

TABLE 9
Problem X: *Iterations of the algorithm.*

		${x_i}^0 = 1$	${x_i}^0 = 10$	${x_i}^0 = 100$
N	Dimension $\|H\|$	${z_i}^0 = 1$	${z_i}^0 = 10$	${z_i}^0 = 100$
25	147	20	19	21
50	423	16	15	18

$$\begin{aligned} p_i + c_{ij} - p_j &\geq 0, \\ x_{ij} &\geq 0, \\ x_{ij}(p_i + c_{ij} - p_j) &= 0. \end{aligned}$$

As pointed out by Portugal and Júdice in [17], an important characteristic of this model is its network structure. The model can be represented by a directed graph where the node i represents the region i and the directed arc (i, j) represents the connection from region i to region j. The problem may be formulated as an LCP whose matrix M and vector c satisfy

$$M = G^T BG, \qquad c = -(G^T \alpha + c),$$

where G is the $N \times |H|$ node-arc incidence matrix related to the flow conservation equations in the graph $\Psi = (N, H)$ with $H = \{(i, j) : a_i - a_j + c_{ij} > 0\}$. $G^T BG$ is a singular symmetric positive semidefinite matrix of order $|H|$, where $|H|$ represents the number of edges in the set H. To generate the instances used to test our algorithm, we used the technique suggested in [17]. Namely, we generated the α_i's and b_i's to be random numbers in the interval $[0, 100]$ and the c_{ij}'s to be random numbers in the interval $[0, 80]$ that satisfy the triangular inequality $c_{ij} + c_{jk} \geq c_{ik}$ for all $i, j, k = 1, \ldots, N$.

Table 9 illustrates the performance of the algorithm for two values of n with three different starting points.

TABLE 10
Problem XI: *Iterations of the algorithm.*

Starting point	Iterations
${x_i}^0 = {z_i}^0 = 1$	–
${x_i}^0 = {z_i}^0 = .1$	17
${x_i}^0 = {z_i}^0 = 30$	31
${x_i}^0 = {z_i}^0 = 50$	29
${x_i}^0 = {z_i}^0 = 100$	34

Problem XI. Next we tested the algorithm on a 63-variable problem generated by linearizing a nonlinear complementarity problem. This is a problem of finding economic equilibria in a model of duopoly (see [12]) and is studied extensively in [23]. We generated our test problems by linearizing around different points, and then we attempted to solve the resulting LCPs as individual problems. These problems are difficult to solve because their matrices are nonsymmetric and indefinite. Theoretically, our algorithm is well defined as long as $Z + XM$ is nonsingular, and this experience shows that it works for problems of this sort.

Table 10 illustrates the performance of the algorithm on four instances of the problem. The values of x^0 given in the table represent the point around which the linearization was done. Moreover, the same x^0 was used as the starting point. In all cases, we took ${x_i}^0 = {z_i}^0 \ \forall \ i = 1, \ldots, n$. In one of these instances the algorithm did not converge. After 14 iterations, α_k was below 10^{-10}, implying that the LCP generated is probably infeasible.

Problem XII. Next we tested our algorithm on an infeasible 2×2 problem with matrix

$$M = \begin{pmatrix} -2 & -3 \\ -1 & 4 \end{pmatrix},$$

and

$$c = (2, 1)^T.$$

In this case, the sequence of α_k converged to zero quickly and the algorithm stopped. Specifically, after nine iterations the algorithm terminated with $\alpha_k = 2.411 \times 10^{-13}$. This happened because the matrix $X^k M + Z^k$ became more and more nearly singular. As a result, the vector of directions Δv^k started to diverge. Since the iterates (x^k, z^k) are bounded in $\Omega(\epsilon)$ the steplength α_k was driven to zero.

Problem XIII. Finally, we tested our algorithm on one more infeasible 2×2 problem taken from [10]. For this problem,

$$M = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix},$$

and

$$c = (1, 1)^T.$$

This time, α_k dropped to 2.6×10^{-10} in seven iterations and the algorithm was stuck.

This LCP has the characteristic that its matrix belongs in the P_0-class; thus, $X^k M + Z^k$ is nonsingular for every $(x^k, z^k) > 0$. Nevertheless, it is easy to check that the problem has no solution. The algorithm was stuck because z_2^k was being driven to zero which makes the matrix become singular in the limit.

5. Conclusions. In this paper, we presented an infeasible-interior-point algorithm with backtracking line search to solve the LCP. Convergence of this algorithm with an ℓ_2-norm was proven in section 3 under two assumptions, i.e., that the matrix $XM + Z$ used in the generation of the moving direction remains nonsingular at every step and that the matrix M satisfies that if $|x_i^k| \to +\infty$ for $i \in J \subset \{1, 2, \ldots, n\}$, then there exists a $j \in J$ such that also $|[Mx^k]_j| \to +\infty$.

Computational experience revealed that when the problem is infeasible, the matrix $XM + Z$ becomes singular and at least one element of the direction vector Δv becomes arbitrarily large. As a result, the steps α_k tend rapidly to zero (see also [24]). Interestingly, the condition that drives them to zero is the centering condition rather than the feasibility condition. Actually, both the condition that ensures that feasibility is given a higher priority than complementarity and the backtracking line search are rarely invoked by the algorithm in practice.

Our method is not restricted to P_*-matrices, unlike most algorithms for nonmonotone LCPs documented in the literature. No particularly strong requirements were imposed on matrix M. This is important since such requirements are often not satisfied in applications of LCPs.

Even though our convergence result does not exclude the possibility that the algorithm will terminate without a solution when a solution exists, the method was remarkably successful on all the problems tested. The numerical results presented in section 4 prove that the dimension of the problem has no effect on the performance of the algorithm. Moreover, degeneracy did not appear to have any significant effect even though it is known to cause difficulties in general. What appears to influence performance is the starting point. Trying different starting points can result, in some cases, in significant changes in the performance of the algorithm. Unfortunately, none of the strategies we tried proved to be successful independent of the problem. Actually, simply setting ${x_i}^0 = {z_i}^0 = \beta$, $\beta > 0$, worked better in certain cases than the more sophisticated strategy that involves the calculation of the max-norm. We believe that further research should be directed to finding “good” starting points for the algorithm.

REFERENCES

[1] R. Cottle, J. S. Pang, and R. E. Stone, *The Linear Complementarity Problem*, Academic Press, New York, San Diego, 1992.

[2] T. A. Davis, *Users' Guide for the Unsymmetric-Pattern Multifrontal Package (UMFPACK)*, Technical report, Computer and Information Sciences Department, University of Florida, Gainesville, FL, 1993.

[3] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1983.

[4] A. S. El-Bakry, R. A. Tapia, T. Tsuchiya, and Y. Zhang, *On the Formulation and Theory of the Primal-Dual Newton Interior-Point Method for Nonlinear Programming*, Technical report TR92-40, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1992.

[5] L. Fernandes, J. Júdice, and J. Patrício, *An Investigation of Interior-Point and Block Pivoting Algorithms for Large-Scale Symmetric Monotone Linear Complementarity Problems*, Research report, Department of Mathematics, University of Coimbra, Portugal, 1994.

[6] P. Harker and J. Pang, *A damped-Newton method for the linear complementarity problem,* in Simulation and Optimization of Large Systems, G. Allgower and K. Georg, eds., Lectures in Applied Mathematics 26, American Mathematical Society, Providence, RI, 1990, pp. 265–284.

[7] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economics and Mathematical Systems 187, Springer-Verlag, Berlin, Germany, 1981.

[8] C. Kanzow, *A New Approach to Continuation Methods for Complementarity Problems with P–Functions*, Research report, Institute of Applied Mathematics, University of Hamburg, Germany, 1994.

[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, N. Megiddo, T. Noma, and A. Yoshise, *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems*, Lecture Notes in Computer Science 538, Springer-Verlag, Berlin, Germany, 1991.

[11] M. Kojima, T. Noma, and A. Yoshise, *Global convergence in infeasible-interior-point algorithms*, Math. Programming, 65 (1994), pp. 43–72.

[12] E. Maskin and J. Tirole, *A theory of dynamic oligopoly,* II*: Price competition, kinked demand curves, and edgeworth cycles*, Econometrica, 56 (1988), pp. 571–579.

[13] J. Miao, *A quadratically convergent $O((1+k)\sqrt{n}l)$–iteration algorithm for the $P_*(k)$–matrix linear complementarity problem*, Math. Programming, 69 (1995), pp. 355–368.

[14] S. Mizuno, *Polynomiality of infeasible–interior–point algorithms for linear programming*, Math. Programming, 67 (1994), pp. 109–119.

[15] K. G. Murty, *Linear Complementarity, Linear and Nonlinear Programming*, Sigma Series in Applied Mathematics 3, Heldermann Verlag, Berlin, Germany, 1988.

[16] J.-S. Pang and P. S. C. Lee, *A parametric linear complementarity technique for the computation of equilibrium prices in a single commodity model*, Math. Programming, 20 (1981), pp. 81–102.

[17] L. Portugal and J. Júdice, *A Hybrid Algorithm for the Solution of a Single Commodity Spatial Equilibrium Model*, Research report, Department of Mathematics, University of Coimbra, Portugal, 1994.

[18] F. Potra and R. Sheng, *A Large-Step Infeasible–Interior–Point Method for the $P_*(k)$–Matrix LCP*, Reports on Computational Mathematics 64, Department of Mathematics, The University of Iowa, Iowa City, IA, 1994.

[19] F. A. Potra, *A quadratically convergent predictor–corrector method for solving linear programs from infeasible starting points*, Math. Programming, 67 (1994), pp. 383–406.

[20] F. A. Potra and R. Sheng, *Predictor–Corrector Algorithms for Solving $P_*(k)$–Matrix LCP from Arbitrary Positive Starting Points*, Reports on Computational Mathematics 58, Department of Mathematics, The University of Iowa, Iowa City, IA, 1994.

[21] F. A. Potra and R. Sheng, *A Superlinearly Convergent Infeasible–Interior–Point Algorithm for Degenerate LCP*, Reports on Computational Mathematics 66, Department of Mathematics, The University of Iowa, Iowa City, IA, 1995.

[22] D. F. Shanno, *Computational Experience with Logarithmic Barrier Methods for Linear and Nonlinear Complementarity Problems*, Rutcor Research report 18-93, RUTCOR, New Brunswick, NJ, 1993.

[23] E. Simantiraki and D. Shanno, *Computing Equilibria of Oligopolistic Pricing Models*, Rutcor Research report 41-95, RUTCOR and Graduate School of Management, Rutgers University, New Brunswick, NJ, 1995.

[24] E. Simantiraki and D. Shanno, *An Infeasible-Interior-Point Algorithm for Solving Mixed Complementarity Problems*, Rutcor Research report 37-95, RUTCOR and Graduate School of Management, Rutgers University, New Brunswick, NJ, 1995.

[25] S. J. Wright, *An infeasible-interior-point algorithm for linear complementarity problems*, Math. Programming, 67 (1994), pp. 29–51.

[26] Y. Ye, *A further result on the potential reduction algorithm for the P–matrix linear complementarity problem*, in Advances in Optimization and Parallel Computing, P.M. Pardalos, ed., Elsevier Science Publishers B.V., New York, 1992, pp. 311–316.

[27] 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.

[28] Y. Zhang and R. A. Tapia, *A superlinearly convergent polynomial primal-dual interior-point algorithm for linear programming*, SIAM J. Optim., 3 (1993), pp. 118–133.

SIAM J. OPTIM.
Vol. 7, No. 3, pp. 641–662, August 1997

004

STRONG DUALITY FOR SEMIDEFINITE PROGRAMMING[*]

MOTAKURI V. RAMANA[†], LEVENT TUNÇEL[‡], AND HENRY WOLKOWICZ[‡]

Abstract. It is well known that the duality theory for linear programming (LP) is powerful and elegant and lies behind algorithms such as simplex and interior-point methods. However, the standard Lagrangian for nonlinear programs requires constraint qualifications to avoid duality gaps.

Semidefinite linear programming (SDP) is a generalization of LP where the nonnegativity constraints are replaced by a semidefiniteness constraint on the matrix variables. There are many applications, e.g., in systems and control theory and combinatorial optimization. However, the Lagrangian dual for SDP can have a duality gap.

We discuss the relationships among various duals and give a unified treatment for strong duality in semidefinite programming. These duals guarantee strong duality, i.e., a zero duality gap and dual attainment. This paper is motivated by the recent paper by Ramana where one of these duals is introduced.

Key words. semidefinite linear programming, strong duality, Löwner partial order, symmetric positive semidefinite matrices

AMS subject classifications. Primary, 65K10; Secondary, 90C25, 90M45, 15A45, 47D20

PII. S1052623495288350

1. Introduction.

1.1. Semidefinite programming (SDP).

We study strong duality theorems for the semidefinite linear programming problem

$$\text{(P)} \qquad \begin{array}{rcl} p^* = & \sup & c^t x \\ & \text{subject to} & Ax \preceq b \\ & & x \in \Re^m, \end{array}$$

where $c, x \in \Re^m$; $b = Q_0 \in \mathcal{S}_n$, the *space of symmetric $n \times n$ matrices*; the linear operator $Ax = \sum_{i=1}^m x_i Q_i$ for $Q_i \in \mathcal{S}_n$, $i = 1, \ldots, m$; and $\preceq$ denotes the Löwner partial order, i.e., $X \preceq (\prec)\ Y$ means $Y - X$ is positive semidefinite (positive definite). We let $\mathcal{P}$ denote the cone of semidefinite matrices. By a *cone* we mean a convex cone, i.e., a set K satisfying $K + K \subset K$ and $\lambda K \subset K$ for all $\lambda \geq 0$. We consider the space of symmetric matrices, $\mathcal{S}_n$, as a vector space with the *trace inner product* $\langle U, X \rangle := \operatorname{trace} UX$. (Over the space of $n \times n$ matrices, $\langle U, X \rangle := \operatorname{trace}(U^t X)$.) The corresponding norm is the *Frobenius matrix norm* $||X|| = \sqrt{\operatorname{trace} X^2}$.

We let F denote the feasible set of (P), and we assume that the optimal value p^* is finite. (This implies that the feasible set $F \neq \emptyset$.)

1.2. Background.

1.2.1. Cone of semidefinite matrices.

The cone of positive semidefinite matrices has been studied extensively for both its importance and geometric elegance. Positive definite matrices arise naturally in many areas, including differential equations, statistics, and systems and control theory. The cone $\mathcal{P}$ induces a partial order on $\mathcal{S}_n$ called the Löwner partial order. Various monotonicity results were studied

[*] Received by the editors June 22, 1995; accepted for publication (in revised form) April 23, 1996. http://www.siam.org/journals/siopt/7-3/28835.html

[†] Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611 (ramana@iseipc.ise.ufl.edu).

[‡] Department of Combinatorics and Optimization, Faculty of Mathematics, Waterloo, Ontario N2L 3G1, Canada (ltuncel@math.uwaterloo.ca, henry@orion.uwaterloo.ca).

with respect to this partial order [28, 29]. An early paper in this area is the one by Bohnenblust [9]. Optimization problems over cones of matrices are also discussed in the monograph by Berman [8].

More recently, we have seen a strong renewed interest in semidefinite programming. This is due to new applications in engineering (e.g., Ben-Tal and Nemirovskii [7], Boyd et al. [14], and Vandenberghe and Boyd [36]) and combinatorial optimization (e.g., Alizadeh [1], Goemans and Williamson [19], Lovász and Schrijver [27], Nesterov and Nemirovskii [30], Delorme and Poljak [15], and Helmberg et al. [22]). Other applications of SDP arise from the study of correlation matrices in statistics, e.g., Pukelsheim [31]; matrix completion problems, see [20, 5, 24]; and multiquadratic programs, e.g., [32].

Nesterov and Nemirovskii's book provides a unifying framework for polynomial-time interior-point algorithms in convex programming (which includes SDP). Currently, interior-point algorithms seem to be the best algorithms (from both theoretical and practical viewpoints) for solving SDP problems, e.g., [36]. An infeasible-start interior-point algorithm was presented in Freund [17]. Complexity of the algorithm depends on the distances (in a norm induced by the initial solution) of the initial solution to the sets of approximately feasible and approximately optimal solutions, where approximate feasibility and optimality are defined in terms of given tolerances. The algorithm does not assume that the zero duality gap (or even feasibility) is attainable. Indeed, for the case when the given problem exhibits a finite nonzero duality gap, we can ask for a tolerance in the duality gap that is not attainable (for such a tolerance, the distance from the set of approximately optimal solutions would be infinite for any starting point). This illustrates some of the difficulties encountered with nonzero duality gaps. Our goal here is to study and unify the ways in which a dual problem can be modified to ensure a zero duality gap at optimality.

1.2.2. Early duality results. Extensions of finite linear programming duality to infinite dimensions and/or to optimization problems over cones have been studied in the literature. We do not give a comprehensive survey, but we mention several early results.

In [16], Duffin studies infinite linear programs, i.e., programs for which there are an infinite number of constraints and/or an infinite number of variables. Also studied in [16] is the notion of optimization with respect to a partial order induced by a cone. Duality theory is also central in the related notion of continuous programming, e.g., [25, 26, 34], which is closely tied in with infinite programming. A major question is the formulation of duals that close the duality gap. Infinite dimensional linear programming is also studied in the books by Glashoff and Gustafson [18] and Anderson and Nash [2].

More recently, duals that guarantee strong duality for general abstract convex programs have been given in [13, 12, 11, 10]. The special case of a linear program with cone constraints is treated in [38].

1.3. Outline. This paper is motivated by the recent paper of Ramana [33]. A dual program, called an *extended Lagrange–Slater dual program* and denoted (ELSD), is presented therein. Strong duality holds for this dual and, in addition, it can be written down in polynomial time. Previous work on general (convex) cone constrained programs [13, 38, 11, 10] also presented dual programs for which strong duality holds. The results were based on regularization and on finding the so-called minimal cone of the program (P). We denote these duals by (DRP). A procedure for defining the minimal cone was presented in [11]. This procedure started with an initial feasible point and reduced the program, in a finite number of steps, to a regularized program.

The main result in this paper is to show that the extended Lagrange dual program (ELSD) is equivalent to the regularized dual (DRP). This equivalence is in the sense that the constraints and the set of Lagrange multipliers are the same. The difference in the duals is the fact that the feasible set of Lagrange multipliers, denoted $(\mathcal{P}^f)^+$, is expressed implicitly in (ELSD) as the solution of m systems of constraints included in the dual, whereas it is defined explicitly in (DRP) as the output of the separate procedure mentioned above. This separate procedure finds the minimal cone by solving a system of constraints equivalent to that in (ELSD). Also presented is an extended dual of the dual; i.e., this closes the duality gap from the dual side.

The fact that the two duals (ELSD) and (DRP) are found using different techniques and then result in being equivalent is more than a coincidence. In fact, we show that such duals are uniquely identified in a certain sense.

In section 2 we discuss the geometry of the cone of semidefinite matrices. In particular, we present old and new results on the faces of this cone. Lemmas 2.1 and 2.2 provide a description of the faces and characterization of the cases in which the sum of the positive semidefinite cone and a subspace is closed. The two strong duality schemes are outlined in section 3. The relationships between the duals is presented in section 4. We include the results on the extended Lagrange–Slater dual of the Lagrangian dual of (P). In section 5, we present a homogenized program which is equivalent to SDP and provides a different view of optimality conditions. We conclude with some remarks on perturbations of SDP and computational complexity issues.

2. Geometry of the SDP cone.

2. Geometry of the SDP cone. We now outline several known and some new results on the geometry of the cone $\mathcal{P}$. More details can be found in [3, 4]. For an introduction to the geometry of convex sets, see Rockafellar [35].

The cone $K \subset T$ is a *face* of the cone T, denoted $K \lhd T$, if

$$x, y \in T,\ x + y \in K \Rightarrow x, y \in K. \tag{2.1}$$

The faces of $\mathcal{P}$ have a very special structure. Each face, $K \lhd \mathcal{P}$, is characterized by a unique subspace, $S \subset \Re^n$:

$$K = \{X \in \mathcal{P} : \mathcal{N}(X) \supset S\}.$$

Moreover,

$$\operatorname{relint}(K) = \{X \in \mathcal{P} : \mathcal{N}(X) = S\}.$$

The *complementary* (or conjugate) face of K is $K^c = K^\perp \cap \mathcal{P}$ and

$$K^c = \{X \in \mathcal{P} : \mathcal{N}(X) \supset S^\perp\}. \tag{2.2}$$

Moreover,

$$\operatorname{relint}(K^c) = \{X \in \mathcal{P} : \mathcal{N}(X) = S^\perp\}.$$

Equivalent characterizations for K and K^c are given in (2.6) and (2.7).

Two additional facts about the faces of the cone $\mathcal{P}$ are as follows:

(i) Each face K (respectively, K^c) is *exposed*; i.e., it is equal to the intersection of $\mathcal{P}$ with a supporting hyperplane; the supporting hyperplane corresponds to any

$X \in \operatorname{relint}(K^c)$ (respectively, $\operatorname{relint}(K)$). Also, complementary faces are orthogonal and satisfy $XY = 0$ for all $X \in K, Y \in K^c$.

(ii) The cone $\mathcal{P}$ is *projectionally exposed* (see [11]); i.e., every face of $\mathcal{P}$ is the image of $\mathcal{P}$ under some projection. In fact, if $Q \in \mathcal{S}_n$ is the projection onto the subspace S, the null space of matrices in $\operatorname{relint}(K)$, then the face K satisfies

$$K = (I - Q)\mathcal{P}(I - Q).$$

The *minimal cone* of (P) is defined as

$$\mathcal{P}^f = \cap\{K \lhd \mathcal{P} : K \supset (b - A(F))\}, \tag{2.3}$$

i.e., the minimal cone is the intersection of all faces of $\mathcal{P}$ containing the feasible slacks.

The following lemma shows that we can express the orthogonal complement of a face completely in terms of a system of semidefinite inequalities. The semidefinite inequalities are based on the data of the original problem. The description is made possible by using a semidefinite completion problem.

LEMMA 2.1. *Suppose that C is a convex cone and $C \subset \mathcal{P}$. Let*

$$K := \{W + W^t : U \succeq WW^t \textit{ for some } U \in C\}.$$

Then

$$\begin{aligned} ((\mathcal{F}(C))^c)^\perp &= K \\ &= \left\{ W + W^t : \begin{bmatrix} I & W^t \\ W & U \end{bmatrix} \succeq 0 \textit{ for some } U \in C \right\}. \end{aligned} \tag{2.4}$$

Proof. Suppose that $W + W^t \in K$, i.e., $U \succeq WW^t$ for some $U \in C$. Since $x^t(U - WW^t)x \geq 0$ for all x, we get $\mathcal{N}(U) \subset \mathcal{N}(W^t)$. Equivalently, $\mathcal{R}(U) \supset \mathcal{R}(W)$. Since $UU^\dagger$ is the orthogonal projection onto the range of U, where $U^\dagger$ denotes the Moore–Penrose generalized inverse of U, we conclude that $W = UU^\dagger W$. We have shown that

$$U \succeq WW^t \Rightarrow W = UH \text{ for some H}. \tag{2.5}$$

(See, e.g., [33].) Therefore, $\operatorname{trace} WV = 0$ for all $V \in (\mathcal{F}(C))^c$, i.e., $W + W^t \in ((\mathcal{F}(C))^c)^\perp$. To prove the converse inclusion, suppose that $V \in ((\mathcal{F}(C))^c)^\perp$ and $U \in C \cap \operatorname{relint}(\mathcal{F}(C))$. Let U be orthogonally diagonalized by $Q = [Q_1, Q_2]$:

$$U = Q\operatorname{Diag}(d_1\ 0)Q^t,\ Q^tQ = I,$$

with $Q_1, n\times r,\ d_1 > 0$. Therefore, the minimal face can be written using block matrices as follows:

$$\begin{aligned} \mathcal{F}(C) &= \{Q_1BQ_1^t : B \succeq 0,\ B \in \mathcal{S}_r\} \\ &= \left\{ Q \begin{bmatrix} B & 0 \\ 0 & 0 \end{bmatrix} Q^t : B \succeq 0,\ B \in \mathcal{S}_r \right\} \end{aligned} \tag{2.6}$$

and

$$\begin{aligned} (\mathcal{F}(C))^c &= \{Q_2BQ_2^t : B \succeq 0,\ B \in \mathcal{S}_{n-r}\} \\ &= \left\{ Q \begin{bmatrix} 0 & 0 \\ 0 & B \end{bmatrix} Q^t : B \succeq 0,\ B \in \mathcal{S}_{n-r} \right\}. \end{aligned} \tag{2.7}$$

This implies that V in $((\mathcal{F}(C))^c)^\perp$ can be written in terms of blocks as

$$V = Q\left(\begin{bmatrix} .5T & C \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} .5T & 0 \\ C^t & 0 \end{bmatrix}\right)Q^t.$$

We then have

$$X = Q\left(\begin{bmatrix} T^2 + CC^t & 0 \\ 0 & 0 \end{bmatrix}\right)Q^t \preceq \alpha U$$

for sufficiently large α, i.e., $V + V^t \in K$.

The alternate expression for K in (2.4) follows from the Schur complement. □

Now, we note the following interesting and surprising closure property of the faces of $\mathcal{P}$. This is surprising because it is not true in general that the sum of a cone and a subspace is closed.

LEMMA 2.2. *Suppose that the face K satisfies*

$$\{0\} \neq K \lhd \mathcal{P}, \;\; K \neq \mathcal{P}.$$

Then

$$\mathcal{P} + K^\perp = \overline{\mathcal{P} + \operatorname{span} K^c}; \tag{2.8}$$

$$\mathcal{P} + \operatorname{span} K \quad \textit{is not closed.} \tag{2.9}$$

Proof. Since $\operatorname{span} K^c \subset K^\perp$, we get

$$\mathcal{P} + K^\perp \supset \mathcal{P} + \operatorname{span} K^c.$$

From the characterization of faces in [3, 4], there exists a subspace $S \subset \Re^n$, with dimension k, such that

$$K = \{X \succeq 0 : \mathcal{N}(X) \supset S\}.$$

After applying an orthogonal transformation to $\Re^n$, we can assume that S is the span of the first k unit vectors. Therefore, $X \in K$ has a $k \times k$ zero block, i.e.,

$$X = \begin{bmatrix} 0_k & 0 \\ 0 & \bar{X} \end{bmatrix}.$$

Moreover, for X in the relative interior of K, we have $\bar{X} \succ 0$. This implies that

$$K^\perp = \left\{Y : Y = \begin{bmatrix} C & D \\ D^t & 0 \end{bmatrix}, \; C \in \mathcal{S}_k, \; D \in \mathcal{M}_{k,n-k}\right\}.$$

Now suppose that we are given $T^n \in K^\perp, \; P^n \in \mathcal{P}, \; n = 1, 2, \ldots$ and the sequence

$$T^n + P^n \to L = \begin{bmatrix} L_1 & L_2 \\ L_2^t & L_3 \end{bmatrix}.$$

Comparing the corresponding bottom right blocks, we see that necessarily $L_3 \succeq 0$. Therefore,

$$L = \begin{bmatrix} L_1 & L_2 \\ L_2^t & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & L_3 \end{bmatrix},$$

i.e., $L \in K^\perp + \mathcal{P}$. This proves that $\mathcal{P} + K^\perp$ is closed, i.e.,

$$\mathcal{P} + K^\perp \supset \overline{\mathcal{P} + \operatorname{span} K^c}.$$

To prove the converse inclusion, suppose that

$$W \in (\mathcal{P} + K^\perp) \setminus (\overline{\mathcal{P} + \operatorname{span} K^c}).$$

Then there exists a separating hyperplane, i.e., there exists Φ such that

$$\text{trace}\,\Phi W < 0 \leq \text{trace}\,\Phi(P + w) \;\forall P \in \mathcal{P},\; w \in \operatorname{span} K^c. \tag{2.10}$$

This implies that $\Phi \succeq 0$ and $\Phi \in (K^c)^\perp$. But then $\text{trace}\,\Phi W = \text{trace}\,\Phi P + \text{trace}\,\Phi w$, with $P \in \mathcal{P}, w \in K^\perp$. From Lemma 2.1 and (2.5) we get that $w = UH + H^t U$ for some $U \in K^c$, so $\text{trace}\,\Phi w = 0$. This implies that $\text{trace}\,\Phi W = \text{trace}\,\Phi P \geq 0$, which contradicts (2.10). This completes the proof of (2.8).

Now suppose that $X \in \operatorname{relint}(K)$ and $X = QDQ^t,\ Q = [Q_1,\ Q_2],\ QQ^t = I$, is an orthogonal diagonalization of X with the columns of Q_1 spanning $\mathcal{N}(X)$ and the columns of Q_2 spanning $\mathcal{R}(X)$. Then $K = \{Q_2 B Q_2^t : B \succeq 0\}$, and $\operatorname{span} K = \{Q_2 B Q_2^t : B \in \mathcal{S}\}$. Now let $B \succ 0, T \succ 0$, and $n = 1, 2, \ldots$. Choose T, L so that

$$[Q_1\ Q_2] \begin{bmatrix} \frac{1}{n}T & L \\ L^t & nB \end{bmatrix} \begin{bmatrix} Q_1^t \\ Q_2^t \end{bmatrix} \in \mathcal{P}.$$

But

$$[Q_1\ Q_2] \begin{bmatrix} 0 & 0 \\ 0 & -nB \end{bmatrix} \begin{bmatrix} Q_1^t \\ Q_2^t \end{bmatrix} \in \operatorname{span} K.$$

However, the limit of the sum of the two sequences is

$$[Q_1\ Q_2] \begin{bmatrix} 0 & L \\ L^t & 0 \end{bmatrix} \begin{bmatrix} Q_1^t \\ Q_2^t \end{bmatrix},$$

which is not in the sum $(\mathcal{P} + \operatorname{span} K)$. □

COROLLARY 2.1.

$$(\mathcal{P}^f)^+ = \mathcal{P}^+ + (\mathcal{P}^f)^\perp = \mathcal{P} + (\mathcal{P}^f)^\perp = \overline{\mathcal{P} + \operatorname{span}(\mathcal{P}^f)^c}.$$

Proof. From the definition of a face and the closure condition above, we get

$$\begin{aligned}(\mathcal{P}^f)^+ &= (\mathcal{P} \cap \mathcal{P}^f)^+ \\ &= (\mathcal{P} \cap \operatorname{span}(\mathcal{P}^f))^+ \\ &= \mathcal{P}^+ + (\mathcal{P}^f)^\perp. \qquad □\end{aligned}$$

3. Duality schemes.

3.1. Lagrangian duality.

The Lagrangian for (P) is

$$L(x, U) = c^t x + \text{trace}\, U(b - Ax).$$

Consider the max-min problem

$$p^* = \max_x \min_{U \succeq 0} L(x, U).$$

The inner minimization problem has the hidden constraint $Ax \preceq b$; i.e., the minimization problem is unbounded otherwise. Once this hidden constraint is added to the outer maximization problem, the minimization problem has optimum $U = 0$. Therefore we see that this max-min problem is equivalent to the primal (P). This illustrates that we have the correct constraint on the dual variable U. (See, for instance, the arguments in Duffin [16] and Alizadeh [1].)

The Lagrangian dual to (P) is obtained by reversing the max-min to a min-max and rewriting the Lagrangian, i.e.,

$$p^* \le d^* = \min_{U \succeq 0} \max_{x} \left\{ L(x,U) = \operatorname{trace} bU + x^t(c - A^*U) \right\}.$$

Here A^* denotes the *adjoint* of the linear operator A, i.e.,

$$(A^*U)_i = \operatorname{trace} Q_i U. \tag{3.1}$$

The inner maximization now has the hidden constraint $c - A^*U = 0$. Once this hidden constraint is added to the outer minimization problem, the inner maximization has optimum $x = 0$. Therefore, we see that this min-max problem is equivalent to the following dual program:

$$\text{(D)} \qquad \begin{array}{rcc} d^* = & \min & \operatorname{trace} bU \\ & \text{subject to} & A^*U = c \\ & & U \succeq 0. \end{array}$$

3.1.1. Linear programming special case. We note that the SDP pair (P) and (D) look exactly like LP duals but with $\ge$ replaced by $\succeq$. In fact, if the adjoint operator A^* includes constraints that force U to be diagonal, then we see that LP is a special case of SDP.

Now suppose that we consider (P) and (D) as LPs, i.e., suppose that we replace $\succeq$ with $\ge$. Then the operator A is an $n \times m$ matrix, and $U \in \Re^n$. In this special case (since we assumed that the primal feasible set is nonempty), we always have strong duality, i.e., $p^* = d^*$ and d^* is attained. Moreover, we can have more than one dual of (P). Let $P^=$ denote the set of indices of the rows of A corresponding to the implicit equality constraints, i.e.,

$$P^= := \{ i : x \in F \text{ implies } A_{i:}x = b_i \},$$

where $A_{i:}$ denotes the ith row of A. Then we can consider the equality constraints $A_{i:}x = b_i$ for any subset of $P^=$, without changing (P). This is equivalent to allowing the dual variables $U_i,\ i \in P^=$, to be free rather than nonnegative. Thus we see that we can have different duals for (P) while maintaining strong duality. In fact, there are an infinite number of duals, since the space of dual variables can be any set which includes the nonnegative orthant and restricts $U_i \ge 0,\ i \notin P^=$.

It is clearly better to have a smaller set of dual variables. In fact, in the case of LP discussed above, if some of the inactive constraints at the optimum can be identified, then we can restrict the corresponding dual variables to be 0. This is equivalent to ignoring the inactive constraints. Of course, we do not in general know which constraints will be active at the optimum.

Having more than one dual program occurs because there is no strictly feasible solution for (P). We see below that a similar phenomenon occurs for (P) in the SDP case but with the additional complication of possible loss of strong duality. In addition, the semidefinite constraint is not as simple as the nonnegativity constraint in LP. The question arises whether or not we get the same dual if we treat the semidefinite constraint $U \succeq 0$ as a functional constraint using the smallest eigenvalue of U.

3.2. Strong duality and regularization. If a *constraint qualification*, denoted CQ (see section 5), holds for P, then we have strong duality for the Lagrange dual program; i.e., $p^* = d^*$ and d^* is attained. The usual CQ is *Slater's condition:* there exists $\hat{x}$ such that $(b - A\hat{x}) \in \operatorname{int} \mathcal{P}$. Examples where $p^* < d^*$ and/or one of d^*, p^* is not attained have appeared in the literature; see, e.g., [17]. One can close the duality gap by using the minimal cone of $\mathcal{P}$. Therefore, an equivalent program is the *regularized primal program*; see [11, 38]:

$$\text{(RP)} \qquad \begin{array}{rcc} p^* = & \max & c^t x \\ & \text{subject to} & Ax \preceq_{\mathcal{P}^f} b \\ & & x \in \Re^m. \end{array}$$

Moreover, by the definition of faces, there exists $\hat{x}$ such that $(b - A\hat{x}) \in \operatorname{relint}(\mathcal{P}^f)$. Therefore, the generalized Slater's constraint qualification holds; i.e., strong duality holds for this program. (This is proved in detail in [11, 38].) Thus, the following is a dual program for (P) for which strong duality holds:

$$\text{(DRP)} \qquad \begin{array}{rcc} p^* = & \min & \operatorname{trace} bU \\ & \text{subject to} & A^*U = c \\ & & U \succeq_{(\mathcal{P}^f)^+} 0, \end{array}$$

where the polar cone

$$(\mathcal{P}^f)^+ := \{U : \operatorname{trace} UP \geq 0 \ \forall P \in \mathcal{P}^f\}.$$

One can also close the duality gap from the dual side. Let F_D denote the feasible set of (D). The *minimal cone* of (D) is defined as

$$\mathcal{P}^f_D = \cap \{K : K \lhd \mathcal{P}, K \supset F_D\}. \tag{3.2}$$

Therefore, an equivalent program is the *regularized dual program*

$$\text{(RD)} \qquad \begin{array}{rcc} d^* = & \min & \operatorname{trace} bU \\ & \text{subject to} & A^*U = c \\ & & U \succeq_{\mathcal{P}^f_D} 0. \end{array}$$

Strong duality holds for this program. We therefore get the following strong dual of (D).

$$\text{(DRD)} \qquad \begin{array}{rcc} d^* = & \max & c^t x \\ & \text{subject to} & Ax \preceq_{(\mathcal{P}^f_D)^+} b \\ & & x \in \Re^m. \end{array}$$

The above presents two pairs of symmetric dual programs: (RP) and (DRP); (RD) and (DRD). The following theorem states that these dual pairs have all the nice properties of dual pairs in ordinary linear programming, i.e., [38, Theorem 4.1]. (Part 3 of Theorem 3.1 modifies and corrects the statement in [3.8].) This extends the duality results over polyhedral cones presented in [6].

THEOREM 3.1. *Consider the paired regularized programs (RP) and (DRP).*

1. *If one of the problems is inconsistent, then the other is inconsistent or unbounded.*

2. *Let the two problems be consistent, and let x^0 be a feasible solution for (P) and U^0 be a feasible solution for (DRP). Then*

$$c^t x^0 \leq \operatorname{trace} bU^0.$$

3. *If both (RP) and (DRP) are consistent, then their optimal values are equal and (DRP) has an optimal solution.*

4. *Let x^0 and U^0 be feasible solutions of (RP) and (DRP), respectively. Then x^0 and U^0 are optimal if and only if*

$$\operatorname{trace} U^0(b - Ax^0) = 0$$

and if and only if

$$U^0(b - Ax^0) = 0.$$

5. *The vector $x^0 \in \Re^m$ and matrix $U \in \mathcal{S}_n$ are optimal solutions of (RP) and (DRP), respectively, if and only if (x^0, U^0) is a saddle point of the Lagrangian $L(x, U)$ for all (x, U) in $\Re^m \times (\mathcal{P}^f)^+$. Then,*

$$L(x^0, U^0) = c^t x^0 = \operatorname{trace} bU^0.$$

3.3. Extended duals. The above dual program (DRP) uses the minimal cone explicitly. In [33], the *extended Lagrange–Slater dual* program, $(ELSD)$, is proposed. First define the following sets:

$$\begin{aligned} \mathcal{C}_k &= \{(U_i, W_i)_{i=1}^k : A^*(U_i + W_{i-1}) = 0,\ \operatorname{trace} b(U_i + W_{i-1}) = 0, \\ &\qquad U_i \succeq W_i W_i^t\ \forall i = 1, \ldots, k, W_0 = 0\}, \\ \mathcal{U}_k &= \{U_k : (U_i, W_i)_{i=1}^k \in \mathcal{C}_k\}, \\ \mathcal{W}_k &= \{W_k : (U_i, W_i)_{i=1}^k \in \mathcal{C}_k\}. \end{aligned} \tag{3.3}$$

Note that Schur complements imply that

$$U_i \succeq W_i W_i^t \iff \begin{bmatrix} I & W_i^t \\ W_i & U_i \end{bmatrix} \succeq 0.$$

In [33] it is shown that strong duality holds for the following (ELSD) dual of (P):

$$\text{(ELSD)} \qquad \begin{array}{ccc} p^* = & \min & \operatorname{trace} b(U + W) \\ & \text{subject to} & A^*(U + W) = c \\ & & W \in \mathcal{W}_m \\ & & U \succeq 0. \end{array}$$

The advantage for this dual is that it is stated completely in terms of the data of the original program, whereas (DRP) uses the minimal cone explicitly. Moreover, the size of (ELSD) is bounded by a polynomial function of the size of the input problem (P).

At a first glance, the duals (DRP) and (ELSD) appear very different. This is especially true in light of the fact that the matrices W do not have to be symmetric. However, the adjoint operator A^* involves traces which are unchanged by taking the symmetric part of the matrices. Therefore, we can replace W by $W + W^t$ or, equivalently, replace $\mathcal{W}_m$ by $\mathcal{W}_m^s$. We show below that after this change, the two duals are actually the same, i.e., $\mathcal{P} + \mathcal{W} = (\mathcal{P}^f)^+$, where

$$\mathcal{W} = \mathcal{W}_m^S = \{W + W^t : W \in \mathcal{W}_m\}.$$

4. Relationship between duals.

4.1. Duals of (P). We now show the relationships between the above two strong dual programs.

The algorithm to find the minimal cone is based on [11, Lemma 7.1], which we now phrase for our specific problem (P). We include a proof for completeness.

LEMMA 4.1. *Suppose $\mathcal{P}^f \lhd K \lhd \mathcal{P}$. For every solution U of the system*

$$A^*U = 0, U \succeq_{K^+} 0, \operatorname{trace} Ub = 0, \tag{4.1}$$

we have

$$\text{the minimal cone } \mathcal{P}^f \subset \{U\}^\perp \cap K \lhd K. \tag{4.2}$$

Proof. Since $\operatorname{trace} U(Ax - b) = 0$ for all x, we get $(A(F) - b) \subset \{U\}^\perp$, i.e., $\mathcal{P}^f \subset \{U\}^\perp$. Also, the fact that $\{U\}^\perp \cap K$ is a face of K follows from $U \succeq_{K^+} 0$. ☐

The result in [11, Lemma 7.1] is for more general convex, vector valued functions. However, the linearity of (P) means that it is equivalent to our statement above.

We now use the algorithm for finding $\mathcal{P}^f$ (presented in [11]) to show the relation between the two duals of (P). We see that each step of the algorithm finds a smaller dimensional face $\mathcal{P}_k$ which contains the minimal cone $\mathcal{P}^f$. We show that

$$\mathcal{P}_k^+ = \mathcal{P} + \mathcal{W}_k^s,\ \mathcal{W}_k^s = (\mathcal{P}_k)^\perp.$$

There is one difference with the algorithm discussed here and the one from [11]; here we find the points in the relative interior of the complementary faces, rather than an arbitrary point (which may be on the boundary). This guarantees the immediate correspondence with the dual (ELSD).

Step 1

Define $\mathcal{P}_0 := \mathcal{P}$ and note that, since $W_0 = 0$ in (3.3),

$$\mathcal{U}_1 := \{U \succeq 0 : A^*U = 0, \operatorname{trace} Ub = 0\}.$$

Choose $\hat{U}_1 \in \operatorname{relint}(\mathcal{U}_1)$. (If $\hat{U}_1 = 0$, then Slater's condition holds for (P) and we STOP.) Further, let

$$\mathcal{P}_1 := (\mathcal{F}(\mathcal{U}_1))^c \ (= \{\hat{U}_1\}^\perp \cap \mathcal{P}_0 \lhd \mathcal{P}_0).$$

We can now define the following equivalent program to (P) and its Lagrangian dual.

$$(\mathrm{RP}_1) \qquad \begin{array}{rll} p^* = & \max & c^t x \\ & \text{s.t.} & Ax \preceq_{\mathcal{P}_1} b \\ & & x \in \Re^m. \end{array}$$

$$(\mathrm{DRP}_1) \qquad \begin{array}{rll} d_1^* = & \min & \operatorname{trace} bU \\ & \text{s.t.} & A^*U = c \\ & & U \succeq_{(\mathcal{P}_1)^+} 0. \end{array}$$

Note that $p^* \leq d_1^* \leq d^*$. From Corollary 2.1 and Lemma 2.1 we conclude that

$$(\mathcal{P}_1)^+ = (\mathcal{P} \cap \mathcal{P}_1)^+ = \mathcal{P} + (\mathcal{P}_1)^\perp$$

so that

$$(\mathcal{P}_1)^+ = \mathcal{P} + ((\mathcal{F}(\mathcal{U}_1))^c)^\perp,\ (\mathcal{P}_1)^\perp = \mathcal{W}_1^S.$$

Therefore, we get the following equivalent program to (DRP$_1$).

$$(\text{ELSD}_1)\qquad \begin{array}{rll} d_1^* = & \min & \operatorname{trace} b(U + (W + W^t)) \\ & \text{s.t.} & A^*(U + (W + W^t)) = c \\ & & A^* U_1 = 0, \operatorname{trace} U_1 b = 0 \\ & & U \succeq 0, \begin{bmatrix} I & W^t \\ W & U_1 \end{bmatrix} \succeq 0. \end{array}$$

Step 2
We can now apply the same procedure to the program (RP$_1$). Since $\mathcal{W}_1^S = (\mathcal{P}_1)^\perp$, we get

$$\mathcal{U}_2 := \{U \succeq 0 : (U + V) \succeq_{(\mathcal{P}_1)^+} 0, A^*(U + V) = 0, \operatorname{trace}(U + V)b = 0\}.$$

Choose $\hat{U}_2 \in \operatorname{relint}(\mathcal{U}_2)$. (If $\hat{U}_2 = 0$, then the generalized Slater's condition holds for (RP$_1$) and we STOP.)

$$\mathcal{P}_2 := (\mathcal{F}(\mathcal{U}_2))^c \ (= \{\hat{U}_2\}^\perp \cap \mathcal{P}_1 \lhd \mathcal{P}_1).$$

We get a new equivalent program to (P) and its Lagrangian dual.

$$(\text{RP}_2)\qquad \begin{array}{rll} p^* = & \max & c^t x \\ & \text{s.t.} & Ax \preceq_{\mathcal{P}_2} b \\ & & x \in \Re^m. \end{array}$$

$$(\text{DRP}_2)\qquad \begin{array}{rll} d_2^* = & \min & \operatorname{trace} bU \\ & \text{s.t.} & A^* U = c \\ & & U \succeq_{(\mathcal{P}_2)^+} 0. \end{array}$$

We now have $p^* \leq d_2^* \leq d_1^* \leq d^*$. From Corollary 2.1 and Lemma 2.1 we conclude that

$$(\mathcal{P}_2)^+ = (\mathcal{P} \cap \mathcal{P}_2)^+ = \mathcal{P} + (\mathcal{P}_2)^\perp$$

and

$$(\mathcal{P}_2)^+ = \mathcal{P} + ((\mathcal{F}(\mathcal{U}_2))^c)^\perp,\ (\mathcal{P}_2)^\perp = \mathcal{W}_2^S.$$

Therefore, we get the following equivalent program to (DRP$_2$).

$$(\text{ELSD}_2)\qquad \begin{array}{rll} d_2^* = & \min & \operatorname{trace} b(U + (W + W^t)) \\ & \text{s.t.} & A^*(U + (W + W^t)) = c \\ & & A^* U_1 = 0, \operatorname{trace} U_1 b = 0 \\ & & A^*(U_2 + (W_1 + W_1^t)) = 0, \\ & & \operatorname{trace}(U_2 + (W_1 + W_1^t))b = 0 \\ & & U \succeq 0, \begin{bmatrix} I & W_1^t \\ W_1 & U_1 \end{bmatrix} \succeq 0 \\ & & \begin{bmatrix} I & W^t \\ W & U_2 \end{bmatrix} \succeq 0. \end{array}$$

... **Step k** ...

The remaining steps of the algorithm and the regularization are similar, and we see that after $k \leq \min\{m,n\}$ steps we obtain the equivalence of (RP) with (RP$_{\rm k}$), and (ELSD) with (ELSD$_{\rm k}$). The following theorem clarifies some of the relationships between the various sets.

THEOREM 4.1. *For some* $k \leq \min\{m,n\}$, *we have*

$$\mathcal{F}(\mathcal{U}_k) = (\mathcal{P}_k)^c, \text{ and } \mathcal{U}_1 \subset \mathcal{U}_2 \subset \cdots \subset \mathcal{U}_k = \cdots = \mathcal{U}_m = (\mathcal{P}^f)^c. \tag{4.3}$$

$$\mathcal{W}_k^s = (\mathcal{P}_k)^\perp = ((\mathcal{F}(\mathcal{U}_k))^c)^\perp, \quad \mathcal{W}_1^S \subset \cdots \subset \mathcal{W}_k^S = \cdots = \mathcal{W}_m^S = (\mathcal{P}^f)^\perp. \tag{4.4}$$

Proof. The nesting is clear from the definitions and is discussed in [33, Lemma 3] (for $\mathcal{W}_k$). Moreover, in [33, Lemma 2] it is shown that for $k \in \{1,2,\ldots,m\}$,

$$(b - Ax)U = 0 \text{ and } (b - Ax)W = 0 \ \forall x \in F, U \in \mathcal{U}_k, W \in \mathcal{W}_k.$$

Therefore, the inclusions in $(\mathcal{P}^f)^c, (\mathcal{P}^f)^\perp$ follow. Equality follows from the dimension of the feasible set, $F \subset \Re^m$, and a partial converse of Lemma 4.1; i.e., if $\mathcal{U}_k^c \neq \mathcal{P}^f$, then the system (4.1), with $U \neq 0$, is consistent. See [11, Corollary 7.1]. □

4.2. Duals of (D). Similar results can be obtained for the dual of (D); i.e., we can use the minimal cone to close the duality gap and we can get an explicit representation for the minimal cone. The extended Lagrange–Slater dual of the dual (D) is

$$\text{(ELSDD)} \qquad \begin{array}{rcc} d^* = & \max & \operatorname{trace} c^t x \\ & \text{subject to} & A(x + (Z + Z^t)) \preceq b \\ & & Z \in \mathcal{Z}_m, \end{array}$$

for $\mathcal{Z}_m$ to be derived below.

We can reformulate the dual (D) to the form of (P), i.e., define the cone

$$S = \Re^m \times \mathcal{P}, \ (S^+ = \{0\}^m \times \mathcal{P})$$

and the constraint operator $G : \Re^m \times \mathcal{S}_n \to \mathcal{S}_n$

$$G\begin{pmatrix} x \\ V \end{pmatrix} := Ax + V, \quad G^*U = \begin{pmatrix} A^*U \\ U \end{pmatrix}.$$

The dual (D) is equivalent to

$$\text{(ED)} \qquad \begin{array}{rcc} d^* = & \min & \operatorname{trace} bU \\ & \text{subject to} & G^*U \succeq_{S^+} \begin{pmatrix} c \\ 0 \end{pmatrix}. \end{array}$$

We have the following equivalence to Lemma 4.1.

LEMMA 4.2. *Suppose* $S_D^f \lhd K \lhd S^+$. *The system*

$$\phi = \begin{pmatrix} x \\ Ax \end{pmatrix} \succeq_{K^+} 0, \ \operatorname{trace} x^t c = 0 \tag{4.5}$$

is consistent only if

$$\text{(4.6)} \qquad \textit{the minimal cone } S^f_D \subset (\{\phi\}^\perp \cap K) \lhd K.$$

Proof. Suppose that ϕ is found from (4.5) and $U \in F_D$. Now

$$\begin{aligned}\left\langle \phi, G^*U - \begin{pmatrix} c \\ 0 \end{pmatrix} \right\rangle &= x^t(A^*U - c) + \operatorname{trace} U(Ax) \\ &= -x^tc + \operatorname{trace} U(Ax - Ax) = 0,\end{aligned}$$

since $x^tc = 0$. We get $G(F_D) - \left(\begin{smallmatrix} c \\ 0 \end{smallmatrix}\right) \subset \phi^\perp$, i.e., the minimal cone $S^f_D \subset \{\phi\}^\perp$. Finally, the fact that $\{\phi\}^\perp \cap K$ is a face of K follows from $\phi \in K^+$; i.e., $\{\phi\}^\perp$ is a supporting hyperplane containing S^f. □

The faces of S and S^+ directly correspond to faces of $\mathcal{P}$.

LEMMA 4.3.

1. *If $D \subset S^+$, then $\mathcal{F}(D) = 0 \times K$, where $K \lhd \mathcal{P}$.*
2. *If $D \subset S$, then $\mathcal{F}(D) = \Re^m \times K$, where $K \lhd \mathcal{P}$.*

Proof. The statements follow from the definitions. □

We also need a result similar to Lemma 2.1.

LEMMA 4.4. *Suppose that D is a convex cone and $D \subset S$. Let*

$$K := \left\{ \begin{pmatrix} x \\ W + W^t \end{pmatrix} : x \in \Re^m,\ U \succeq WW^t \textit{ for some } \begin{pmatrix} y \\ U \end{pmatrix} \in D \right\}.$$

Then

$$\begin{aligned} K &= ((\mathcal{F}(D))^c)^\perp \\ &= \left\{ \begin{pmatrix} x \\ W + W^t \end{pmatrix} : \begin{bmatrix} I & W^t \\ W & U \end{bmatrix} \succeq 0 \textit{ for some } \begin{pmatrix} y \\ U \end{pmatrix} \in D \right\}.\end{aligned}$$

Proof. The proof is very similar to the proof of Lemma 2.1. The difference is that we have to account for the cone S^+ being the direct sum $0^m \times \mathcal{P}$. We include the details for completeness.

Suppose that $\left(\begin{smallmatrix} x \\ W + W^t \end{smallmatrix}\right) \in K$, i.e., $U \succeq WW^t$ for some $\left(\begin{smallmatrix} y \\ U \end{smallmatrix}\right) \in D$. Then there exists a matrix H such that $W = UH$; see (2.5). Therefore, $\operatorname{trace} WV = 0$ for all $\left(\begin{smallmatrix} 0 \\ V \end{smallmatrix}\right) \in (\mathcal{F}(D))^c \subset S^+$; i.e.,

$$\begin{pmatrix} x \\ W + W^t \end{pmatrix} \in ((\mathcal{F}(D))^c)^\perp.$$

To prove the converse, suppose that $\left(\begin{smallmatrix} x \\ V \end{smallmatrix}\right) \in ((\mathcal{F}(D))^c)^\perp$ and $\left(\begin{smallmatrix} y \\ U \end{smallmatrix}\right) \in D \cap \operatorname{relint}(\mathcal{F}(D))$. Let U be orthogonally diagonalized by $Q = [Q_1 Q_2]$:

$$U = Q^t \operatorname{Diag}(d_1\ 0) Q,\ Q^tQ = I,$$

with $Q_1, n \times r,\ d_1 > 0$. Therefore,

$$\mathcal{F}(D) = \left\{ \begin{pmatrix} x \\ Q_1 B Q_1^t \end{pmatrix} : B \succeq 0,\ B \in \mathcal{S}_r,\ x \in \Re^m \right\}$$

and

$$(\mathcal{F}(D))^c = \left\{ \begin{pmatrix} 0 \\ Q_2 B Q_2^t \end{pmatrix} : B \succeq 0,\ B \in \mathcal{S}_{n-r}, 0 \in \{0\}^m \right\}.$$

Now

$$\begin{pmatrix} x \\ V \end{pmatrix} \in ((\mathcal{F}(D))^c)^\perp$$

implies that

$$0 = \operatorname{trace} V Q_2 B Q_2^t = \operatorname{trace} Q_2^t V Q_2 B \ \forall B \succeq 0,$$

i.e.,

$$Q_2^t V Q_2 = 0.$$

This implies that $Q_2 Q_2^t V Q_2 Q_2^t = 0$ as well. Note that $Q_2 Q_2^t$ is the orthogonal projection onto $\mathcal{N}(U)$. Therefore, the nonzero eigenvalues of V correspond to eigenvectors in the eigenspace formed from the column space of Q_1. Since the same must be true for VV^t, this implies that $\alpha U \succeq VV^t$ for some $\alpha > 0$ large enough; i.e., $V \in K$. □

Now define the following sets:

$$\begin{aligned} \mathcal{D}_k &= \{(V_i, Z_i)_{i=1}^k : Ax_i + (Z_{i-1} + Z_{i-1}^t) \succeq 0,\ x_i^t c = 0, \\ &\qquad V_i = Ax_i,\ V_i \succeq Z_i Z_i^t \ \forall i = 1, \ldots, k, Z_0 = 0\} \\ \mathcal{V}_k &= \{V_k : (V_i, Z_i)_{i=1}^k \in \mathcal{D}_k\} \\ \mathcal{Z}_k &= \{Z_k : (V_i, Z_i)_{i=1}^k \in \mathcal{D}_k\}. \end{aligned}$$

The extended Lagrange–Slater dual of the dual (D) can now be stated.

$$\text{(ELSDD)} \qquad \begin{array}{rcc} d^* = & \max & \operatorname{trace} c^t x \\ & \text{subject to} & A(x + (Z + Z^t)) \preceq b \\ & & Z \in \mathcal{Z}_m. \end{array}$$

Step 1
Define $T_0 := S^+$ and $\mathcal{P}_0 := \mathcal{P}$ and note that, since $Z_0 = 0$,

$$\begin{aligned} \mathcal{V}_1 &:= \left\{ Ax : \phi = \begin{pmatrix} x \\ Ax \end{pmatrix}, \phi \succeq_{T_0^+} 0,\ x^t c = 0 \right\} \\ &= \{V : V = Ax \succeq 0,\ x^t c = 0\}. \end{aligned}$$

Choose $\hat{V}_1 \in \operatorname{relint}(\mathcal{V}_1)$. (If $\hat{V}_1 = 0$, then the generalized Slater's condition holds for (ED) and we STOP.) Further, let

$$T_1 := (\mathcal{F}(\mathcal{V}_1))^c \ (= \{\hat{V}_1\}^\perp \cap T_0 \lhd T_0).$$

Therefore,

$$T_1 = \{0\}^m \times \mathcal{P}_1,$$

thus defining the face $\mathcal{P}_1 \lhd \mathcal{P}_0$.

We can now define the following equivalent program to (ED) and its Lagrangian dual.

$$(\mathrm{RED}_1)\qquad \begin{array}{rcl} d^* = & \min & \operatorname{trace} bU \\ & \text{s.t.} & A^*U = c \\ & & U \succeq_{\mathcal{P}_1} 0 \\ & \text{or} & G^*U \succeq_{T_1} \begin{pmatrix} c \\ 0 \end{pmatrix}. \end{array}$$

$$(\mathrm{DRED}_1)\qquad \begin{array}{rcl} p_1^* = & \max & c^t x \\ & \text{subject to} & Ax \preceq_{(\mathcal{P}_1)^+} b \\ & & x \in \Re^m \\ & \text{or} & G\phi =_{T_1^+} b,\ \phi \succeq_{T_1^+} 0. \end{array}$$

Note that $p^* \le p_1^* \le d^*$. From Corollary 2.1 we conclude that

$$(\mathcal{P}_1)^+ = (\mathcal{P} \cap \mathcal{P}_1)^+ = \mathcal{P} + (\mathcal{P}_1)^\perp$$

so that

$$(T_1)^+ = S + \left((\mathcal{F}(\mathcal{V}_1))^c\right)^\perp.$$

Therefore, Lemma 4.4 yields the following equivalent SDP to (DRED$_1$).

$$(\mathrm{ELSDD}_1)\qquad \begin{array}{rcl} p_1^* = & \max & c^t x \\ & \text{s.t.} & Ax + (Z + Z^t) \preceq b \\ & & Ay \succeq 0, c^t y = 0 \\ & & \begin{bmatrix} I & Z^t \\ Z & Ay \end{bmatrix} \succeq 0. \end{array}$$

Step 2

We can now apply the same procedure to the program (RED$_1$).

$$\begin{aligned} \mathcal{V}_2 := & \left\{ Ax : \phi = \begin{pmatrix} x \\ Ax \end{pmatrix}, \phi \succeq_{T_1^+} 0,\ x^t c = 0 \right\} \\ = & \{V : V = Ax \succeq_{\mathcal{P}_1} 0,\ x^t c = 0\}. \end{aligned}$$

Choose $\hat{V}_2 \in \operatorname{relint}(\mathcal{V}_2)$. (If $\hat{V}_2 = 0$, then the generalized Slater's condition holds for (DRP$_1$) and we STOP.) Let

$$T_2 := (\mathcal{F}(\mathcal{V}_2))^c \ \ (= \{\hat{V}_2\}^\perp \cap T_1 \lhd T_1).$$

We get a new equivalent program to (D) and its Lagrangian dual.

$$(\mathrm{RED}_2)\qquad \begin{array}{rcl} d^* = & \min & \operatorname{trace} bU \\ & \text{s.t.} & A^*U = c \\ & & U \succeq_{\mathcal{P}_2} 0 \\ & \text{or} & G^*U \succeq_{T_2} \begin{pmatrix} c \\ 0 \end{pmatrix}. \end{array}$$

$$(\mathrm{DRED}_2)\qquad \begin{array}{rcl} p_2^* = & \max & c^t x \\ & \text{subject to} & Ax \preceq_{(\mathcal{P}_2)^+} b \\ & & x \in \Re^m \\ & \text{or} & G\phi =_{T_2^+} b,\ \phi \succeq_{T_2^+} 0. \end{array}$$

We now have $p^* \leq p_1^* \leq p_2^* \leq d^*$. From Corollary 2.1 we get

$$(\mathcal{P}_2)^+ = (\mathcal{P} \cap \mathcal{P}_2)^+ = \mathcal{P} + (\mathcal{P}_2)^\perp$$

so that

$$(T_2)^+ = S + ((\mathcal{F}(\mathcal{V}_2))^c)^\perp.$$

Therefore, Lemma 4.4 yields the following equivalent SDP to (DRP$_2$).

$$(\mathrm{ELSDD}_2) \qquad \begin{array}{rll} p_2^* = & \max & c^t x \\ & \text{s.t.} & Ax + (Z + Z^t) \preceq b \\ & & Ay + (Z + Z^t) \succeq 0, c^t y = 0 \\ & & \begin{bmatrix} I & Z^t \\ Z & Ay \end{bmatrix} \succeq 0 \\ & & Ay_1 + (Z_1 + Z_1^t) \succeq 0, c^t y = 0 \\ & & \begin{bmatrix} I & Z_1^t \\ Z_1 & Ay_1 \end{bmatrix} \succeq 0. \end{array}$$

... **Step k** ...

5. Homogenization.

In section 3.1.1, we have shown that an ordinary linear programming problem can have an infinite number of dual programs for which strong duality holds. This includes the standard Lagrangian dual. However, this is not the case for SDP. First, the standard Lagrangian dual can result in a duality gap; see [33, Example 1]. Moreover, the duality gap may be 0, but the dual may not be attained, see [33, Example 5].

However, we have seen that the two equivalent duals (DRP) and (ELSD) both provide a zero duality gap and dual attainment, i.e., strong duality. Since LP is a special case of SDP ($\Re^n_+$ arises as the direct sum of n 1×1 semidefinite cones), we conclude that there are examples of SDP where there are many duals for which strong duality holds. A natural question to ask is whether there is any type of uniqueness for the strong duals, and, among the strong duals, what is the "strongest"; i.e., which is the "closest" to the standard Lagrangian dual.

Therefore, we now look at general optimality conditions for (P). We do this by using the homogenized semidefinite program (assume the optimal objective function value p^* is known):

$$(\mathrm{HP}) \qquad \begin{array}{rllr} 0 = & \max & c^t x + t(-p^*) & (= \langle a, w \rangle) \\ & \text{subject to} & Ax + t(-b) + Z = 0 & (Bw = 0) \\ & & w \in K = \Re^m \times \Re_+ \times \mathcal{P} & \left(w = \begin{pmatrix} x \\ t \\ Z \end{pmatrix} \right). \end{array}$$

The above defines the vector a, the linear operator B, and the convex cone K. Let F_H denote the feasible set, i.e.,

$$F_H = \mathcal{N}(B) \cap K,$$

where $\mathcal{N}$ denotes null space.

Note that if $t = 0$ in a feasible solution of (HP), then $B(\alpha w) = 0$ for all $\alpha \in \Re$, and

$$w = \begin{pmatrix} x \\ 0 \\ Z \end{pmatrix}.$$

Therefore, $c^t x > 0$ implies that $p^* = \infty$ (since there exists x such that $Ax \preceq 0$, $c^t x > 0$ implies (P) is unbounded). If $t > 0$ in a feasible solution of (HP), then

$$w = \begin{pmatrix} \frac{1}{t}x \\ 1 \\ \frac{1}{t}Z \end{pmatrix}$$

is feasible, which implies that $c^t x + t(-p^*) \leq 0$. Therefore,

$$Bw = 0, w \in K \text{ implies } \langle a, w \rangle \leq 0. \tag{5.1}$$

This shows that 0 is in fact the optimal value of (HP), and (HP) is an equivalent problem to (P).

One advantage of (HP) is that we know a feasible solution, namely, the origin. Recall the *polar* of a set C:

$$C^+ = \{\phi : \langle \phi, c \rangle \geq 0 \forall c \in C\}.$$

With this definition, the optimality conditions for (HP) are simply that the negative of the gradient of the objective function is in the polar of the feasible set; i.e., from (5.1) we conclude that

$$a = \begin{pmatrix} c \\ -p \\ 0 \end{pmatrix} \in -(\mathcal{N}(B) \cap K)^+ \quad \begin{pmatrix} \text{optimality} \\ \text{conditions} \\ \text{for HP} \end{pmatrix}. \tag{5.2}$$

This yields the asymptotic optimality conditions (up to closure):

$$\begin{pmatrix} c \\ -p \\ 0 \end{pmatrix} \in -(\overline{\mathcal{R}(B^*) + K^+}), \tag{5.3}$$

where the adjoint operator

$$B^*U = \begin{pmatrix} A^*U \\ -\operatorname{trace} bU \\ U \end{pmatrix}$$

and the polar cone

$$K^+ = \{0\} \times \Re_+ \times \mathcal{P}.$$

We have used the fact that the polar of the intersection of sets is the closure of the sum of the polars of the sets and that $\mathcal{P}$ is self-polar; i.e., $\mathcal{P} = \mathcal{P}^+$. Note that if the closure in (5.3) is not needed, then these optimality conditions, along with weak duality for (P) and (D), $p \leq \operatorname{trace} bU$, yield optimality conditions for (P); i.e., (5.3) with closure is equivalent to

$$\begin{pmatrix} c \\ -p \\ 0 \end{pmatrix} = \begin{pmatrix} A^*U \\ -\operatorname{trace} bU \\ U \end{pmatrix} - \begin{pmatrix} 0 \\ \alpha \\ V \end{pmatrix} \begin{pmatrix} \text{dual feasibility} \\ \text{strong duality} \\ \text{dual feasibility} \end{pmatrix} \tag{5.4}$$

for some $\alpha \geq 0,\ V \succeq 0$. This yields the optimality conditions for (P):

$$\begin{aligned} A^*U = c, U \succeq 0 &(\text{dual feasibility}), \\ p = \operatorname{trace} bU &\text{ (strong duality)}. \end{aligned}$$

(Note that strong duality is equivalent to complementary slackness.) We have proved the following.

THEOREM 5.1. *$p \in \Re$ is the optimal value of* (P) *if and only if* (5.3) *holds. Moreover, suppose that* (5.3) *holds but*

$$\begin{pmatrix} c \\ -p \\ 0 \end{pmatrix} \notin \mathcal{R}(B^*) - K^+. \tag{5.5}$$

Then p is still the optimal value of (P)*, but either there is a duality gap or the dual* (D) *is unattained*; *i.e., strong duality fails for* (P) *and* (D). □

The above theorem provides a way of generating examples where strong duality fails; i.e., we need to find examples where the right-hand side of (5.5) is not closed, and then we can pick a vector that is in the closure but not the preclosure.

There are many conditions, called *constraint qualifications*, that guarantee the closure condition in (5.3). In fact, this closure has been referred to as a *weakest constraint qualification*, [21, 37]. As an example of a closure condition, see, e.g., [23, pp. 104–105]. If C, D are closed convex sets and the intersection of their recession cones is $\{0\}$, then $D - C$ is closed. (Here the recession cone of a convex set C is the set of all points x such that $x + C \subset C$.) Therefore, for a subspace $\mathcal{V}$ and a convex cone K,

$$\mathcal{V} \cap K = \{0\} \text{ implies } \mathcal{V} + K \text{ is closed.}$$

In our case, several conditions for the closure (constraint qualifications) are given in [13, Theorem 3.1]. For example, the cone generated by the set $F_H - K$ is the whole space or Slater's condition

$$\exists \hat{x} \in F \text{ such that } A\hat{x} \prec b.$$

One approach to guarantee the closure condition is to find sets, T, to add to attain the closure. Equivalently, find sets, C, $C^+ = T$, to intersect with K to attain the closure so that

$$(\mathcal{N}(B) \cap K)^+ = (\mathcal{N}(B) \cap (K \cap C))^+ = \mathcal{R}(B^*) + K^+ + C^+. \tag{5.6}$$

On the other hand, note that the following is always true:

$$(\mathcal{N}(B) \cap (K \cap C))^+ = \overline{\mathcal{R}(B^*) + K^+ + C^+}.$$

There are some trivial choices for the set, e.g., $C = \mathcal{N}(B) \cap K$. Another choice would be $(\mathcal{N}(B) \cap K)^f$.

The above translates into choosing sets that contain the minimal cone $\mathcal{P}^f$. Since we want a small set of dual multipliers, we would like to find large sets that contain $\mathcal{P}^f$ but for which the above closure conditions hold. Some SDPs can be decomposed into parts, a linear part and a nonlinear part. Multipliers for the linear part correspond to linear programming; i.e., we choose the standard set of multipliers. However, we cannot choose a smaller set than $(\mathcal{P}^f)^+$ for the nonlinear part. (For a similar result, see, for instance, Boyd et al. [14, pp. 31–32].)

Suppose both problems (P) and (D) have feasible solutions (so that if there is a duality gap then it is finite). Consider the set

$$\mathcal{Z} = \{Z \in \mathcal{P} : \ Z = b - Ax \text{ for some } x \in \Re^m\}.$$

If $\mathcal{Z} \cap int(\mathcal{P}) \neq \emptyset$ then we have an interior point and strong duality holds for the Lagrangian dual. Otherwise, $\mathcal{Z} \subset \partial\mathcal{P}$. In particular, there exists a permutation matrix P and a block diagonal matrix structure in $\mathcal{S}_n$ such that $Z \in \mathcal{Z}$ implies that $P\,Z\,P^T$ is a block diagonal matrix which lies in the subspace defined by the block diagonal structure. We pick P such that each of the blocks has one of the following properties:

- Type I blocks: Block i is an LP (that is, the block matrix is a diagonal matrix). In this case strong duality holds for many duals including the Lagrangian dual.
- Type II blocks: Block i is not an LP, but (5.3) holds and (5.5) does not hold. In this case strong duality holds for many duals including the Lagrangian dual.
- Type III blocks: Block i is not an LP, but conditions (5.3) and (5.5) both hold. In this case, we can find linear objective functions for which (D) is feasible but strong duality does not hold for the Lagrangian dual.

In the case where the objective function is separable with respect to this partition, the duality for Type I and Type II blocks is well understood. For Type III blocks we showed that as long as (5.3) and (5.5) hold, there will be objective functions for which (D) is feasible, yet strong duality does not hold for (P) and (D). The reader may find it useful to generate examples by taking direct sums of examples from Freund [17] and Ramana [32].

Finally, we make some remarks about the ramifications of these results. We assumed throughout that (P) is feasible. Under this assumption, (ELSD) is feasible if and only if $p^* < +\infty$. If we also assume that $p^* < +\infty$, then we have $d^* = p^*$ (here, d^* is the optimal value of (ELSD)) and d^* is attained. We showed that in the dual problem (ELSD), the set W_m is precisely the subspace $(\mathcal{P}^f)^\perp$. Let us consider the following family of problems parameterized by $M > 0$:

$$(\tilde{\mathrm{P}}_{\mathrm{M}}) \qquad \begin{array}{cc} \sup & c^t x \\ \text{subject to} & Ax \preceq b \\ & A^*(I)^t x \leq M - \operatorname{trace}(b) \\ & x \in \Re^m, \end{array}$$

$$(\mathrm{ELS}\tilde{\mathrm{D}}_{\mathrm{M}}) \qquad \begin{array}{cc} \min & \operatorname{trace} b(U+W) + (M - \operatorname{trace}(b))z \\ \text{subject to} & A^*(U+W) - zA^*(I) = c \\ & W \in \mathcal{W}_m = (\mathcal{P}^f)^\perp \\ & U \succeq 0,\ z \geq 0. \end{array}$$

PROPOSITION 5.1. *Suppose* (P) *is feasible and* $p^* < +\infty$. *Then there exists a feasible solution* $(\tilde{U}, \tilde{W}, \tilde{z})$ *of* $(\mathrm{ELS}\tilde{\mathrm{D}}_{\mathrm{M}})$ *such that* $\tilde{U} \succ 0,\ \tilde{z} > 0$. *Moreover, for a given* M, *there exist optimal solutions of* (P) *with* $\operatorname{trace}(b - Ax) \leq M$ *if and only if there exist optimal solutions of* $(\tilde{\mathrm{P}}_{\mathrm{M}})$ *and every optimal solution of* $(\tilde{\mathrm{P}}_{\mathrm{M}})$ *is an optimal solution of* (P).

Proof. We apply the strong duality theorem to the pair (P) and (ELSD) to establish the existence of $(\bar{U}, \bar{W})$ such that $A^*(\bar{U} + \bar{W}) = c$, $\bar{W} \in \mathcal{W}_m$, and $\bar{U} \succeq 0$. Now, defining $\tilde{U} := \bar{U} + I$, $\tilde{W} := \bar{W}$, $\tilde{z} := 1$, we see that the first part of the proposition is proved. The second part of the proposition easily follows from the definition of $(\tilde{\mathrm{P}}_{\mathrm{M}})$. □

6. Conclusion. In this paper we have studied dual programs that guarantee strong duality for SDP. In particular, we have seen the relationships that exist between (DRP) (the dual of the regularized primal program (RP)) and (ELSD) (the extended

Lagrange–Slater dual). (DRP) uses the minimal cone $\mathcal{P}^f$ which, in general, cannot be computed exactly. (ELSD) shows that a regularized dual can be written down explicitly.

The pair (P) and (D) are the usual pair of dual programs used in SDP. This yields primal–dual interior-point methods when both programs satisfy the Slater CQ, i.e., strict feasibility. However, there are classes of problems where the CQ fails; see e.g., [39]. These problems arise from relaxations of 0,1 combinatorial optimization problems with linear constraints. In fact, for these problems, the Slater CQ fails for the primal while it is satisfied for the dual. Therefore, in theory, there is no duality gap between (P) and (D).

However, one can question whether (D) is still the true dual of (P) in this case. It is true that perturbations in b will yield the dual value d^* as the perturbations go to 0 when we can guarantee that we maintain the semidefinite constraint exactly. If we could do this, then we could solve any SDP independent of any regularity condition; i.e., we would only have to solve a perturbed dual to get the optimum value of the primal. However, the key here is that we cannot maintain the semidefinite constraint exactly; i.e., (D) is not a true dual of (P) in this case. It is the dual with respect to perturbations in the equality constraint $Ax + Z = b$ but not if we allow perturbations in the constraint $Z \succeq 0$ as well (i.e., not if we replace $Z \succeq 0$ by a nonnegativity constraint on the smallest eigenvalue $\lambda_{\min}(Z) \geq 0$).

Unlike LP, the solutions and optimal values of SDP may be doubly exponential rational numbers or even irrational. Note that the optimal value being doubly exponential means that the size (the number of bits required to express the value in binary) is an exponential function of the size of the input problem (P). However, in some cases it may be possible to find, a priori, upper bounds on the sizes of some primal and dual optimal solutions. Alizadeh [1] suggests that it may even be possible to bound the feasible solution sets of (P) and (D) a priori. Nevertheless, this is impossible even for an LP. For if the feasible region of (P) is bounded then the feasible region of (D) is unbounded and vice versa. Hence, one cannot hope to solve an SDP to exact optimality or, for that matter, find feasible solutions of semidefinite inequality systems in polynomial time. However, a challenging open problem is to determine if a given rational semidefinite system has a solution. This problem is called the semidefinite feasibility problem (SDFP). In [33] it was shown, by using (ELSD), that SDFP is not NP-complete unless NP=Co-NP.

It may be interesting to try to interpret the significance of (ELSD) in terms of the computational complexity of solving SDPs which do not satisfy the Slater CQ. We do have a dual program, (ELSD), that can be written down in polynomial time. However, we still do not know how to solve (P) and (ELSD) in polynomial time by a symmetric, primal–dual interior-point algorithm.

Acknowledgments. The second and third authors thank the National Science and Engineering Research Council Canada for their support.

REFERENCES

[1] F. Alizadeh, *Interior point methods in semidefinite programming with applications to combinatorial optimization*, SIAM J. Optim., 5 (1995), pp. 13–51.

[2] E. Anderson and P. Nash, *Linear Programming in Infinite Dimensional Spaces*, John Wiley and Sons, New York, 1987.

[3] G. P. Barker, *The lattice of faces of a finite dimensional cone*, Linear Algebra Appl., 7 (1973), pp. 71–82.

[4] G. P. Barker and D. Carlson, *Cones of diagonally dominant matrices*, Pacific J. Math., 57 (1975), pp. 15–32.

[5] W. W. Barrett, C. R. Johnson, and R. Loewy, *The Real Positive Definite Completion Problem: Cycle Completability*, Technical report, Department of Mathematics, College of William and Mary, Williamsburg, VA, 1993.

[6] A. Ben-Israel, *Linear equations and inequalities on finite dimensional, real or complex, vector spaces: A unified theory*, J. Math. Anal. Appl., 27 (1969), pp. 367–389.

[7] A. Ben-Tal and A. Nemirovskii, *Potential reduction polynomial time method for truss topology design*, SIAM J. Optim., 4 (1994), pp. 596–612.

[8] A. Berman, *Cones, Matrices and Mathematical Programming*, Springer-Verlag, Berlin, New York, 1973.

[9] F. Bohnenblust, *Joint Positiveness of Matrices*, California Institute of Technology, Pasadena, CA, 1948, manuscript.

[10] J. M. Borwein and H. Wolkowicz, *Characterizations of optimality for the abstract convex program with finite dimensional range*, J. Austral. Math. Soc. Ser. A, 30 (1981), pp. 390–411.

[11] J. M. Borwein and H. Wolkowicz, *Regularizing the abstract convex program*, J. Math. Anal. Appl., 83 (1981), pp. 495–530.

[12] J. M. Borwein and H. Wolkowicz, *Characterizations of optimality without constraint qualification for the abstract convex program*, Math. Programming Study, 19 (1982), pp. 77–100.

[13] J. M. Borwein and H. Wolkowicz, *A simple constraint qualification in infinite dimensional programming*, Math. Programming, 35 (1986), pp. 83–96.

[14] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*, in Studies in Applied Mathematics 15, SIAM, Philadelphia, PA, 1994.

[15] C. Delorme and S. Poljak, *The performance of an eigenvalue bound on the max-cut problem in some classes of graphs*, Discrete Math., 111 (1993), pp. 145–156.

[16] R. J. Duffin, *Infinite programs*, in Linear Equalities and Related Systems, A. W. Tucker, ed., Princeton University Press, Princeton, NJ, 1956, pp. 157–170.

[17] R. M. Freund, *Complexity of an Algorithm for Finding an Approximate Solution of a Semidefinite Program with No Regularity Assumption*, Technical report OR 302-94, MIT, Cambridge, MA, 1994.

[18] K. Glashoff and S. Gustafson, *Linear optimization and approximation*, in Applied Mathematical Sciences 45, Springer-Verlag, Verlag Basel, 1978.

[19] M. X. Goemans and D. P. Williamson, *Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming*, Technical report, Department of Mathematics, MIT, 1994.

[20] B. Grone, C. Johnson, E. Marques De Sa, and H. Wolkowicz, *Positive definite completions of partial Hermitian matrices*, Linear Algebra Appl., 58 (1984), pp. 109–124.

[21] M. Guignard, *Generalized Kuhn–Tucker conditions for mathematical programming problems in a banach space*, SIAM J. Control, 7 (1969), pp. 232–241.

[22] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz, *An interior point method for semidefinite programming*, SIAM J. Optim., 6 (1996), pp. 342–361.

[23] R. B. Holmes, *Geometric Functional Analysis and its Applications*, Springer-Verlag, Berlin, 1975.

[24] C. Johnson, B. Kroschel, and H. Wolkowicz, *An interior-point method for approximate positive semidefinite completions*, Comput. Optim. Appl., to appear.

[25] K. Kretschmer, *Programming in paired spaces*, Canad. J. Math., 13 (1961), pp. 221–238.

[26] N. Levinson, *A class of continuous linear programming problems*, J. Math. Anal. Appl., 16 (1966), pp. 73–83.

[27] L. Lovász and A. Schrijver, *Cones of matrices and set-functions and* 0–1 *optimization*, SIAM J. Optim., 1 (1991), pp. 166–190.

[28] K. Löwner, *Uber monotone matrixfunctionen*, Math. Z., 49 (1934), pp. 375–392.

[29] A. W. Marshall and I. Olkin, *Inequalities: Theory of Majorization and its Applications*, Academic Press, New York, NY, 1979.

[30] Y. E. Nesterov and A. S. Nemirovskii, *Interior Point Polynomial Algorithms in Convex Programming: Theory and Algorithms*, SIAM, Philadelphia, PA, 1994.

[31] F. Pukelsheim *Optimal Design of Experiments*, Wiley, New York, 1993.

[32] M. V. Ramana, *An Algorithmic Analysis of Multiquadratic and Semidefinite Programming Problems*, Ph.D. thesis, Johns Hopkins University, Baltimore, MD, 1993.

[33] M. V. Ramana, *An Exact Duality Theory for Semidefinite Programming and its Complexity Implications*, DIMACS Technical report 95-02R, RUTCOR, Rutgers University, New

Brunswick, NJ, 1995.

[34] T. W. Reiland, *Optimality conditions and duality in continuous programming.* ii. *The linear problem revisited*, J. Math. Anal. Appl., 77 (1980), pp.329–343.

[35] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[36] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95.

[37] H. Wolkowicz, *Geometry of optimality conditions and constraint qualifications: The convex case*, Math. Programming, 19 (1980), pp. 32–60.

[38] H. Wolkowicz, *Some applications of optimization in matrix theory*, Linear Algebra Appl., 40 (1981), pp. 101–118.

[39] Q. Zhao, S. Karisch, F. Rendl, and H. Wolkowicz, *Semidefinite Programming Relaxations for the Quadratic Assignment Problem*, Research report, University of Waterloo, Waterloo, Ontario, Canada, 1995.

SIAM J. Optim.
Vol. 7, No. 3, pp. 663–678, August 1997

PRIMAL–DUAL PATH-FOLLOWING ALGORITHMS FOR SEMIDEFINITE PROGRAMMING*

RENATO D. C. MONTEIRO†

Abstract. This paper deals with a class of primal–dual interior-point algorithms for semidefinite programming (SDP) which was recently introduced by Kojima, Shindoh, and Hara [*SIAM J. Optim.*, 7 (1997), pp. 86–125]. These authors proposed a family of primal-dual search directions that generalizes the one used in algorithms for linear programming based on the scaling matrix $X^{1/2}S^{-1/2}$. They study three primal–dual algorithms based on this family of search directions: a short-step path-following method, a feasible potential-reduction method, and an infeasible potential-reduction method. However, they were not able to provide an algorithm which generalizes the long-step path-following algorithm introduced by Kojima, Mizuno, and Yoshise [*Progress in Mathematical Programming: Interior Point and Related Methods*, N. Megiddor, ed., Springer-Verlag, Berlin, New York, 1989, pp. 29–47]. In this paper, we characterize two search directions within their family as being (unique) solutions of systems of linear equations in symmetric variables. Based on this characterization, we present a simplified polynomial convergence proof for one of their short-step path-following algorithms and, for the first time, a polynomially convergent long-step path-following algorithm for SDP which requires an extra $\sqrt{n}$ factor in its iteration-complexity order as compared to its linear programming counterpart, where n is the number of rows (or columns) of the matrices involved.

Key words. semidefinite programming, interior-point methods, polynomial complexity, path-following methods, primal–dual algorithms

AMS subject classifications. 65K05, 90C25, 90C30

PII. S1052623495293056

1. Introduction.

This paper studies primal–dual path-following algorithms for semidefinite programming (SDP) based on a search direction that has been proposed by Kojima, Shindoh, and Hara [11] as a natural extension of the one used in algorithms for linear programming based on the scaling matrix $X^{1/2}S^{-1/2}$. The first primal–dual algorithm for linear programming (LP) to use this scaling matrix was presented by Kojima, Mizuno, and Yoshise [10] and is referred in here to as the *long-step path-following method*. Another variant independently developed by Kojima, Mizuno, and Yoshise [9] and Monteiro and Adler [12, 13], referred to here as the *short-step path-following method*, improves the worst-case iteration complexity of the algorithm of [10] by a factor of $\sqrt{n}$ by generating iterates in a narrower neighborhood of the central path.

Several authors have discussed generalizations of interior-point algorithms for linear programming to the context of SDP. The landmark work in this direction is due to Nesterov and Nemirovskii [14, 15], where a general approach for using interior-point methods for solving convex programs is proposed based on the notion of self-concordant functions. (See their book [17] for a comprehensive treatment of this subject.) They show that the problem of minimizing a linear function over a convex set K can be solved in "polynomial time" as long as a self-concordant barrier function for K is known. In particular, Nesterov and Nemirovskii show that linear programs,

* Received by the editors October 11, 1995; accepted for publication (in revised form) April 2, 1996.

http://www.siam.org/journals/siopt/7-3/29305.html

† School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332 (monteiro@isye.gatech.edu). The work of this author was based on research supported by the National Science Foundation under grant INT-9600343 and the Office of Naval Research under grants N00014-93-1-0234 and N00014-94-1-0340.

convex quadratic programs with convex quadratic constraints, and semidefinite programs all have explicit and easily computable self-concordant functions and hence can be solved in "polynomial time." Subsequently, Alizadeh [2] extends in a direct way Ye's projective potential-reduction algorithm (see [21]) for LP to the context of SDP and argues that many known interior-point LP algorithms can also be transformed into an algorithm for SDP in a mechanical way. Since then, several authors have proposed interior-point algorithms for solving SDP problems, including Helmberg et al. [5], Jarre [8], Kojima, Shindoh, and Hara [11], Nesterov and Nemirovskii [16], Nesterov and Todd [19, 18] and Vandenberghe and Boyd [20].

Among the above works, Kojima, Shindoh, and Hara [11] and Nesterov and Todd [18] present some algorithms which extend the primal–dual methods for linear programming based on the scaling $X^{1/2}S^{-1/2}$. In particular, they both provide short-step path-following methods for SDP which generalize the algorithm in [9, 12, 13]; however, no extensions of the long-step path-following algorithm in [10] are provided. In fact, Kojima, Shindoh, and Hara mention in section 9 of [11] that they encountered difficulty in providing such an extension.

In this paper, by characterizing two of the search directions introduced in [11] as solutions of systems of linear equations in symmetric variables, we present a simplified polynomial convergence proof for a short-step path-following algorithm in [11] and for the first time, a polynomially convergent long-step path-following algorithm for SDP. We show that the long-step method requires $\mathcal{O}(n^{3/2}\log(t^0\epsilon^{-1}))$ iterations to generate a feasible solution with objective function within ϵ of the optimal value when initialized at an interior feasible point whose duality gap is t^0. Hence, the algorithm of [10] when extended to SDP has its iteration-complexity increased by a factor of $\sqrt{n}$.

This paper is organized as follows. In section 2, we describe the generic primal–dual algorithm for SDP which will be the subject of our study in this paper. Section 3 contains some matrix results that are frequently used in our presentation. Section 4 discusses the short-step path-following method for SDP while section 5 discusses its long-step counterpart.

1.1. Notation and terminology.

The following notation is used throughout the paper. The superscript ^{T} denotes transpose. $\Re^p$, $\Re^p_+$, and $\Re^p_{++}$ denote the p-dimensional Euclidean space, the nonnegative orthant of $\Re^p$, and the positive orthant of $\Re^p$, respectively. The ith component of a vector $u \in \Re^p$ is denoted by u_i. The set of all $p \times q$ matrices with real entries is denoted by $\Re^{p\times q}$. The (i,j)th entry of a matrix $Q \in \Re^{p\times q}$ is denoted by Q_{ij}. The set of all symmetric $p \times p$ matrices is denoted by $\mathcal{S}^{(p)}$ or, simply, by $\mathcal{S}$ when the dimension p is clear from the context. For $Q \in \mathcal{S}$, $Q \succeq 0$ ($Q \preceq 0$) means Q is positive (negative) semidefinite and $Q \succ 0$ ($Q \prec 0$) means Q is positive (negative) definite. The trace of a matrix $Q \in \Re^{p\times p}$ is denoted by $\operatorname{Tr} Q \equiv \sum_{i=1}^n Q_{ii}$. The eigenvalues of $Q \in \mathcal{S}^{(p)}$ are denoted by $\lambda_i(Q)$, $i = 1,\dots,p$, and its smallest and largest eigenvalues are denoted by $\lambda_{\min}(Q)$ and $\lambda_{\max}(Q)$, respectively. Given P and Q in $\Re^{p\times q}$, the inner product between them is defined as $P \bullet Q \equiv \operatorname{Tr} P^TQ = \sum_{i=1,j=1}^n P_{ij}Q_{ij}$. Given u and v in $\Re^p$, $u \le v$ means $u_i \le v_i$ for every $i = 1,\dots,p$. The Euclidean norm and its associated operator norm are both denoted by $\|\cdot\|$; hence, $\|Q\| \equiv \max_{\|u\|=1}\|Qu\|$ for any $Q \in \Re^{p\times p}$. The Frobenius norm of $Q \in \Re^{p\times p}$ is $\|Q\|_F \equiv (Q \bullet Q)^{1/2}$. $\mathcal{S}_+$ and $\mathcal{S}_{++}$ denote the set of all matrices in $\mathcal{S}$ which are positive semidefinite and positive definite, respectively. Finally, $\mathcal{S}_\perp^{(p)}$, or simply $\mathcal{S}_\perp$ when p is understood from the context, denote the set of all skew-symmetric matrices in $\Re^{p\times p}$. Since $\mathcal{S}^{(p)} + \mathcal{S}_\perp^{(p)} = \Re^{p\times p}$ and $U \bullet V = 0$ for every $U \in \mathcal{S}^{(p)}$ and $V \in \mathcal{S}_{\perp}^{(p)}$, it follows that $\mathcal{S}_{\perp}^{(p)}$ is the orthogonal complement of $\mathcal{S}^{(p)}$ with respect to the inner product $\bullet$.

2. The primal–dual algorithm and some technical results.

In this section we describe the generic primal–dual algorithm which will be the subject of our study in this paper. We then show that the search direction used by the generic algorithm is a particular one from the family of the search directions introduced in the revised version of [11]. We end the section by giving some basic results about the generic algorithm.

This paper studies primal–dual path-following algorithms for solving the *semidefinite programming problem* (SDP)

$$(1) \qquad (P) \qquad \min\{C \bullet X : A_i \bullet X = b_i,\, i = 1, \ldots, m,\, X \succeq 0\}$$

and its associated dual SDP

$$(2) \qquad (D) \qquad \max\left\{b^T y : \sum_{i=1}^{m} y_i A_i + S = C,\, S \succeq 0\right\},$$

where $C \in \Re^{n\times n}$, $A_i \in \Re^{n\times n}$, $i = 1, \ldots, m$, and $b = (b_1, \ldots, b_m)^T \in \Re^m$ are the data, and $X \in \mathcal{S}_{+}^{(n)}$ and $(S, y) \in \mathcal{S}_{+}^{(n)} \times \Re^m$ are the primal and dual variables, respectively. We assume without loss of generality that the matrices C and A_i, $i = 1, \ldots, m$, are symmetric (otherwise, replace C by $(C + C^T)/2$ and A_i by $(A_i + A_i^T)/2$).

The set of *interior feasible solutions* of (1) and (2) are

$$F^0(P) \equiv \{X \in \mathcal{S} : A_i \bullet X = b_i,\, i = 1, \ldots, m,\, X \succ 0\},$$
$$F^0(D) \equiv \left\{(S, y) \in \mathcal{S} \times \Re^m : \sum_{i=1}^{m} y_i A_i + S = C,\, S \succ 0\right\},$$

respectively. Throughout this paper, we assume that $F^0(P) \times F^0(D) \neq \emptyset$. Under this assumption, it is well known that both (1) and (2) have optimal solutions X^* and (S^*, y^*) such that $C \bullet X^* = b^T y^*$ (that is, the optimal values of (1) and (2) are equal). This last condition alternatively can be expressed as $X^* \bullet S^* = 0$, since for feasible solutions X and (S, y) for (1) and (2) there hold $C \bullet X - b^T y = (\sum_{i=1}^{n} y_i A_i + S) \bullet X - b^T y = X \bullet S + \sum_{i=1}^{n} y_i (A_i \bullet X) - b^T y = X \bullet S + \sum_{i=1}^{n} y_i b_i - b^T y = X \bullet S$. For simplicity, we will also assume that the matrices A_i, $i = 1, \ldots, m$ are linearly independent.

We next outline a generic interior-point primal–dual algorithm for solving the pair of SDPs (1) and (2) which was introduced in [11]. The system of linear equations defining the search direction in the following algorithm is actually different from the one used in [11], but the resulting search direction is the same as will be shown in Lemma 2.1.

Generic primal–dual Algorithm.

Step 0. Let $X^0 \in F^0(P)$ and $(S^0, y^0) \in F^0(D)$ be given and set $k = 0$.

Step 1. Let $X = X^k$, $(S, y) = (S^k, y^k)$ and $\mu = (X \bullet S)/n$;

Step 2. Choose a centrality parameter $\sigma = \sigma_k \in [0, 1]$ and set
$H \equiv (\sigma\mu I - X^{1/2} S X^{1/2})$;

Step 3. Compute the search direction $(\Delta X, \Delta S, \Delta y) \in \mathcal{S} \times \mathcal{S} \times \Re^m$
by solving the following system of linear equations:

$$X^{-1/2}(X\Delta S + \Delta X S)X^{1/2} + X^{1/2}(\Delta S X + S\Delta X)X^{-1/2} = 2H, \tag{3}$$

$$A_i \bullet \Delta X = 0, \quad \text{for all } i = 1, \ldots, m, \tag{4}$$

$$\sum_{i=1}^{m} \Delta y_i A_i + \Delta S = 0; \tag{5}$$

```
    Step 4. Choose a step-size $\alpha = \alpha_k \geq 0$ such that $\hat{X} \equiv X + \alpha\Delta X \in \mathcal{S}_{++}$,
            and $(\hat{S}, \hat{y}) \equiv (S, y) + \alpha(\Delta S, \Delta y) \in \mathcal{S}_{++} \times \Re^m$;
    Step 5. Let $X^{k+1} = \hat{X}$, $(S^{k+1}, y^{k+1}) = (\hat{S}, \hat{y})$, replace $k$ by $k+1$,
            and go to step (1).
End
```

In what follows, we show that the search direction used by the generic algorithm is a particular one from the family of the search directions introduced in the revised version of [11]. We first describe this family of search directions. Given a fixed $t \in [0, 1]$, Kojima et al. show that the system of linear equations consisting of (4), (5), and the equation

$$X(\Delta S + tW) + (\Delta X + (1-t)W)S = \sigma\mu I - XS, \tag{6}$$

has a unique solution $(\Delta X(t), \Delta S(t), \Delta y(t), W(t)) \in \mathcal{S} \times \mathcal{S} \times \Re^m \times \mathcal{S}_\perp$ (see Theorem 4.2 of [11]). The search direction for their algorithm is $(\Delta X(t), \Delta S(t), \Delta y(t))$ for some fixed $t \in [0, 1]$. (They have in fact introduced a larger family of search directions but this one suffices for the purpose of our discussion.) The following result shows that system (4), (5), and (6) with $t = 1$ determines exactly the same direction as system (3)–(5) does; that is, $(\Delta X(1), \Delta S(1), \Delta y(1)) = (\Delta X, \Delta S, \Delta y)$.

LEMMA 2.1. $(\Delta X(1), \Delta S(1), \Delta y(1))$ *is the unique solution of the system* (3)–(5).

Proof. Let $(\hat{\Delta}X, \hat{\Delta}S, \hat{\Delta}y, \hat{W}) \equiv (\Delta X(1), \Delta S(1), \Delta y(1), W(1))$. We first show that $(\hat{\Delta}X, \hat{\Delta}S, \hat{\Delta}y)$ is a solution of (3)–(5). It suffices to show that $(\hat{\Delta}X, \hat{\Delta}S, \hat{\Delta}y)$ satisfies (3). Indeed, by definition, $(\hat{\Delta}X, \hat{\Delta}S, \hat{\Delta}y)$ satisfies (6) with $t = 1$. After multiplying this relation on the left by $X^{-1/2}$ and on right by $X^{1/2}$, we obtain

$$X^{1/2}(\hat{\Delta}S + \hat{W})X^{1/2} + X^{-1/2}\hat{\Delta}X S X^{1/2} = \sigma\mu I - X^{1/2} S X^{1/2}.$$

Hence, the sum of the symmetric parts of the two terms on the left-hand side is equal to the right-hand side. This fact together with the fact that $\hat{W} + \hat{W}^T = 0$ imply

$$\begin{aligned}
&2\left(\sigma\mu I - X^{1/2} S X^{1/2}\right) \\
&= X^{1/2}(2\hat{\Delta}S + \hat{W} + \hat{W}^T)X^{1/2} + X^{-1/2}\hat{\Delta}X S X^{1/2} + X^{1/2} S \hat{\Delta}X X^{-1/2} \\
&= 2X^{1/2}\hat{\Delta}S X^{1/2} + X^{-1/2}\hat{\Delta}X S X^{1/2} + X^{1/2} S \hat{\Delta}X X^{-1/2} \\
&= X^{-1/2}(X\hat{\Delta}S + \hat{\Delta}X S)X^{1/2} + X^{1/2}(\hat{\Delta}S X + S\hat{\Delta}X)X^{-1/2}.
\end{aligned}$$

That is, $(\hat{\Delta}X, \hat{\Delta}S, \hat{\Delta}y)$ satisfies (3). To show that $(\hat{\Delta}X, \hat{\Delta}S, \hat{\Delta}y)$ is the only solution of (3)–(5), assume that $(\Delta X, \Delta S, \Delta y)$ is an arbitrary solution of (3)–(5) and let $E \equiv X^{-1/2}(X\Delta S + \Delta X S)X^{1/2}$. Then, by (3) we have $E + E^T = 2H$, and hence $W \equiv X^{-1/2}(H - E)X^{-1/2} = X^{-1/2}(E^T - E)X^{-1/2}/2$ is skew symmetric. A simple algebraic manipulation shows that $(\Delta X, \Delta S, \Delta y, W)$ satisfies (6) with $t = 1$ and, hence, that it is a solution of the system defined by (4), (5), and (6) with $t = 1$. Since $(\hat{\Delta}X, \hat{\Delta}S, \hat{\Delta}y, \hat{W})$ is the unique solution of this system in $\mathcal{S} \times \mathcal{S} \times \Re^m \times \mathcal{S}_\perp$, we conclude that $(\Delta X, \Delta S, \Delta y, W) = (\hat{\Delta}X, \hat{\Delta}S, \hat{\Delta}y, \hat{W})$. ◻

In a similar vein, it is possible to characterize $(\Delta X(0), \Delta S(0), \Delta y(0))$ as the unique solution in $\mathcal{S} \times \mathcal{S} \times \Re^m$ of the system of linear equations consisting of (4), (5), and the equation

$$S^{1/2}(X\Delta S + \Delta X S)S^{-1/2} + S^{-1/2}(\Delta S X + S\Delta X)S^{1/2} = 2(\sigma\mu I - S^{1/2}XS^{1/2}). \tag{7}$$

Results analogous to the ones proved in this paper easily can be obtained with respect to path-following algorithms based on this search direction.

It should be noted that the two systems of linear equations (3)–(5) and (3), (4), (7) were introduced for the first time in a preliminary version of this paper. The result stated in Lemma 2.1 was subsequently pointed out by Masakazu Kojima to the author in a personal communication. The present version of this paper is essentially a modification of the previous version which takes into account this important observation.

From the discussion above, we see that both directions $(\Delta X(0), \Delta S(0), \Delta y(0))$ and $(\Delta X(1), \Delta S(1), \Delta y(1))$ are solutions of systems of linear equations in symmetric matrices, a property which is also shared by the NT-direction introduced by Nesterov and Todd [18], namely, the unique solution $(\Delta X, \Delta S, \Delta y)$ of (4), (5), and the equation

$$\begin{aligned}(X^{1/2}SX^{1/2})^{1/2}X^{-1/2}\Delta X X^{-1/2}(X^{1/2}SX^{1/2})^{1/2}&\\ + X^{1/2}\Delta S X^{1/2} = \sigma\mu I - X^{1/2}SX^{1/2}.&\end{aligned} \tag{8}$$

But unlike the NT-direction, computing the directions $(\Delta X(t), \Delta S(t), \Delta y(t))$ do not require computation of matrix square roots, which is certainly an advantage from the computational point of view.

Another primal–dual search direction which has been considered by a few authors (see, for example, Adler and Alizadeh [1] and Alizadeh, Haeberly, and Overton [3]) is the one that is the solution of the linear system consisting of (4), (5), and the equation

$$X\Delta S + \Delta S X + S\Delta X + \Delta X S = 2\sigma\mu I - XS - SX. \tag{9}$$

At the time of this writing, no polynomial convergence has been proven for an algorithm based on this direction.

We end this section by stating the following straightforward result regarding the generic algorithm.

Lemma 2.2. *Let $X \in F^0(P)$ and $(S, y) \in F^0(D)$ be given and suppose that $(\Delta X, \Delta S, \Delta y)$ is a solution of (3)–(5) for some $H \in \Re^{n\times n}$. Then, the following statements hold:*

(a) *$\Delta S \bullet \Delta X = 0$,*

(b) *$X \bullet \Delta S + S \bullet \Delta X = Tr\ H$,*

(c) *if $H = \sigma\mu I - X^{1/2}SX^{1/2}$ where $\sigma \in \Re$ and $\mu \equiv (X \bullet S)/n$, then*

$$(X + \alpha\Delta X) \bullet (S + \alpha\Delta S) = (1 - \alpha + \alpha\sigma)(X \bullet S) \quad \forall \alpha \in \Re.$$

Proof. Using (4) and (5), we obtain

$$\Delta S \bullet \Delta X = -\left(\sum_{i=1}^{n} \Delta y_i A_i\right) \bullet \Delta X = -\sum_{i=1}^{n} \Delta y_i (A_i \bullet \Delta X) = 0,$$

and hence (a) follows. In view of (3), we have

$$\begin{aligned} 2\,\mathrm{Tr}\ H &= \mathrm{Tr}\ X^{-1/2}(X\Delta S + \Delta X S)X^{1/2} + \mathrm{Tr}\ X^{1/2}(\Delta S X + S\Delta X)X^{-1/2} \\ &= \mathrm{Tr}\ (X\Delta S + \Delta X S) + \mathrm{Tr}\ (\Delta S X + S\Delta X) \\ &= 2\,\mathrm{Tr}\ (X\Delta S + S\Delta X) \ = \ 2(X \bullet \Delta S + S \bullet \Delta X), \end{aligned}$$

and hence (b) follows. Using statements (a) and (b) and the fact that $H = \sigma\mu I - X^{1/2}SX^{1/2}$ and Tr $(X^{1/2}SX^{1/2}) = X \bullet S = n\mu$, we obtain

$$\begin{aligned}(X+\alpha\Delta X)\bullet(S+\alpha\Delta S) &= X\bullet S+\alpha(X\bullet\Delta S+S\bullet\Delta X)+\alpha^2(\Delta X\bullet\Delta S)\\ &= X\bullet S+\alpha\mathrm{Tr}\ \left(\sigma\mu I - X^{1/2}SX^{1/2}\right)\\ &= X\bullet S+\alpha\,(\sigma n\mu - X\bullet S)\\ &= (1-\alpha+\alpha\sigma)(X\bullet S)\end{aligned}$$

for every $\alpha \in \Re$. Hence, (c) holds. □

3. Some technical results about matrices. This section states some inequalities about matrices which play an important role in the convergence analysis of the algorithms presented in sections 4 and 5.

In the next result, we collect some useful facts about symmetric matrices. For its proof, we refer the reader to Golub and Van Loan [4] or Horn and Johnson [6].

LEMMA 3.1. *For any $E \in \mathcal{S}^{(p)}$, we have*

$$\lambda_{\max}(E) = \max_{\|u\|=1} u^T E u, \tag{10}$$

$$\lambda_{\min}(E) = \min_{\|u\|=1} u^T E u, \tag{11}$$

$$\|E\| = \max_{i=1,\ldots,p} |\lambda_i(E)|, \tag{12}$$

$$\|E\|_F^2 = \sum_{i=1}^{p} [\lambda_i(E)]^2. \tag{13}$$

The following result about general matrices is also useful.

LEMMA 3.2. *For any $W \in \Re^{p\times p}$, the following relations hold:*

$$\max_{i=1,\ldots,n} Re[\lambda_i(W)] \le \frac{1}{2}\lambda_{\max}(W+W^T), \tag{14}$$

$$\min_{i=1,\ldots,n} Re[\lambda_i(W)] \ge \frac{1}{2}\lambda_{\min}(W+W^T), \tag{15}$$

$$\sum_{i=1}^{p} |\lambda_i(W)|^2 \le \|W\|_F^2 = \|W^T\|_F^2, \tag{16}$$

$$\lambda_{\max}(W^TW) = \|W^TW\| \ = \ \|W\|^2 \ = \ \|W^T\|^2. \tag{17}$$

Proof. Inequality (14) is stated as an exercise in Horn and Johnson; see [7], page 187, exercise 20. Inequality (15) follows from (14) applied to the matrix $-W$. For a proof of (16) and (17), see Golub and Van Loan [4], pages 58 and 336. □

As a consequence of Lemma 3.2, we obtain the following result.

LEMMA 3.3. *Suppose that $W \in \Re^{p\times p}$ is a nonsingular matrix. Then, for any $E \in \mathcal{S}^{(p)}$, we have*

$$\lambda_{\max}(E) \le \frac{1}{2}\lambda_{\max}\left(WEW^{-1}+(WEW^{-1})^T\right), \tag{18}$$

$$\lambda_{\min}(E) \ge \frac{1}{2}\lambda_{\min}\left(WEW^{-1}+(WEW^{-1})^T\right), \tag{19}$$

$$\|E\| \le \frac{1}{2}\left\|WEW^{-1}+(WEW^{-1})^T\right\|, \tag{20}$$

$$\|E\|_F \leq \frac{1}{2}\,\|WEW^{-1} + (WEW^{-1})^T\|_F. \tag{21}$$

Proof. Using (14), we obtain

$$\lambda_{\max}(E) = \lambda_{\max}(WEW^{-1}) \leq \frac{1}{2}\lambda_{\max}\left(WEW^{-1} + (WEW^{-1})^T\right)$$

for every $E \in \Re^{p\times p}$, and hence (18) follows. Inequality (19) is proved in a similar way by using (15). Inequality (20) follows from (18), (19), and (12). To prove (21), we use (13) and (16) to get

$$\|E\|_F^2 = \sum_{i=1}^{p}[\lambda_i(E)]^2 = \sum_{i=1}^{p}[\lambda_i(WEW^{-1})]^2 \leq \|WEW^{-1}\|_F^2.$$

Hence, we obtain

$$\begin{aligned}4\,\|E\|_F^2 &\leq 2\,\|WEW^{-1}\|_F^2 + 2\|E\|_F^2 \;=\; 2\,\|WEW^{-1}\|_F^2 + 2\,\mathrm{Tr}\;E^2\\ &= 2\,\|WEW^{-1}\|_F^2 + 2\,\mathrm{Tr}\;WE^2W^{-1} \;=\; 2\,\|WEW^{-1}\|_F^2 + 2\,\mathrm{Tr}\;(WEW^{-1})^2\\ &= \|WEW^{-1} + (WEW^{-1})^T\|_F^2,\end{aligned}$$

which clearly implies (21). □

We observe that (20) is not needed in our presentation, but it could be useful in proving polynomial convergence of other primal–dual variants not studied in this paper. The other inequalities in Lemma 3.3 play a crucial role in the analysis of the short-step and the long-step path-following methods of sections 4 and 5, respectively.

4. Short-step path-following primal–dual algorithm.

As mentioned previously, Kojima, Shindoh, and Hara [11] have studied a short-step path-following algorithm based on the search direction $(\Delta X(t), \Delta S(t), \Delta y(t))$ for any $t \in [0,1]$ (see (6)). In this section, we give a simplified polynomial convergence proof of their short-step path-following algorithm based on the search direction $(\Delta X(1), \Delta S(1), \Delta y(1))$ or, equivalently, the one determined by (3)–(5). It is a straightforward task to carry out a similar analysis with respect to the search direction $(\Delta X(0), \Delta S(0), \Delta y(0))$.

The short-step path-following algorithm generates iterates in the following (narrow) neighborhood of the central path:

$$\begin{aligned}\mathcal{N}_F(\gamma) &\equiv \{(X,S,y) \in F^0(P)\times F^0(D) : \|X^{1/2}SX^{1/2} - \mu I\|_F \leq \gamma\mu\}\\ &= \left\{(X,S,y) \in F^0(P)\times F^0(D) : \left(\sum_{i=1}^{n}(\lambda_i(XS)-\mu)^2\right)^{1/2} \leq \gamma\mu\right\},\end{aligned}$$

where $\mu \equiv (X \bullet S)/n$ and γ is a constant such that $0 < \gamma < 1$. This neighborhood is a natural extension of the one used by the short-step path-following algorithm studied in [9, 12, 13]. The algorithm, which is a special case of the generic algorithm discussed in section 2, selects the sequence of step-sizes $\{\alpha_k\}$ and centrality parameters $\{\sigma_k\}$ according to the following rule.

SHORT-STEP METHOD. *For all $k \geq 0$, let $\alpha_k = 1$ and $\sigma_k \equiv 1 - \delta/\sqrt{n}$, where $\delta > 0$ is a constant which is specified in Theorem* 4.1 *below.*

The following result analyzes the behavior of one iteration of the short-step path-following method. Its proof will be given at the end of this section.

THEOREM 4.1. *Let $\gamma \in (0,1)$ and $\delta \in [0, n^{1/2})$ be constants satisfying*

$$\frac{\gamma^2+\delta^2}{2(1-\gamma)^2(1-\delta/\sqrt{n})} \leq \gamma, \quad \gamma \leq \frac{1}{2}. \tag{22}$$

Suppose that $(X,S,y) \in \mathcal{N}_F(\gamma)$ and let $(\Delta X, \Delta S, \Delta y)$ denote the solution of (3)–(5) *with $H = \sigma\mu I - X^{1/2}SX^{1/2}$ and $\sigma = 1-\delta/\sqrt{n}$. Then,*

(a) $(\hat{X}, \hat{S}, \hat{y}) \equiv (X+\Delta X, S+\Delta S, y+\Delta y) \in \mathcal{N}_F(\gamma)$;

(b) $\hat{X} \bullet \hat{S} = (1-\delta/\sqrt{n})(X \bullet S)$.

An example of constants γ and δ satisfying the conditions stated in Theorem 4.1 is $\gamma = \delta = 0.3$. As an immediate consequence, we obtain the following result for the short-step path-following method.

COROLLARY 4.2. *Let γ and δ be as in Theorem* 4.1 *and let $(X^0, S^0, y^0) \in \mathcal{N}_F(\gamma)$ be given. Then the short-step path-following method generates a sequence of points $\{(X^k, S^k, y^k)\} \subset \mathcal{N}_F(\gamma)$ such that $X^k \bullet S^k \leq (1-\delta/\sqrt{n})^k (X^0 \bullet S^0)$ for all $k \geq 0$. Moreover, given a tolerance $\epsilon > 0$, the short-step path-following method computes an iterate (X^k, S^k, y^k) satisfying $X^k \bullet S^k \leq \epsilon$ in at most $\sqrt{n}\delta^{-1}\log[\epsilon^{-1}(X^0 \bullet S^0)] = \mathcal{O}(\sqrt{n}\log[\epsilon^{-1}(X^0 \bullet S^0)])$ iterations.*

We now turn our efforts towards proving Theorem 4.1.

LEMMA 4.3. *Suppose that $X \in F^0(P)$, $(S,y) \in F^0(D)$, and let $(\Delta X, \Delta S, \Delta y)$ denote the solution of* (3)–(5) *with $H \equiv \sigma\mu I - X^{1/2}SX^{1/2}$. For any $\alpha \in \Re$, let*

$$(X(\alpha), S(\alpha), y(\alpha)) \equiv (X,S,y) + \alpha(\Delta X, \Delta S, \Delta y), \tag{23}$$

$$\mu(\alpha) \equiv (X(\alpha) \bullet S(\alpha))/n, \tag{24}$$

$$Q(\alpha) \equiv X^{-1/2}[X(\alpha)S(\alpha) - \mu(\alpha)I]X^{1/2}. \tag{25}$$

Then,

$$\begin{aligned} Q(\alpha) + Q(\alpha)^T &= 2(1-\alpha)(X^{1/2}SX^{1/2} - \mu I) \\ &+ \alpha^2\left[X^{-1/2}\Delta X \Delta S X^{1/2} + X^{1/2}\Delta S \Delta X X^{-1/2}\right]. \end{aligned} \tag{26}$$

Proof. Let $\alpha \in \Re$ be given. By Lemma 2.2(c), we have $\mu(\alpha) = (1-\alpha+\sigma\alpha)\mu$. Hence, we obtain

$$\begin{aligned} X(\alpha)S(\alpha) - \mu(\alpha)I &= (X+\alpha\Delta X)(S+\alpha\Delta S) - (1-\alpha+\alpha\sigma)\mu I \\ &= (1-\alpha)(XS-\mu I) + \alpha(XS - \sigma\mu I) \\ &\quad + \alpha(X\Delta S + \Delta X S) + \alpha^2 \Delta X \Delta S. \end{aligned}$$

This relation, together with (3), implies

$$\begin{aligned} Q(\alpha)+Q(\alpha)^T &= 2(1-\alpha)(X^{1/2}SX^{1/2} - \mu I) + 2\alpha(X^{1/2}SX^{1/2} - \sigma\mu I) \\ &\quad + \alpha\left[X^{-1/2}(X\Delta S + \Delta X S)X^{1/2} + X^{1/2}(S\Delta X + \Delta S X)X^{-1/2}\right] \\ &\quad + \alpha^2(X^{-1/2}\Delta X \Delta S X^{1/2} + X^{1/2}\Delta S \Delta X X^{-1/2}) \\ &= 2(1-\alpha)(X^{1/2}SX^{1/2} - \mu I) \\ &\quad + \alpha^2(X^{-1/2}\Delta X \Delta S X^{1/2} + X^{1/2}\Delta S \Delta X X^{-1/2}). \qquad \square \end{aligned}$$

The following lemma bounds the size of the scaled directions $X^{-1/2}\Delta X X^{-1/2}$ and $X^{1/2}\Delta S X^{1/2}$ for points $(X,S,y) \in F^0(P) \times F^0(D)$, which are "well centered."

Alternative bounds on the size of these quantities which are valid for any $(X, S, y) \in F^0(P) \times F^0(D)$ are given in Lemma 5.6, but the proof of the result below is considerably simpler than that of Lemma 5.6. The following inequality involving norms is used in the proof of the lemma below and in other places in our presentation: for any $A_1, A_2 \in \Re^{n\times n}$, we have $\|A_1A_2\|_F \le \|A_1\|\,\|A_2\|_F$ and $\|A_1A_2\|_F \le \|A_1\|_F\,\|A_2\|$ (see exercise 20 of section 5.6 of [6]).

Lemma 4.4. *Let $X \in F^0(P)$ and $(S, y) \in F^0(D)$ be such that $\|X^{1/2}SX^{1/2} - \nu I\| \le \nu\gamma$ for some $\gamma \in [0, 1)$ and $\nu > 0$. Suppose that $(\Delta X, \Delta S, \Delta y) \in \Re^{n\times n} \times \Re^{n\times n} \times \Re^m$ is a solution of* (3)–(5) *for some $H \in \Re^{n\times n}$ and let $\delta_x \equiv \nu\|X^{-1/2}\Delta X X^{-1/2}\|_F$ and $\delta_s \equiv \|X^{1/2}\Delta S X^{1/2}\|_F$. Then,*

$$\delta_x\delta_s \le \frac{1}{2}\left(\delta_x^2 + \delta_s^2\right) \le \frac{\|H\|_F^2}{2(1-\gamma)^2}.$$

Proof. Using (3) and simple algebraic manipulation, we obtain

$$H = X^{1/2}\Delta S X^{1/2} + \nu X^{-1/2}\Delta X X^{-1/2} + \frac{1}{2}X^{-1/2}\Delta X X^{-1/2}(X^{1/2}SX^{1/2} - \nu I)$$
$$+ \frac{1}{2}(X^{1/2}SX^{1/2} - \nu I)X^{-1/2}\Delta X X^{-1/2},$$

from which it follows that

$$\begin{aligned}\|H\|_F &\ge \|X^{1/2}\Delta S X^{1/2} + \nu X^{-1/2}\Delta X X^{-1/2}\|_F \\ &\quad - \|X^{1/2}SX^{1/2} - \nu I\|\,\|X^{-1/2}\Delta X X^{-1/2}\|_F \\ &\ge \left(\|X^{1/2}\Delta S X^{1/2}\|_F^2 + \nu^2\|X^{-1/2}\Delta X X^{-1/2}\|_F^2\right)^{1/2} - (\gamma\nu)(\delta_x/\nu) \\ &= \sqrt{\delta_x^2 + \delta_s^2} - \gamma\delta_x \;\ge\; (1-\gamma)\sqrt{\delta_x^2 + \delta_s^2},\end{aligned}$$

where the second inequality follows from the assumption that $\|X^{1/2}SX^{1/2} - \nu I\| \le \nu\gamma$ and the fact that $(X^{-1/2}\Delta X X^{-1/2}) \bullet (X^{1/2}\Delta S X^{1/2}) = \Delta X \bullet \Delta S = 0$, due to Lemma 2.2(a). The result now follows trivially from the last inequality. □

We are now ready to prove Theorem 4.1.

Proof of Theorem 4.1. Statement (b) is an immediate consequence of Lemma 2.2(c) with $\alpha = 1$ and the fact that $\sigma = (1 - \delta/\sqrt{n})$. Hence,

$$\hat{\mu} \equiv (\hat{X} \bullet \hat{S})/n = (1 - \delta/\sqrt{n})\mu. \tag{27}$$

Using the fact that $(X^{1/2}SX^{1/2} - \mu I) \bullet I = 0$, $(X, S, y) \in \mathcal{N}_F(\gamma)$, and $\sigma = (1 - \delta/\sqrt{n})$, we obtain

$$\begin{aligned}\|\sigma\mu I - X^{1/2}SX^{1/2}\|_F^2 &= \|(\sigma - 1)\mu I\|_F^2 + \|\mu I - X^{1/2}SX^{1/2}\|_F^2 \\ &\le \{(1-\sigma)^2 n + \gamma^2\}\mu^2 \;=\; (\delta^2 + \gamma^2)\mu^2.\end{aligned} \tag{28}$$

Since $\|X^{1/2}SX^{1/2} - \mu I\| \le \gamma\mu$, it follows from Lemma 4.4 with $\nu = \mu$ and $H = \sigma\mu I - X^{1/2}SX^{1/2}$ that

$$\|X^{-1/2}\Delta X X^{-1/2}\|_F \le \frac{\|\sigma\mu I - X^{1/2}SX^{1/2}\|_F}{(1-\gamma)\mu} \tag{29}$$

and

$$\|X^{-1/2}\Delta X X^{-1/2}\|_F\,\|X^{1/2}\Delta S X^{1/2}\|_F \le \frac{\|\sigma\mu I - X^{1/2}SX^{1/2}\|_F^2}{2(1-\gamma)^2\mu}. \tag{30}$$

Let $\hat{Q} \equiv Q(1) = X^{-1/2}(\hat{X}\hat{S} - \hat{\mu}I)X^{1/2}$. Using (26) with $\alpha = 1$, (30), (28), (22), and (27), we obtain

$$\begin{aligned}
(31) \qquad \frac{1}{2}\|\hat{Q} + \hat{Q}^T\|_F &= \frac{1}{2}\|X^{-1/2}\Delta X \Delta S X^{1/2} + X^{1/2}\Delta S \Delta X X^{-1/2}\|_F \\
&\le \|X^{-1/2}\Delta X \Delta S X^{1/2}\|_F \\
(32) \qquad &\le \|X^{-1/2}\Delta X X^{-1/2}\|_F \, \|X^{1/2}\Delta S X^{1/2}\|_F \\
(33) \qquad &\le \frac{\|\sigma\mu I - X^{1/2}SX^{1/2}\|_F^2}{2(1-\gamma)^2\mu} \ \le \ \frac{(\gamma^2+\delta^2)\mu}{2(1-\gamma)^2} \\
(34) \qquad &\le \gamma(1 - \delta/\sqrt{n})\mu \ = \ \gamma\hat{\mu}.
\end{aligned}$$

Using (29), (28), and (22), we obtain

$$\begin{aligned}
\|X^{-1/2}\Delta X X^{-1/2}\|_F \le \frac{\|\sigma\mu I - X^{1/2}SX^{1/2}\|_F}{\mu(1-\gamma)} \ &\le \ \frac{(\delta^2+\gamma^2)^{1/2}}{1-\gamma} \\
&\le \left[2\gamma(1-\delta/\sqrt{n})\right]^{1/2} < 1.
\end{aligned}$$

It is easy to see that the last relation implies that $I + X^{-1/2}\Delta X X^{-1/2} \succ 0$ and, hence, $\hat{X} \equiv X + \Delta X = X^{1/2}(I + X^{-1/2}\Delta X X^{-1/2})X^{1/2} \succ 0$. In particular, $\hat{X}^{1/2}$ exists. Applying Lemma 3.3 with $E = \hat{X}^{1/2}\hat{S}\hat{X}^{1/2} - \hat{\mu}I$ and $W = X^{-1/2}\hat{X}^{1/2}$ and noting that $\hat{Q} = WEW^{-1}$, we conclude that

$$(35) \qquad \|\hat{X}^{1/2}\hat{S}\hat{X}^{1/2} - \hat{\mu}I\|_F \le \frac{1}{2}\,\|\hat{Q} + \hat{Q}^T\|_F \le \ \gamma\hat{\mu},$$

where the last inequality is due to (34). This implies that $\lambda_{\min}(\hat{X}^{1/2}\hat{S}\hat{X}^{1/2}) \ge (1-\gamma)\hat{\mu} > 0$, and hence $\hat{X}^{1/2}\hat{S}\hat{X}^{1/2} \succ 0$. Thus, $\hat{S} \succ 0$. Using (4), (5), and the fact that $(X, S, y) \in F^0(P) \times F^0(D)$, it is now easy to see that $(\hat{X}, \hat{S}, \hat{y}) \in F^0(P) \times F^0(D)$. In view of (35), we conclude that $(\hat{X}, \hat{S}, \hat{y}) \in \mathcal{N}_F(\gamma)$. □

5. Long-step path-following algorithm. In this section, we present a long-step path-following algorithm whose iterates lie within a larger conical neighborhood of the central path. The algorithm extends the long-step primal–dual path-following method of Kojima, Mizuno, and Yoshise [10] for solving linear programming problems. We show that the algorithm finds an approximate strictly feasible point (X^k, S^k, y^k) satisfying $X^k \bullet S^k \le \epsilon$ within $\mathcal{O}(n^{3/2}\log(\epsilon^{-1}(X^0 \bullet S^0)))$ iterations, therefore requiring an extra $\sqrt{n}$ factor compared to the complexity of the algorithm in [10].

To describe the algorithm, we need to introduce the following neighborhood of the central path: for $\gamma \in [0, 1)$ and $\Gamma \ge 0$, let

$$\mathcal{N}(\gamma, \Gamma) \equiv \left\{ (X, S, y) \in F^0(P) \times F^0(D) : \begin{array}{l} (1-\gamma)\mu \le \lambda_i(XS) \le (1+\Gamma)\mu \\ \text{for all } i = 1, \ldots, n \end{array} \right\}$$

and

$$\mathcal{N}(\gamma, \infty) \equiv \left\{ (X, S, y) \in F^0(P) \times F^0(D) : \lambda_{\min}(XS) \ge (1-\gamma)\mu \right\},$$

where $\mu \equiv (X \bullet S)/n$. Clearly, $\mathcal{N}(\gamma, \Gamma) \subset \mathcal{N}(\gamma, \infty)$. We will describe the long-step path-following algorithm in terms of the neighborhood $\mathcal{N}(\gamma, \Gamma)$ with $0 \le \Gamma < \infty$. The following straightforward result shows that the corresponding algorithm based on the neighborhood $\mathcal{N}(\gamma,\infty)$ is a special case of the algorithm described in terms of $\mathcal{N}(\gamma,\Gamma)$ for specific values of Γ.

LEMMA 5.1. *For any $\Gamma \geq (n-1)\gamma$ and $\gamma \in [0,1)$, we have $\mathcal{N}(\gamma,\infty) = \mathcal{N}(\gamma,\Gamma)$.*

Proof. Let $(X,S,y) \in \mathcal{N}(\gamma,\infty)$ be given and $\lambda_1 \leq \cdots \leq \lambda_n$ denote the eigenvalues of XS. We know that $\lambda_1 + \cdots + \lambda_n = X \bullet S = n\mu$. Hence, we have

$$\begin{aligned}\lambda_{\max}(XS) = \lambda_n \ &= \ n\mu - (\lambda_1 + \cdots + \lambda_{n-1}) \ \leq \ n\mu - (n-1)(1-\gamma)\mu \\ &= [1+(n-1)\gamma]\mu \ \leq \ (1+\Gamma)\mu,\end{aligned}$$

and hence, $(X,S,y) \in \mathcal{N}(\gamma,\Gamma)$. □

We next describe the path-following algorithm studied in this section. Since the algorithm is a special case of the generic algorithm of section 2, it is enough to specify the choices of the sequence of step-sizes $\{\alpha_k\}$ and centrality parameters $\{\sigma_k\}$. Fix $\gamma \in (0,1)$, $\Gamma \geq \gamma$, $\bar{\sigma} \in (0,1)$, and, for all $k \geq 0$, let $(\Delta X^k, \Delta S^k, \Delta y^k)$ denote the solution of (3)–(5) with $(X,S) = (X^k,S^k)$ and $H = \sigma_k \mu_k I - (X^k)^{1/2} S^k (X^k)^{1/2}$, where $\mu_k \equiv (X^k \bullet S^k)/n$.

LONG-STEP PATH-FOLLOWING METHOD. *For all $k \geq 0$, let $\sigma_k = \bar{\sigma}$ and*

$$(36) \quad \alpha_k = \max\left\{\alpha \in [0,1] : \begin{array}{l}(X^k,S^k,y^k) + \alpha'(\Delta X^k, \Delta S^k, \Delta y^k) \in \mathcal{N}(\gamma,\Gamma) \\ \textit{for all } \alpha' \in [0,\alpha]\end{array}\right\}.$$

The following result describes the behavior of one iteration of the long-step path-following method. Its proof will be given at the end of the section after we have stated and proved several preliminary lemmas.

THEOREM 5.2. *Suppose that $(X,S,y) \in \mathcal{N}(\gamma,\Gamma)$ for some constants $\gamma \in [0,1)$ and $\Gamma \geq \gamma$, and let $(\Delta X, \Delta S, \Delta y)$ denote the solution of (3)–(5) with $H = \sigma\mu I - X^{1/2}SX^{1/2}$ and $\sigma \in [0,1]$. Let*

$$(37) \qquad \tilde{\alpha} \equiv \frac{\sigma\gamma(1-\gamma)^{1/2}}{n(1+\Gamma)^{1/2}}\left((1-\sigma)^2 + \frac{\gamma\sigma^2}{1-\gamma}\right)^{-1}.$$

Then, for any $\alpha \in [0,\tilde{\alpha}]$, we have

(a) *$(X(\alpha),S(\alpha),y(\alpha)) \equiv (X+\alpha\Delta X, S+\alpha\Delta S, y+\alpha\Delta y) \in \mathcal{N}(\gamma,\Gamma)$;*

(b) *$X(\alpha) \bullet S(\alpha) = (1-\alpha+\alpha\sigma)(X \bullet S)$.*

As an immediate consequence of Theorem 5.2, we obtain the following convergence result for the long-step path-following method.

COROLLARY 5.3. *The sequence of iterates $\{(X^k,S^k,y^k)\} \subset \mathcal{N}(\gamma,\Gamma)$ generated by the long-step path-following algorithm satisfies $X^k \bullet S^k \leq (1-\bar{\tau})^k(X^0 \bullet S^0)$ for all $k \geq 0$, where*

$$\bar{\tau} \equiv \frac{\bar{\sigma}(1-\bar{\sigma})\gamma(1-\gamma)^{1/2}}{n(1+\Gamma)^{1/2}}\left((1-\bar{\sigma})^2 + \frac{\gamma\bar{\sigma}^2}{1-\gamma}\right)^{-1}.$$

Moreover, given a tolerance $\epsilon > 0$, the long-step path-following method computes an iterate satisfying $X^k \bullet S^k \leq \epsilon$ in at most $\bar{\tau}^{-1}\log[\epsilon^{-1}(X^0 \bullet S^0)] = \mathcal{O}(n\Gamma^{1/2}\log[\epsilon^{-1}(X^0 \bullet S^0)])$ iterations.

Proof. It follows from Theorem 5.2, relation (36), and the fact that $\sigma_k = \bar{\sigma}$ that $\alpha_k \geq \bar{\tau}/(1-\bar{\sigma})$ for all $k \geq 0$. In view of Theorem 5.2(b), we conclude that $X^{k+1} \bullet S^{k+1} = [1-(1-\bar{\sigma})\alpha_k](X^k \bullet S^k) \leq (1-\bar{\tau})(X^k \bullet S^k)$ for all $k \geq 0$. Hence, the first part of the corollary follows. The second part of the result follows from the first part and some standard arguments. □

It follows from Corollary 5.3 that if the size of the quantity

$$\max\{\gamma^{-1}, (1-\gamma)^{-1}, \sigma^{-1}, (1-\sigma)^{-1}\}$$

is independent of n then the long-step path-following algorithm finds an ϵ-approximate solution in $\mathcal{O}(n\Gamma^{1/2}\log[\epsilon^{-1}(X^0 \bullet S^0)])$ iterations. In view of Lemma 5.1, we conclude that this number of iterations is equal to $\mathcal{O}(n^{3/2}\log[\epsilon^{-1}(X^0 \bullet S^0)])$ when the algorithm uses the neighborhood $\mathcal{N}(\gamma,\infty) = \mathcal{N}(\gamma,(n-1)\gamma)$.

We now turn our efforts towards proving Theorem 5.2.

LEMMA 5.4. *Suppose that $(X,S,y) \in \mathcal{N}(\gamma,\Gamma)$ for some $\gamma \geq 0$ and $\Gamma \geq 0$ and let $(\Delta X, \Delta S, \Delta y)$ denote the solution of (3)–(5) with $H = \sigma\mu I - X^{1/2}SX^{1/2}$ and $\sigma \in [0,1]$. Let $\mu(\alpha)$ and $Q(\alpha)$ be defined as in (24) and (25) for any $\alpha \in \Re$. Then,*

$$(38) \qquad -\gamma\,\mu(\alpha) \leq \frac{1}{2}\lambda_{\min}\left(Q(\alpha)+Q(\alpha)^T\right) \leq \frac{1}{2}\lambda_{\max}\left(Q(\alpha)+Q(\alpha)^T\right) \leq \Gamma\,\mu(\alpha)$$

for any $\alpha \in [0,\bar{\alpha}]$, where

$$(39) \qquad \bar{\alpha} \equiv \min\left\{1,\ \frac{\sigma\mu\min\{\gamma,\Gamma\}}{\|X^{-1/2}\Delta X\Delta S X^{1/2}\|}\right\}.$$

Proof. Let $\alpha \in [0,\bar{\alpha}]$ be given. By Lemma 2.2(c), we have $\mu(\alpha) = (1-\alpha+\sigma\alpha)\mu$. This relation, (12), (26), and the fact that $\lambda_{\max}(X^{1/2}SX^{1/2}-\mu I) \leq \Gamma\mu$, $0 \leq \alpha \leq \bar{\alpha} \leq 1$ and $\lambda_{\max}(\cdot)$ is a homogeneous convex function on the space of symmetric matrices imply that

$$\begin{aligned}
&\frac{1}{2}\lambda_{\max}\left(Q(\alpha)+Q(\alpha)^T\right)\\
&\leq (1-\alpha)\lambda_{\max}(X^{1/2}SX^{1/2}-\mu I)\\
&\qquad + \frac{1}{2}\alpha^2\lambda_{\max}(X^{-1/2}\Delta X\Delta S X^{1/2} + X^{1/2}\Delta S\Delta X X^{-1/2})\\
&\leq (1-\alpha)\Gamma\mu + \frac{1}{2}\alpha^2\|X^{-1/2}\Delta X\Delta S X^{1/2} + X^{1/2}\Delta S\Delta X X^{-1/2}\|\\
&\leq \Gamma\mu(\alpha) - \alpha\sigma\Gamma\mu + \alpha^2\|X^{-1/2}\Delta X\Delta S X^{1/2}\|\\
&\leq \Gamma\mu(\alpha) - \alpha\left(\sigma\Gamma\mu - \bar{\alpha}\|X^{-1/2}\Delta X\Delta S X^{1/2}\|\right) \ \leq\ \Gamma\mu(\alpha),
\end{aligned}$$

where the last inequality is due to (39). Working with the function $\lambda_{\min}(\cdot)$, which is homogeneous and concave over the space of symmetric matrices, and using (12), (26), (39), and the fact that $\lambda_{\min}(X^{1/2}SX^{1/2}-\mu I) \geq -\gamma\mu$, $0 \leq \alpha \leq \bar{\alpha} \leq 1$ and $\mu(\alpha) = (1-\alpha+\sigma\alpha)\mu$, we obtain

$$\begin{aligned}
&\frac{1}{2}\lambda_{\min}\left(Q(\alpha)+Q(\alpha)^T\right)\\
&\geq (1-\alpha)\lambda_{\min}(X^{1/2}SX^{1/2}-\mu I)\\
&\qquad + \frac{1}{2}\alpha^2\lambda_{\min}(X^{-1/2}\Delta X\Delta S X^{1/2} + X^{1/2}\Delta S\Delta X X^{-1/2})\\
&\geq -(1-\alpha)\gamma\mu - \frac{1}{2}\alpha^2\|X^{-1/2}\Delta X\Delta S X^{1/2} + X^{1/2}\Delta S\Delta X X^{-1/2}\|\\
&\geq -\gamma\mu(\alpha) + \alpha\sigma\gamma\mu - \alpha^2\|X^{-1/2}\Delta X\Delta S X^{1/2}\|\\
&\geq -\gamma\mu(\alpha) + \alpha\left(\sigma\gamma\mu - \bar{\alpha}\|X^{-1/2}\Delta X\Delta S X^{1/2}\|\right) \ \geq\ -\gamma\mu(\alpha).
\end{aligned}$$

We have thus shown that (38) holds. □

LEMMA 5.5. *Suppose that* $(X, S, y) \in \mathcal{N}(\gamma, \Gamma)$ *for some* $\gamma \in [0, 1)$ *and* $\Gamma \geq 0$ *and let* $(\Delta X, \Delta S, \Delta y)$ *denote the solution of* (3)–(5) *with* $H = \sigma\mu I - X^{1/2}SX^{1/2}$ *and* $\sigma \in [0, 1]$. *Let* $(X(\alpha), S(\alpha), y(\alpha))$ *be defined as in* (23) *for any* $\alpha \in \Re$. *Then,* $(X(\alpha), S(\alpha), y(\alpha)) \in \mathcal{N}(\gamma, \Gamma)$ *for any* $\alpha \in [0, \hat{\alpha})$, *where*

$$\hat{\alpha} \equiv \min\left\{1, \frac{1}{\|X^{-1/2}\Delta X X^{-1/2}\|}, \frac{\sigma\mu \min\{\gamma, \Gamma\}}{\|X^{-1/2}\Delta X \Delta S X^{1/2}\|}\right\}. \tag{40}$$

Proof. Fix some $\alpha \in [0, \hat{\alpha})$. We first show that $X(\alpha) \in F^0(P)$. Indeed, using (4) and the fact that X is strictly feasible, we easily see that $A_i \bullet X(\alpha) = b_i$ for every $i = 1, \ldots, m$. By (40) and the fact that $\alpha < \hat{\alpha}$, we have $\alpha\|X^{-1/2}\Delta X X^{-1/2}\| < 1$, which in turn implies that $I + \alpha X^{-1/2}\Delta X X^{-1/2} \succ 0$. Thus, $X(\alpha) \equiv X + \alpha\Delta X = X^{1/2}(I + \alpha X^{-1/2}\Delta X X^{-1/2})X^{1/2} \succ 0$. Hence, $X(\alpha) \in F^0(P)$.

Let $\mu(\alpha)$ and $Q(\alpha)$ be defined as in (24) and (25) and let $W(\alpha) \equiv X^{-1/2}[X(\alpha)]^{1/2}$ and $E(\alpha) \equiv [X(\alpha)]^{1/2}S(\alpha)[X(\alpha)]^{1/2} - \mu(\alpha)I$. Clearly, $W(\alpha)$ is nonsingular and $W(\alpha)E(\alpha)W(\alpha)^{-1} = Q(\alpha)$. In view of Lemma 3.3, we conclude that

$$\frac{1}{2}\lambda_{\min}\left(Q(\alpha) + Q(\alpha)^T\right) \leq \lambda_{\min}(E(\alpha)) \leq \lambda_{\max}(E(\alpha)) \leq \frac{1}{2}\lambda_{\max}\left(Q(\alpha) + Q(\alpha)^T\right).$$

Using this relation, Lemma 5.4, and the fact that $\hat{\alpha} \leq \bar{\alpha}$, we conclude that

$$-\gamma\mu(\alpha) \leq \lambda_{\min}(E(\alpha)) \leq \lambda_{\max}(E(\alpha)) \leq \Gamma\mu(\alpha).$$

To conclude the proof, it remains to show that $(S(\alpha), y(\alpha)) \in F^0(D)$. Indeed, the first inequality of the last relation, the definition of $E(\alpha)$, and the assumption that $\gamma < 1$ imply that

$$\lambda_{\min}\left([X(\alpha)]^{1/2}S(\alpha)[X(\alpha)]^{1/2}\right) \geq (1 - \gamma)\mu(\alpha) > 0.$$

Hence, $[X(\alpha)]^{1/2}S(\alpha)[X(\alpha)]^{1/2} \succ 0$, which in turn implies that $S(\alpha) \succ 0$. Using both (5) and the fact that (S, y) is strictly feasible, we easily see that $\sum_{i=1}^n y_i(\alpha)A_i + S(\alpha) = C$. Hence, $(S(\alpha), y(\alpha)) \in F^0(D)$. We have thus shown that $(X(\alpha), S(\alpha), y(\alpha)) \in \mathcal{N}(\gamma, \Gamma)$. □

We now state the following result due to Kojima, Shindoh, and Hara [11].

LEMMA 5.6. *Suppose that* $X \in F^0(P)$, $(S, y) \in F^0(D)$ *and let* $(\Delta X, \Delta S, \Delta y)$ *denote the solution of* (3)–(5) *with* $H \equiv \sigma\mu I - X^{1/2}SX^{1/2}$. *Then,*

$$\|X^{-1/2}\Delta X S^{1/2}\|_F \leq \sqrt{\mu}\,\|\sigma R^{-T} - R\|_F, \tag{41}$$

$$\|X^{-1/2}\Delta X X^{-1/2}\|_F \leq \|R^{-1}\|\,\|\sigma R^{-T} - R\|_F, \tag{42}$$

$$\|S^{-1/2}\Delta S S^{-1/2}\|_F \leq \|R^{-1}\|\,\|\sigma R^{-T} - R\|_F, \tag{43}$$

$$\|S^{-1/2}\Delta S X^{1/2}\|_F \leq \sqrt{\mu}\,\|R\|\,\|R^{-1}\|\,\|\sigma R^{-T} - R\|_F, \tag{44}$$

where $R \equiv \mu^{-1/2}X^{1/2}S^{1/2}$.

Proof. Using the definition of R and standard norm inequalities, it is easy to see that (41) implies (42) and that (43) implies (44). In view of Lemma 2.1, there exists $W \in \mathcal{S}_\perp$ such that $(\Delta X, \Delta S, \Delta y, W)$ is a solution of the system consisting of (4), (5), and the equation $X(\Delta S + W) + \Delta X S = \sigma\mu I - XS$. In view of Corollary 7.7 of [11], we conclude that

$$\|X^{1/2}(\Delta S + W)S^{-1/2}\|_F \leq \|\sigma\mu X^{-1/2}S^{-1/2} - X^{1/2}S^{1/2}\|_F = \sqrt{\mu}\|\sigma R^{-T} - R\|_F \tag{45}$$

and

$$\|X^{-1/2}\Delta X S^{1/2}\|_F \leq \|\sigma\mu X^{-1/2}S^{-1/2} - X^{1/2}S^{1/2}\|_F = \sqrt{\mu}\|\sigma R^{-T} - R\|_F,$$

which shows (41). It remains to show (43). Indeed, relation (45) and the definition of R imply that

$$\begin{aligned}\|S^{-1/2}(\Delta S + W)S^{-1/2}\|_F &\leq \|S^{-1/2}X^{-1/2}\|\,\|X^{1/2}(\Delta S + W)S^{-1/2}\|_F \\ &\leq \|R^{-1}\|\,\|\sigma R^{-T} - R\|_F.\end{aligned}$$

Let $E \equiv S^{-1/2}(\Delta S + W)S^{-1/2}$. Using the fact that $(E + E^T)/2 = S^{-1/2}\Delta S S^{-1/2}$ and $\|(E^T + E)/2\|_F \leq \|E\|_F$ with the above inequality, we obtain (43). □

LEMMA 5.7. *Let R be a nonsingular matrix such that $\|R\|_F = \sqrt{n}$. Then, for any $\sigma \in \Re$, we have*

$$\|\sigma R^{-T} - R\|^2 \leq n(1 - 2\sigma + \sigma^2\|R^{-1}\|^2).$$

Proof. Using (17), we obtain

$$\text{Tr } (R^T R)^{-1} = \sum_{i=1}^{n} \lambda_i\left((R^T R)^{-1}\right) \leq n\lambda_{\max}\left(R^{-1}R^{-T}\right) = n\|R^{-1}\|^2. \tag{46}$$

This relation together with the assumption that $\|R\|_F^2 = n$ imply

$$\begin{aligned}\|R - \sigma R^{-T}\|_F^2 &= \text{Tr }\left(R^T - \sigma R^{-1}\right)\left(R - \sigma R^{-T}\right) \\ &= \text{Tr }\left(R^T R - 2\sigma I + \sigma^2(R^T R)^{-1}\right) \\ &= \|R\|_F^2 - 2\sigma n + \sigma^2 \text{Tr } (R^T R)^{-1} \\ &\leq n(1 - 2\sigma + \sigma^2\|R^{-1}\|^2). \qquad □\end{aligned}$$

We are now ready to prove Theorem 5.2.

Proof of Theorem 5.2. In view of Lemma 5.5, it is sufficient to show that

$$\tilde{\alpha} \leq \min\left\{1, \frac{1}{\|X^{-1/2}\Delta X X^{-1/2}\|}, \frac{\sigma\mu\gamma}{\|X^{-1/2}\Delta X \Delta S X^{1/2}\|}\right\}, \tag{47}$$

where $\tilde{\alpha}$ is defined in (37). Indeed, using (17), the definition of R, and the fact that $(1-\gamma)\mu \leq \lambda_{\min}(XS) \leq \lambda_{\max}(XS) \leq (1+\Gamma)\mu$, we have

$$\|R\|^2 = \lambda_{\max}(R^T R) = \frac{\lambda_{\max}(S^{1/2}XS^{1/2})}{\mu} = \frac{\lambda_{\max}(XS)}{\mu} \leq (1 + \Gamma) \tag{48}$$

and

$$\|R^{-1}\|^2 = \frac{1}{\lambda_{\min}(R^T R)} = \frac{\mu}{\lambda_{\min}(S^{1/2}XS^{1/2})} = \frac{\mu}{\lambda_{\min}(XS)} \leq \frac{1}{(1-\gamma)}. \tag{49}$$

By (49) and Lemma 5.7(a), we have

$$\|\sigma R^{-T} - R\|_F^2 \leq \left(1 - 2\sigma + \frac{\sigma^2}{1-\gamma}\right)n = \left((1-\sigma)^2 + \frac{\gamma\sigma^2}{1-\gamma}\right)n. \tag{50}$$

Using (41), (44), (48), (49), (50), and (37), we obtain

$$\begin{aligned}\tilde{\alpha}\|X^{-1/2}\Delta X\Delta S X^{1/2}\| &\leq \tilde{\alpha}\|X^{-1/2}\Delta X S^{1/2}\|_F\,\|S^{-1/2}\Delta S X^{1/2}\|_F \\ &\leq \tilde{\alpha}\left[\sqrt{\mu}\,\|\sigma R^{-T}-R\|_F\right]\left[\sqrt{\mu}\,\|R\|\,\|R^{-1}\|\,\|\sigma R^{-T}-R\|_F\right] \\ &\leq \tilde{\alpha}\mu\|R\|\,\|R^{-1}\|\,\|\sigma R^{-T}-R\|_F^2 \\ &\leq \tilde{\alpha}\mu\frac{(1+\Gamma)^{1/2}}{(1-\gamma)^{1/2}}\left((1-\sigma)^2+\frac{\gamma\sigma^2}{1-\gamma}\right)n \;=\; \sigma\gamma\mu.\end{aligned}$$

Moreover, using (42), (49), (50), and (37), we obtain

$$\begin{aligned}\tilde{\alpha}\|X^{-1/2}\Delta X X^{-1/2}\| &\leq \tilde{\alpha}\,\|R^{-1}\|\,\|\sigma R^{-T}-R\|_F \\ &\leq \tilde{\alpha}\,\frac{1}{(1-\gamma)^{1/2}}\left((1-\sigma)^2+\frac{\gamma\sigma^2}{1-\gamma}\right)^{1/2}n^{1/2} \\ &= \frac{\sigma\gamma}{(1+\Gamma)^{1/2}n^{1/2}}\left((1-\sigma)^2+\frac{\gamma\sigma^2}{1-\gamma}\right)^{-1/2} \\ &\leq \frac{\sigma\gamma}{(1+\Gamma)^{1/2}n^{1/2}}\,\frac{(1-\gamma)^{1/2}}{\gamma^{1/2}\sigma} \\ &= \frac{\gamma^{1/2}(1-\gamma)^{1/2}}{(1+\Gamma)^{1/2}n^{1/2}} \;<\; 1.\end{aligned}$$

It is also easy to see that $\tilde{\alpha} \leq 1$. We have thus shown that (47) holds. □

6. Concluding remarks. In this paper, we have provided results which make the task of extending polynomially convergent primal–dual path-following algorithms to SDP a routine exercise. We have illustrated these results for two well-known feasible interior-point path-following algorithms: a short-step and a long-step method. The author believes that similar techniques can be used to extend other polynomially convergent *feasible or infeasible* interior-point path-following methods to the context of SDP.

Acknowledgment. The author has mistakenly claimed in a previous version of this manuscript that the search direction determined by systems (3), (4), and (5) (respectively, (7)) was new. The author thanks Prof. Masakazu Kojima for kindly pointing out to the author that this direction was just a special one from the family of search directions introduced in [11]. The author also thanks Prof. Yin Zhang for pointing out an error in the statement of Lemma 3.2 in a previous version of this manuscript.

REFERENCES

[1] I. Adler and F. Alizadeh, *Primal–Ddual Interior Point Algorithms for Convex Quadratically Constrained and Semidefinite Optimization Problems*, manuscript.

[2] F. Alizadeh, *Interior point methods in semidefinite programming with applications to combinatorial optimization*, SIAM J. Optim. 5 (1995), pp. 13–51.

[3] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, *Primal–Dual Interior-Point Methods for Semidefinite Programming*, manuscript. Presented at the Math Programming Symposium, Ann Arbor, MI, 1994.

[4] G. H. Golub and C. E. Van Loan, *Matrix Computations: Second Edition*, The John Hopkins University Press, Baltimore, MD, 1989.

[5] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz, *An interior-point method for semidefinite programming*, SIAM J. Optim., 6 (1996), pp. 342–361.

[6] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, 1985.

[7] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, New York, 1991.

[8] F. Jarre, *An interior-point method for minimizing the maximum eigenvalue of a linear combination of matrices*, SIAM J. Optim., 31 (1993), pp. 1360–1377.

[9] 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.

[10] 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, Berlin, New York, 1989, pp. 29–47.

[11] M. Kojima, S. Shindoh, and S. Hara, *Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices*, SIAM J. Optim., 7 (1997). pp. 86–125.

[12] R. D. C. Monteiro and I. Adler, *Interior path-following primal–dual algorithms. Part* I*: Linear programming*, Math. Programming, 44 (1989), pp. 27–41.

[13] R. D. C. Monteiro and I. Adler, *Interior path-following primal–dual algorithms. Part* II*: Convex quadratic programming*, Math. Programming, 44 (1989), pp. 43–66.

[14] Y. E. Nesterov and A. S. Nemirovskii, *A general approach to the design of optimal methods for smooth convex functions minimization*, Ekonomika i Matem. Metody, 24 (1988), pp. 509–517 (in Russian).

[15] Y. E. Nesterov and A. S. Nemirovskii, *Self-Concordant Functions and Polynomial Time Methods in Convex Programming*, Central Economic & Mathematical Institute, USSR Acad. Sci. Moscow, USSR, 1989, preprint.

[16] Y. E. Nesterov and A. S. Nemirovskii, *Optimization Over Positive Semidefinite Matrices: Mathematical Background and User's Manual*, Technical report, Central Economic & Mathematical Institute, USSR Acad. Sci. Moscow, USSR, 1990.

[17] Y. E. Nesterov and A. S. Nemirovskii, *Interior Point Methods in Convex Programming: Theory and Applications*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1994.

[18] Y. E. Nesterov and M. Todd, *Primal–Dual Interior-Point Methods for Self-Scaled Cones*, Technical report 1125, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1995.

[19] Y. E. Nesterov and M. Todd, *Self-scaled barriers and interior-point methods for convex programming*, Math. Oper. Res., 22 (1997), pp. 1–42.

[20] L. Vandenberghe and S. Boyd, *A primal–dual potential reduction method for problems involving matrix inequalities*, Math. Programming, 69 (1995), pp. 205–236.

[21] Y. Ye, *A class of projective transformations for linear programming*, SIAM J. Comput., 19 (1990), pp. 457–466.

SIAM J. OPTIM.
Vol. 7, No. 3, pp. 679–696, August 1997

CONVEX ANALYSIS OF SPECTRALLY DEFINED MATRIX FUNCTIONS*

ALBERTO SEEGER†

Abstract. The purpose of this work is to carry out a systematic study of a special class of convex functions defined over the space S_n of symmetric matrices of order $n \times n$. The functions under consideration ($\Phi : S_n \to R \cup \{+\infty\}$) are spectrally defined in the sense that the value $\Phi(A)$ depends only on the spectrum $\{\lambda_1(A), \ldots, \lambda_n(A)\}$ of the matrix $A \in S_n$. Fenchel–Legendre conjugation, first- and second-order subdifferentiability, asymptotic behavior, and other concepts of convex analysis are the main ingredients of our exposition.

Key words. spectrally defined functions, convex analysis, Fenchel–Legendre conjugate, subdifferential, second-order subdifferential, asymptotic behavior, regularization, barrier function

AMS subject classifications. 15A42, 49N15, 52A40

PII. S1052623495288866

1. Introduction.

This work deals with a special class of functions ($\Phi : S_n \to R \cup \{+\infty\}$) defined over the space S_n of $n \times n$ real symmetric matrices.

DEFINITION 1.1. $\Phi : S_n \to R \cup \{+\infty\}$ *is said to be* spectrally defined *if there is a symmetric function* $f : R^n \to R \cup \{+\infty\}$ *such that*

$$\Phi(A) = \Phi_f(A) := f(\lambda(A)) \qquad \textit{for all } A \in S_n,$$

where $\lambda(A) := (\lambda_1(A), \ldots, \lambda_n(A))^T$ *is the vector of eigenvalues of* A *in nondecreasing order.*

Recall that a function f over R^n is said to be symmetric if $f(\Pi x) = f(x)$ for all $n \times n$ permutation matrix Π. It is not difficult to prove that Φ is spectrally defined if and only if Φ is *orthonormal invariant* in the sense that

$$\Phi(U^T A U) = \Phi(A) \qquad \text{for all } U \in O_n,$$

where O_n denotes the set of orthonormal matrices of order $n \times n$. The symmetric function f appearing in Definition 1.1 is necessarily unique. In fact, it is given by

$$f(x) = \Phi(\text{diag} x) \qquad \text{for all } x \in R^n,$$

where $\text{diag}\, x$ stands for the diagonal matrix whose entries on the diagonal are the components of x.

Spectrally defined functions arise in various areas of applied mathematics: optimality criteria in experimental design theory [27], [15], barrier functions in matrix optimization [23], [17], matrix updates in quasi-Newton methods [10], [34], potential energy densities for isotropic elastic materials [8, Section 2.3], etc. Some standard examples are shown below.

Example 1.1. Consider the function $A \in S_n \mapsto \Phi(A) = \log(\text{tr}\, e^A)$, where "tr" stands for the trace operator. Φ is spectrally defined in terms of the symmetric function

$$x \in R^n \mapsto f(x) = \log(e^{x_1} + \cdots + e^{x_n}).$$

* Received by the editors July 10, 1995; accepted for publication (in revised form) January 23, 1996.

http://www.siam.org/journals/siopt/7-3/28886.html

† Department of Mathematics, University of Avignon, 33, rue Louis Pasteur, 84000 Avignon, France (alberto.seeger@univ-avignon.fr).

Example 1.2. The largest eigenvalue function $A \in S_n \mapsto \Phi(A) = \lambda_{\max}(A)$ is spectrally defined. In this case,

$$f(x) = \max\{x_1, \ldots, x_n\} \qquad \text{for all } x \in R^n.$$

Example 1.3. The function

$$A \in S_n \mapsto \Phi(A) = \begin{cases} \operatorname{tr} A^{-1} & \text{if } A \text{ is positive definite,} \\ +\infty & \text{otherwise} \end{cases}$$

arises in the theory of optimal experimental design [27]. Φ is spectrally defined in terms of

$$x \in R^n \mapsto f(x) = \begin{cases} \frac{1}{x_1} + \cdots + \frac{1}{x_n} & \text{if } x_1 > 0, \ldots, x_n > 0, \\ +\infty & \text{otherwise.} \end{cases}$$

From the point of view of convex analysis, most of the interesting properties of Φ_f can be derived directly from those of f. For instance, Lewis [17] recently obtained an expression for the conjugate of Φ_f in terms of the conjugate of f. Lewis's formula is an elegant and powerful result that has a large number of applications. For example, it is used in [17] to express the subdifferential of Φ_f in terms of the subdifferential of f. The purpose of our work is to complement Lewis's paper by deepening the analysis of spectrally defined functions. More precisely, we explore this class of functions in connection with the following concepts: Legendre–Fenchel conjugation, first- and second-order subdifferentiability, regularization, unconstrained minimization, diagonal-constrained minimization, good asymptotic behavior, recession analysis, degree of pointedness, and barrier functions.

Most of the spectrally defined functions arising in the literature are associated to symmetric functions that are proper convex lower semicontinuous. This is the case in Examples 1.1–1.3. For notational convenience, we write

$$f \in E(R^n) \overset{\text{def}}{\Leftrightarrow} \begin{cases} f : R^n \to R \cup \{+\infty\} \text{ is a symmetric proper} \\ \text{convex lower-semicontinuous function.} \end{cases}$$

For a matrix $A \in S_n$, we use the standard notation

$$\begin{array}{lll} A > 0 & \text{if} & A \quad \text{is positive definite,} \\ A \geq 0 & \text{if} & A \quad \text{is positive semidefinite.} \end{array}$$

Most of our results remain valid, with obvious changes, for functions defined on the bigger linear space of Hermitian $n \times n$ complex matrices.

2. Fenchel–Legendre conjugation. Recall that the Fenchel–Legendre conjugate Φ^* of the function $\Phi : S_n \to R \cup \{+\infty\}$ is defined by

$$\Phi^*(B) := \sup_{A \in S_n} \{\langle A, B\rangle - \Phi(A)\} \qquad \text{for all } B \in S_n, \tag{2.1}$$

where $\langle \cdot, \cdot \rangle$ stands for the usual inner product in the space S_n, i.e.,

$$\langle A, B\rangle = \operatorname{tr}(AB) \qquad \text{for all } A, B \in S_n.$$

The conjugate function Φ^* provides very valuable information on the function Φ itself. Computing the conjugate Φ_f^* of a spectrally defined function Φ_f can be quite a cumbersome task. If one uses the definition (2.1), then one has to solve a maximization problem over a space of symmetric matrices. As indicated by Lewis [17], the computation of Φ_f^* can be carried out by evaluating the conjugate

$$y \in R^n \mapsto f^*(y) := \sup_{x \in R^n} \{\langle x, y\rangle - f(x)\}$$

of $f : R^n \to R \cup \{+\infty\}$. The symbol $\langle \cdot, \cdot \rangle$ refers this time to the usual inner product in the space R^n.

Theorem 2.1 (see [17, Theorem 2.6]). *Let $f \in E(R^n)$. Then, Φ_f^* is spectrally defined in terms of the symmetric function f^*. In short,*

$$\Phi_f^* = \Phi_{f^*}.$$

Proof. Our proof is different from that in [17, Theorem 2.6]. Take any $B \in S_n$ and write

$$\begin{aligned} \Phi_f^*(B) &= \sup_{A \in S_n} \{\langle A, B\rangle - \Phi_f(A)\} \\ &= \sup_{A \in S_n} \{\langle U^T A U, \operatorname{diag}\lambda(B)\rangle - \Phi_f(A)\}, \end{aligned}$$

where U is an $n \times n$ orthonormal matrix such that $U^T B U = \operatorname{diag}\lambda(B)$. Since

$$\Phi_f(A) = \Phi_f(U^T A U),$$

one gets

$$\Phi_f^*(B) = \sup_{A \in S_n} \{\langle A, \operatorname{diag}\lambda(B)\rangle - \Phi_f(A)\}$$

or, equivalently,

$$\Phi_f^*(B) = \sup\{\langle Q(\operatorname{diag}x)Q^T,\ \operatorname{diag}\lambda(B)\rangle - f(x) : \qquad Q \in O_n, \qquad x \in R^n\}.$$

By choosing Q as the identity matrix, one gets (in particular)

$$\Phi_f^*(B) \geq \sup_{x \in R^n} \{\langle x, \lambda(B)\rangle - f(x)\} = f^*\ (\lambda(B)).$$

To prove that $\Phi_f^*(B) \leq f^*(\lambda(B))$, it suffices to combine the Young–Fenchel inequality

$$f^*(\lambda(B)) + f(\lambda(A)) \geq \langle \lambda(A), \lambda(B)\rangle,$$

and the well-known trace inequality

$$\langle \lambda(A), \lambda(B)\rangle \geq \langle A, B\rangle \qquad \text{(cf. [33], [17]).} \qquad \Box$$

Remark. A result somehow related to Theorem 2.1 can be found in Barbara and Crouzeix [4, Theorem 5.1]. Theorem 2.1 remains true if one drops the convexity and/or the lower semicontinuity of f.

As a way of illustrating Theorem 2.1, consider the following examples.

Example 2.1. In the context of the theory of optimal experimental design, the function

$$A \in S_n \mapsto \Phi(A) = \begin{cases} \lambda_{\max}(A^{-1}) & \text{if } A > 0, \\ +\infty & \text{otherwise} \end{cases}$$

is known as the E-optimality criterion [27]. This function is spectrally defined in terms of

$$x \in R^n \mapsto f(x) = \begin{cases} \max\left\{\dfrac{1}{x_1}, \dots, \dfrac{1}{x_n}\right\} & \text{if } x_1 > 0, \dots, x_n > 0, \\ +\infty & \text{otherwise.} \end{cases}$$

As a matter of calculus one gets

$$f^*(y) = \begin{cases} -2[-(y_1 + \cdots + y_n)]^{1/2} & \text{if } y_1 \leq 0, \dots, y_n \leq 0, \\ +\infty & \text{otherwise} \end{cases}$$

and, consequently,

$$\Phi^*(B) = \begin{cases} -2[\mathrm{tr}(-B)]^{1/2} & \text{if } -B \geq 0, \\ +\infty & \text{otherwise .} \end{cases}$$

The above expression is obtained in [15, Corollary 6.4] by using a rather cumbersome method.

Example 2.2. The spectral radius of a matrix $A \in S_n$ is the number

$$\Phi(A) = \max\{\lambda_{\max}(A), -\lambda_{\min}(A)\}.$$

$\Phi : S_n \to R$ is spectrally defined in terms of the symmetric function $f = \|\cdot\|_\infty$. Since $f^* = \|\cdot\|_1$, one has

$$\Phi^*(B) = |\lambda_1(B)| + \cdots + |\lambda_n(B)| \qquad \text{for all } B \in S_n.$$

As an immediate consequence of Theorem 2.1 one has the following corollaries.

COROLLARY 2.2 (see [17, Corollary 2.7]). *If $f \in E(R^n)$, then Φ_f is proper convex lower semicontinuous.*

COROLLARY 2.3. *If $f \in E\,(R^n)$, then*

$$\Phi_f^*(B) + \Phi_f(A) \geq \langle \lambda(A), \lambda(B) \rangle \qquad \textit{for all } A, B \in S_n.$$

COROLLARY 2.4. *If Φ is spectrally defined, then*

$$(\Phi \circ \mathrm{diag})^*(y) = \Phi^*(\mathrm{diag} y) \qquad \textit{for all } y \in R^n.$$

3. Subdifferentiability. The effective domain of $\Phi : S_n \to R \cup \{+\infty\}$ is defined as the set

$$\mathrm{dom}\ \Phi := \{A \in S_n : \ \Phi(A) < +\infty\}.$$

The *subdifferential* at $A \in$ dom Φ is by definition

$$\begin{aligned} \partial\Phi(A) &:= \{B \in S_n : \Phi(A') \geq \Phi(A) + \langle A' - A, B \rangle \qquad \text{for all } A' \in S_n\} \\ &= \{B \in S_n : \Phi^*(B) + \Phi(A) - \langle A, B \rangle = 0\}. \end{aligned}$$

When Φ is a convex function, the set $\partial\Phi(A)$ reflects the first-order behavior of Φ around A. Higher-order information on Φ can be obtained from the set

$$\begin{aligned} \partial_\epsilon\Phi(A) &:= \{B \in S_n : \Phi(A') \geq \Phi(A) + \langle A' - A, B \rangle - \epsilon \qquad \text{for all } A' \in S_n\} \\ &= \{B \in S_n : \Phi^*(B) + \Phi(A) - \langle A, B \rangle \leq \epsilon\}, \end{aligned}$$

which is known as the ϵ-subdifferential of Φ at A. For $\epsilon > 0$, the set $\partial_\epsilon \Phi(A)$ is an enlargement of $\partial \Phi(A)$. In fact, one has

$$\partial \Phi(A) = \bigcap_{\epsilon > 0} \partial_\epsilon \Phi(A) = \partial_0 \Phi(A).$$

The following calculus rule serves to check whether or not a given matrix $B \in S_n$ belongs to $\partial_\epsilon \Phi_f\ (A)$.

THEOREM 3.1. *Let $f \in E(R^n)$ and $\epsilon \geq 0$. Then, $B \in \partial_\epsilon \Phi_f(A)$ if and only if*

$$\begin{cases} \alpha := \epsilon + \langle A, B \rangle - \langle \lambda(A), \lambda(B) \rangle \geq 0, \\ \lambda(B) \in \partial_\alpha f(\lambda(A)). \end{cases} \tag{3.1}$$

Proof. According to Theorem 2.1, the condition

$$\Phi_f^*(B) + \Phi_f(A) - \langle A, B \rangle \leq \epsilon$$

is equivalent to

$$f^*(\lambda(B)) + f(\lambda(A)) - \langle A, B \rangle \leq \epsilon.$$

However, this can be written in the form

$$f^*(\lambda(B)) + f(\lambda(A)) - \langle \lambda(A), \lambda(B) \rangle \leq \epsilon + \langle A, B \rangle - \langle \lambda(A), \lambda(B) \rangle. \tag{3.2}$$

To complete the proof, it suffices to observe that the term on the left-hand side of (3.2) is nonnegative. □

COROLLARY 3.2. *Let $f \in E(R^n)$ and $\epsilon \geq 0$. If $B \in \partial_\epsilon \Phi_f(A)$, then*

$$U^T B U \in \partial_\epsilon \Phi_f(U^T A U) \qquad \textit{for all } U \in O_n.$$

Proof. It suffices to observe that

$$\lambda(U^T A U) = \lambda(A), \qquad \lambda(U^T B U) = \lambda(B),$$

and

$$\langle U^T A U, U^T B U \rangle = \langle A, B \rangle. \qquad \square$$

The reverse implication in Corollary 3.2 is obviously true. Thus, if $A = UDU^T$ is a polar decomposition of A, then one has

$$\partial_\epsilon \Phi_f(A) = \{UCU^T : C \in \partial_\epsilon \Phi_f(D)\}.$$

In other words, when it comes to computing the set $\partial_\epsilon \Phi_f(A)$, one can always assume that A is a diagonal matrix.

Another important consequence of Theorem 3.1 is a result due to Lewis [17, Theorem 3.2].

COROLLARY 3.3. *Let $f \in E(R^n)$. Then $B \in \partial \Phi_f(A)$ if and only if*

$$\begin{cases} \langle \lambda(A), \lambda(B) \rangle = \langle A, B \rangle, \\ \quad \lambda(B) \in \partial f(\lambda(A)). \end{cases} \tag{3.3}$$

Proof. Set $\epsilon = 0$ in Theorem 3.1. Also remember that

$$\langle \lambda(A), \lambda(B) \rangle \geq \langle A, B \rangle \text{ whenever } A, B \in S_n. \qquad \square$$

Remark. Corollary 3.3 can be used, in particular, to discuss the differentiability of Φ_f (see [19]). As mentioned in [17, Theorem 2.2], the equality $\langle \lambda(A), \lambda(B) \rangle = \langle A, B \rangle$ occurs if and only if there exists an orthonormal matrix V such that

$$V^T A V = \text{diag}\lambda(A) \text{ and } V^T B V = \text{diag}\lambda(B).$$

The next example shows how Corollary 3.3 works in practice.

Example 3.1. Let Φ be the spectrally defined function introduced in Example 2.1. Let $A \in S_n$ be a positive definite matrix whose smallest eigenvalue has multiplicity $p \in \{1, \ldots, n\}$, i.e.,

$$0 < \lambda_1(A) = \cdots = \lambda_p(A) < \lambda_{p+1}(A) \leq \cdots \leq \lambda_n(A).$$

A standard calculus rule on the subdifferential of a maximum function yields here the estimate

$$\partial f(\lambda(A)) = \text{ convex hull of } \left\{ -\left[\frac{1}{\lambda_1(A)}\right]^2 e_1, \ldots, -\left[\frac{1}{\lambda_1(A)}\right]^2 e_p \right\},$$

where $e_1, \ldots, e_p$ are the p first canonical unit vectors in R^n. Thus, $\lambda(B) \in \partial f(\lambda(A))$ if and only if

$$(3.4) \qquad \left\{ \begin{array}{c} \lambda_1(B) \leq 0, \ldots, \lambda_p(B) \leq 0, \\ \lambda_1(B) + \cdots + \lambda_p(B) = -\left[\dfrac{1}{\lambda_1(A)}\right]^2, \\ \lambda_{p+1}(B) = \cdots = \lambda_n(B) = 0. \end{array} \right.$$

In view of (3.4), the condition $\langle \lambda(A), \lambda(B) \rangle = \langle A, B \rangle$ takes the form

$$(3.5) \qquad \langle A, B \rangle = -1/\lambda_1(A).$$

According to Corollary 3.3, conditions (3.4)–(3.5) are necessary and sufficient for $B \in S_n$ to be in $\partial\Phi(A)$. This is consistent with the estimate

$$\partial\Phi(A) = -\left[\lambda_{\max}(A^{-1})\right]^2 \partial\lambda_{\max}(A^{-1})$$

given in [15, Corollary 6.5].

4. Unconstrained minimization. Another consequence of Theorem 2.1 is that the matrix optimization problem

$$(4.1) \qquad \text{Minimize } \{\Phi_f(A) : A \in S_n\}$$

is equivalent to the simpler problem

$$(4.2) \qquad \text{Minimize } \{f(x) : x \in R^n\}.$$

First of all, one has the following proposition.

PROPOSITION 4.1. *Let $f \in E(R^n)$. Then*

$$\inf_{A \in S_n} \Phi_f(A) = \inf_{x \in R^n} f(x).$$

Proof. Observe that

$$\inf_{A \in S_n} \Phi_f(A) = -\Phi_f^*(0) = -f^*(0) = \inf_{x \in R^n} f(x). \qquad \square$$

Also, the solutions of (4.1) and (4.2) are related to each other. Denote by

$$\epsilon\text{-argmin } \Phi := \{A \in S_n : \Phi(A) - \epsilon \le \inf \Phi\}$$

the set of ϵ-minima of the function $\Phi : S_n \to R \cup \{+\infty\}$. For $\epsilon = 0$, one simply has

$$\text{argmin } \Phi := \{A \in S_n : \Phi(A) = \inf \ \Phi\}.$$

PROPOSITION 4.2. *Let $f \in E(R^n)$ and $\epsilon \ge 0$. Then*

$$A \in \epsilon\text{-}argmin\ \Phi_f \Leftrightarrow \lambda(A) \in \epsilon\text{-}argmin\ f.$$

In particular, Φ_f admits a minimum if and only if f admits a minimum.

Proof. It is immediate from Proposition 4.1. $\square$

The results of this section are evident even without Theorem 2.1. They follow from the elementary observation that, for any $x \in R^n$, there exists a permutation matrix P such that $Px = \lambda(\operatorname{diag} x)$, whence $f(R^n) = \Phi_f(S_n)$ for any symmetric function f.

5. Diagonal-constrained minimization. Given a vector $x \in R^n$, consider the diagonal-constrained minimization problem

$$v(x) := \inf_{A \in S_n} \{\Phi(A) : A_{ii} = x_i \qquad \text{for all } i = 1, \ldots, n\}. \tag{5.1}$$

This type of problem arises in a natural way in the context of matrix optimization. An interesting application can be found in a paper by Fletcher [9], in which $\Phi(A)$ is defined as the sum of the m largest eigenvalues of A (with $1 \le m \le n$).

The following result can be seen as an extension of [9, Lemma A.3]. However, our proof is completely different.

PROPOSITION 5.1. *Let Φ be spectrally defined in terms of a given $f \in E(R^n)$. Then,*

(a) *the optimal-value function v coincides with f;*

(b) *if $x \in$ dom f, then $A_0 = \text{diag}x$ is a solution to* (5.1).

Proof. The adjoint $\text{diag}^* : S_n \to R^n$ of the linear mapping $\text{diag} : R^n \to S_n$ is given by

$$\text{diag}^* A = (A_{11}, \ldots, A_{nn})^T \qquad \text{for all } A \in S_n.$$

Thus, the infimal-value function

$$x \in R^n \mapsto v(x) := \inf_{A \in S_n} \{\Phi(A) : \text{diag}^* A = x\}$$

is just the image of Φ under diag^* [26, p. 38]. By applying [26, Theorem 16.3] and Corollary 2.4, one obtains

$$v^* = \Phi^* \circ \text{diag} = (\Phi \circ \text{diag})^*.$$

By taking conjugates again, one gets

$$v^{**} = (\Phi \circ \mathrm{diag})^{**} = \Phi \circ \mathrm{diag}.$$

The last equality is due to the fact that $f = \Phi \circ \mathrm{diag}$ is a convex lower-semicontinuous function. Hence,

$$\Phi(\mathrm{diag} x) = f(x) = v^{**}(x) \leq v(x) \leq \Phi(\mathrm{diag} x).$$

This completes the proof of the proposition. □

Corollary 5.2. *Let $f \in E(R^n)$ be continuous at $x \in$ dom f. Then, $\partial\Phi_f(\mathrm{diag} x)$ contains a diagonal matrix, and*

$$y \in \partial f(x) \iff \mathrm{diag}\, y \in \partial\Phi_f\ (\mathrm{diag}\ x). \tag{5.2}$$

Proof. Since the matrix $A_0 = \mathrm{diag} x$ is a solution to the convex minimization problem

$$\min_{A \in S_n} \{\Phi_f(A) : \mathrm{diag}^* A = x\},$$

it satisfies the first-order optimality condition

$$\begin{cases} \mathrm{diag}^* A_0 = x, \\ \partial\Phi_f(A_0) \text{ intersects } \{\mathrm{diag} y : \ \ y \in R^n\}. \end{cases}$$

These conditions are derived by applying [26, Theorem 28.3] to the Lagrangian function

$$(A, y) \in S_n \times R^n \mapsto L(A, y) = \Phi_f(A) + \langle y, x - \mathrm{diag}^* A\rangle.$$

This proves that there exists a vector $y \in R^n$ (of Lagrange multipliers) such that $\mathrm{diag} y \in \partial\Phi_f$ $(\mathrm{diag} x)$. Formula (5.2) follows by applying Proposition 5.1 and a general rule on the subdifferential of an optimal-value function like v. Formula (5.2) can also be derived from Corollary 3.3. □

6. Regularization. A standard way to regularize a function $\Phi : S_n \to R \cup \{+\infty\}$ is by taking its infimal-convolution

$$C \in S_n \mapsto [\Phi \Box G](C) := \inf_{A \in S_n} \{\Phi(C - A) + G(A)\}$$

with respect to a "kernel" function $G : S_n \to R \cup \{+\infty\}$. The properties imposed on the kernel G depend essentially on the type of regularity for $\Phi \Box G$ that one wishes to achieve. As a common practice, one supposes that G is at least inf compact, in the sense that

$$G(A) \to +\infty \qquad \text{as } \|A\| \to \infty.$$

Among the most typical examples one has the Moreau–Yosida kernel of index $\alpha > 0$ (cf. [3])

$$G_1(A) := \frac{1}{2\alpha}\|A\|^2,$$

the Baire–Wijsman kernel of index $\alpha > 0$ (cf. [12], [5])

$$G_2(A) := \alpha \|A\|,$$

and the rolling ball kernel of index $\alpha > 0$ (cf. [31])

$$G_3(A) := \begin{cases} -[\alpha^2 - \|A\|^2]^{1/2} & \text{if } \|A\| \leq \alpha, \\ +\infty & \text{otherwise.} \end{cases}$$

It turns out that the above three kernels are spectrally defined if the matrix norm $||\cdot||$ is induced by the inner product. The underlying functions $g_1, g_2, g_3 \in E(R^n)$ are, of course, immediate to identify.

The following theorem is a general result concerning the regularization of spectrally defined functions. It has been obtained independently by A. Lewis in his recent work [21].

THEOREM 6.1. *Let $f \in E(R^n)$. If $g \in E(R^n)$ is inf-compact, then Φ_g is inf-compact, and the infimal-convolution $\Phi_f \square \Phi_g$ is spectrally defined in terms of $f \square g \in E(R^n)$. In short,*

$$\Phi_f \square \Phi_g = \Phi_{f \square g}.$$

Proof. The inf-compacity of g allows us to write (see [16, Section 7])

$$f \square g = (f^* + g^*)^*.$$

This proves that $f \square g \in E(R^n)$. That Φ_g is inf-compact follows from the chain of implications

$$\|A\| \to \infty \Rightarrow \|\lambda(A)\| \to \infty \Rightarrow g(\lambda(A)) \to +\infty.$$

Finally, by applying Theorem 2.1, one gets

$$\Phi_f \square \Phi_g = (\Phi_f^* + \Phi_g^*)^* = (\Phi_{f^*} + \Phi_{g^*})^* = (\Phi_{f^* + g^*})^* = \Phi_{(f^*+g^*)^*} = \Phi_{f \square g}.$$

This completes the proof of the theorem. □

7. Good asymptotic behavior. The concept of good asymptotic behavior plays an important role in the design of algorithms for the minimization of a function whose level sets are not necessarily bounded. This concept has been introduced recently by Auslender and Crouzeix [1].

DEFINITION 7.1. *The function $\Phi : S_n \to R \cup \{+\infty\}$ is said to have* good asymptotic behavior *if*

$$\left.\begin{array}{c} \{(A_k, B_k)\}_{k \in N} \subset \text{Gr } \partial\Phi \\ B_k \to 0 \end{array}\right\} \Rightarrow \Phi(A_k) \to \inf \Phi.$$

More details on this notion can be found in the original work [1]; see also Auslender, Cominetti, and Crouzeix [2]. The notation Gr $\partial\Phi$ in Definition 7.1 refers to the graph of the set-valued mapping $\partial\Phi : S_n \to S_n$.

The purpose of this section is to prove the following result.

THEOREM 7.1. *$\Phi_f : S_n \to R \cup \{+\infty\}$ has good asymptotic behavior if and only if the function $f \in E(R^n)$ does also.*

Proof. Suppose that f has good asymptotic behavior. Consider any sequence $\{(A_k, B_k)\}_{k\in N} \subset \mathrm{Gr}\ \partial\Phi_f$ such that $B_k \to 0$. Since $B_k \in \partial\Phi_f(A_k)$, one has

$$\lambda(B_k) \in \partial f(\lambda(A_k)).$$

This and the condition $\lambda(B_k) \to 0$ yield

$$f(\lambda(A_k)) \to \inf f$$

or, equivalently,

$$\Phi_f(A_k) \to \inf \Phi_f.$$

Conversely, suppose Φ_f has good asymptotic behavior, and let $\{(x_k, y_k)\}_{k\in N} \subset \mathrm{Gr}\partial f$ be any sequence such that $y_k \to 0$. According to Lemma A (cf. Appendix), one has

$$\tilde{y}_k \in \partial f(\tilde{x}_k) \qquad \text{for all } k \in N,$$

where $\tilde{z} \in R^n$ is the vector obtained from $z \in R^n$ after rearranging the components in a nondecreasing order. Now, define

$$A_k = \mathrm{diag}\tilde{x}_k \text{ and } B_k = \mathrm{diag}\tilde{y}_k.$$

In this case,

$$\lambda(B_k) \in \partial f(\lambda(A_k))$$

and

$$\langle \lambda(A_k), \lambda(B_k)\rangle = \langle A_k, B_k\rangle = \langle \tilde{x}_k, \tilde{y}_k\rangle.$$

According to Corollary 3.3, one obtains

$$B_k \in \partial\Phi_f(A_k) \qquad \text{for all } k \in N.$$

Since $y_k \to 0$, one has $B_k \to 0$ and

$$\Phi_f(A_k) \to \inf \Phi_f.$$

Thus, it suffices to apply Proposition 4.1 and observe that

$$\Phi_f(A_k) = f(\lambda(A_k)) = f(\tilde{x}_k) = f(x_k). \qquad \square$$

Most of the interesting spectrally defined functions do have good asymptotic behavior. Theorem 7.1 can be used to check that the functions mentioned in all the previous examples belong to this category.

8. Recession analysis. Recall that if $\Phi : S_n \to R \cup \{+\infty\}$ is proper convex lower semicontinuous, then its *recession function* $\Phi_\infty : S_n \to R \cup \{+\infty\}$ is given by

$$\Phi_\infty(D) := \sup_{t>0} \frac{\Phi(A + tD) - \Phi(A)}{t} \qquad \text{for all } D \in S_n,$$

where A is any matrix in dom Φ. An equivalent expression for Φ_∞ is simply (cf. [26, p. 116])

$$\Phi_\infty(D) = \sup_{B\in \mathrm{dom}\ \Phi^*} \langle B, D\rangle \qquad \text{for all } D \in S_n.$$

The above characterization applies also to a function defined over R^n. If f belongs to $E(R^n)$, then so does f_∞. Moreover, we have the following theorem.

THEOREM 8.1. *Let $f \in E(R^n)$. Then the recession function of the spectrally defined function Φ_f is given by*

$$(\Phi_f)_\infty = \Phi_{f_\infty}.$$

Proof. Take any $D \in S_n$. Then,

$$(\Phi_f)_\infty(D) = \sup_{B \in \text{ dom } \Phi_f^*} \langle B, D\rangle = \sup_{B \in \text{ dom } \Phi_f^*} \langle U^T BU, \text{diag}\lambda(D)\rangle,$$

with $U \in O_n$ such that $U^T DU = \text{diag}\lambda(D)$. But Theorem 2.1 allows us to write

$$\text{dom } \Phi_f^* = \{Q(\text{diag}y)Q^T : Q \in O_n, \qquad y \in \text{ dom } f^*\}.$$

Thus,

$$(\Phi_f)_\infty(D) = \sup_{\substack{Q \in O_n \\ y \in \text{ dom } f^*}} \langle U^T Q(\text{diag}y)Q^T U, \text{diag}\lambda(D)\rangle.$$

The choice $Q = U$ yields in particular

$$(\Phi_f)_\infty(D) \geq \sup_{y \in \text{dom } f^*} \langle y, \lambda(D)\rangle = f_\infty(\lambda(D)).$$

The proof of the reverse inequality is as follows. Let $\{u_1, \ldots, u_n\}$ and $\{q_1, \ldots, q_n\}$ be the columns of U and Q, respectively. As a matter of calculus, one has

$$\langle U^T Q(\text{diag}y)Q^T U, \text{diag}\lambda(D)\rangle = \langle y, M_Q\lambda(D)\rangle \qquad \text{for all } \ Q \in O_n,$$

with

$$[M_Q]_{ij} = \langle q_i, u_j\rangle^2 \qquad \text{for } i, j = 1, \ldots, n.$$

Hence,

$$(\Phi_f)_\infty(D) = \sup_{\substack{Q \in O_n \\ y \in \text{dom } f^*}} \langle y, M_Q\lambda(D)\rangle.$$

Since M_Q is a double stochastic matrix for all $Q \in O_n$ and every such matrix can be written as a convex combination of permutation matrices [7], one obtains

$$(\Phi_f)_\infty(D) \leq \sup\left\{\left\langle y, \left(\sum_{\ell=1}^m \alpha_\ell \Pi_\ell\right)\lambda(D)\right\rangle : y \in \text{ dom } f^*, \alpha_\ell \geq 0, \sum_{\ell=1}^m \alpha_\ell = 1\right\},$$

where $\{\Pi_\ell\}_{\ell=1}^m$ is the collection of all permutation matrices of order $n \times n$. But, for all $y \in \text{dom } f^*$ and $\ell \in \{1, \ldots, m\}$, one has

$$\langle y, \Pi_\ell\lambda(D)\rangle \leq f_\infty(\Pi_\ell\lambda(D)) = f_\infty(\lambda(D)).$$

This yields the reverse inequality $(\Phi_f)_\infty(D) \leq f_\infty(\lambda(D))$. $\square$

Remark. If dom f intersects $\{ke : k \in R\}$, with $e = (1, \dots, 1)^T$, then the proof of Theorem 8.1 becomes much shorter. Indeed, dom Φ_f contains a multiple kI_n of the identity matrix I_n, and

$$\begin{aligned}(\Phi_f)_\infty(D) &= \sup_{t>0} \frac{f(\lambda(kI_n + tD)) - f(\lambda(kI_n))}{t} \\ &= \sup_{t>0} \frac{f(ke + t\lambda(D)) - f(ke)}{t} \\ &= f_\infty(\lambda(D)).\end{aligned}$$

The next example serves to illustrate the use of Theorem 8.1.

Example 8.1. The function $A \in S_n \mapsto \Phi(A) = \text{tr } e^A$ is spectrally defined in terms of $x \in R^n \mapsto f(x) = e^{x_1} + \cdots + e^{x_n}$. Since

$$f_\infty(d) = \begin{cases} 0 & \text{if } d \in R^n_-, \\ +\infty & \text{otherwise }, \end{cases}$$

one has

$$\Phi_\infty(D) = \begin{cases} 0 & \text{if } -D \geq 0, \\ +\infty & \text{otherwise }. \end{cases}$$

9. Barrier functions. Consider the problem of minimizing a function $\nu : S_n \to R \cup \{+\infty\}$ over some closed set $P \subset S_n$. Since ν is allowed to have the value $+\infty$, the minimization problem

$$\text{Minimize } \{\nu(A) : A \in P\} \tag{9.1}$$

includes implicitly the constraint $A \in$ dom ν. Suppose this constraint is easy to handle, so that the main computational difficulty lies in the treatment of the constraint $A \in P$.

The barrier method for problem (9.1) consists of solving the "unconstrained" programs

$$\text{Minimize } \{\nu(A) + c_k\Phi(A) : \qquad A \in S_n\},$$

where $\{c_k\}_{k \in N}$ is a sequence of positive numbers decreasing to zero, and $\Phi : S_n \to R \cup \{+\infty\}$ is a barrier function for the set P. The precise meaning of this concept is as follows.

DEFINITION 9.1. *Let $P \subset S_n$ be a closed set whose interior int P is nonempty. Let the boundary of P be denoted by bd P. $\Phi : S_n \to R \cup \{+\infty\}$ is said to be a* barrier function *for P if*

(i) int $P \subset$ dom Φ,

(ii) *for all* $A_0 \in$ bd P, $\displaystyle\lim_{\substack{A \to A_0 \\ A \in \text{ int } P}} \Phi(A) = +\infty.$

A question of practical interest asks how we should construct barrier functions for different types of sets in S_n. In connection with this question, one has the following result.

THEOREM 9.1. *Suppose $K \subset R^n$ is a closed set whose interior intersects the cone $\{x \in R^n : x_1 \leq \cdots \leq x_n\}$. Let the set*

$$P := \{A \in S_n : \lambda(A) \in K\}$$

be such that

$$bd\ P = \{A \in S_n :\ \lambda(A) \in\ bd\ K\}. \tag{9.2}$$

Under these assumptions, if $f \in E(R^n)$ is a barrier function for K, then Φ_f is a barrier function for P.

Proof. Hypothesis (9.2) is equivalent to the condition

$$\text{int } P = \{A \in S_n : \lambda(A) \in \text{ int } K\}.$$

Since int $K \subset$ dom f, it follows that int $P \subset$ dom Φ_f. To check condition (ii) in Definition 9.1, take any $A_0 \in$ bd P. If $\{A_k\}_{k\in N} \subset$ int P converges to A_0, then $\{\lambda(A_k)\}_{k\in N} \subset$ int K converges to $\lambda(A_0) \in$ bd K. Since f is a barrier function for K, it follows that

$$\Phi_f(A_k) = f(\lambda(A_k)) \to +\infty.$$

This shows that Φ_f is a barrier function for P. □

Remark. Theorem 9.1 is reminiscent of a somehow related result by Barbara and Crouzeix [4, Theorem 10.1]. However, we work with a different concept of barrier function.

Example 9.1. A typical barrier function which fits into the framework of Theorem 9.1 is

$$A \in S_n \mapsto \Phi(A) = \begin{cases} -\log\ \det A & \text{if } A > 0, \\ +\infty & \text{otherwise .} \end{cases}$$

This corresponds to the case $K = R^n_+,\ P = \{A \in S_n : A \geq 0\}$, and

$$x \in R^n \mapsto f(x) = \begin{cases} -\sum_{i=1}^{n} \log x_i & \text{if } x_1 > 0, \ldots, x_n > 0, \\ +\infty & \text{otherwise.} \end{cases}$$

10. Degree of pointedness. It is known that if $\Phi : S_n \to R\cup\{+\infty\}$ is a proper convex lower-semicontinuous function, then

$$\text{epi } \Phi_\infty := \{(A,\alpha) \in S_n \times R :\ \Phi_\infty(A) \leq \alpha\}$$

is a closed convex cone in the space $S_n \times R$. The set

$$\ell(\text{epi } \Phi_\infty) = \text{ epi } \Phi_\infty \cap - \text{ epi } \Phi_\infty$$

is the largest subspace of $S_n \times R$ which is contained in epi Φ_∞. Following the author's previous work [32], we refer to the number

$$p[\Phi] := \text{ dim } S_n - \text{ dim } \ell(\text{ epi } \Phi_\infty)$$

as the *degree of pointedness* of Φ. If $p[\Phi] = \dim S_n$, then Φ is said to be *pointed.* According to this definition, Φ is pointed if and only if epi Φ_∞ is a pointed cone. This particular case has been considered by Benoist and Hiriart-Urruty [6, Definition 2.3]. For a function f defined over the Euclidean space R^n, one has, of course,

$$p[f] = \text{ dim } R^n - \text{ dim } \ell(\text{epi} f_\infty).$$

A detailed discussion on the concept of pointedness can be found in [32]. The theorem recorded below deals with the degree of pointedness of a spectrally defined function. The dimension of a nonempty convex set is defined as the dimension of its affine hull (cf. [26, p. 12]).

THEOREM 10.1. *Let $f \in E(R^n)$ and denote by*

$$\textit{lin } f := \{d \in R^n : f_\infty(d) = -f_\infty(-d)\}$$

its lineality space [26, p. 70]. *Then, the degree of pointedness of Φ_f admits the following two characterizations:*

$$\begin{aligned} p[\Phi_f] &= \dim S_n - \dim \{D \in S_n : \lambda(D) \in \text{lin } f\} \\ &= \dim \{B \in S_n : \lambda(B) \in \text{dom } f^*\}. \end{aligned}$$

In particular, Φ_f is pointed if and only if f is pointed.

Proof. As mentioned in [32], the space $\ell(\text{epi}(\Phi_f)_\infty)$ has the same dimension as

$$\text{lin } \Phi_f = \{D \in S_n : (\Phi_f)_\infty(D) = -(\Phi_f)_\infty(-D)\}.$$

But, according to Theorem 8.1, one can write

$$\begin{aligned} \text{lin } \Phi_f &= \{D \in S_n : \Phi_{f_\infty}(D) = -\Phi_{f_\infty}(-D)\} \\ &= \{D \in S_n : f_\infty(\lambda(D)) = -f_\infty(\lambda(-D))\}. \end{aligned}$$

By taking into account the symmetry of f_∞, one gets, finally,

$$\text{lin } \Phi_f = \{D \in S_n : \lambda(D) \in \text{lin } f\}.$$

This proves the first characterization that has been given for the number $p[\Phi_f]$. The second formula follows from Theorem 2.1 and the fact that (see [32, Theorem 1])

$$p[\Phi_f] = \dim(\text{dom. } \Phi_f^*).$$

Finally, the spectrally defined function Φ_f is pointed if and only if lin Φ_f is a zero-dimensional space in S_n, i.e.,

$$\{D \in S_n : \lambda(D) \in \text{lin } f\} = \{0\}.$$

However, the above equality amounts to saying that lin $f = \{0\} \subset R^n$; i.e., f is pointed. □

Example 10.1. The variance of the matrix $A \in S_n$ is the number

$$\Phi(A) = \frac{1}{n}\|A\|^2 - \left(\frac{\text{tr } A}{n}\right)^2,$$

where $\|\cdot\|$ is the norm associated to the inner product $\langle\cdot,\cdot\rangle$. Φ is spectrally defined in terms of

$$x \in R^n \mapsto f(x) = \frac{1}{n}\|x\|^2 - \left(\frac{\langle e, x\rangle}{n}\right)^2,$$

where $e = (1, \ldots, 1)^T$. Since

$$f_\infty(d) = \begin{cases} 0 & \text{if } d_1 = \cdots = d_n, \\ +\infty & \text{otherwise ,} \end{cases}$$

one has $\mathrm{lin}\ f = \{ke : k \in R\}$. Thus,

$$\{D \in S_n : \lambda(D) \in \mathrm{lin}\ f\} = \{kI_n : \ \ k \in R\}$$

is a one-dimensional subspace of S_n. This means that

$$p[\Phi] = \ \dim\ S_n - 1 = (n^2 + n - 2)/2.$$

This number is obtained also as the dimension of

$$\begin{aligned}\mathrm{dom}\ \Phi^* &= \{B \in S_n :\ \lambda(B) \in \ \mathrm{dom}\ f^*\}\\ &= \{B \in S_n :\ \mathrm{tr}\ B = 0\}.\end{aligned}$$

11. Second-order subdifferentiability. Second-order information on the behavior of the proper convex lower-semicontinuous function $\Phi : S_n \to R \cup \{+\infty\}$ is captured by the second-order subdifferential mapping $\partial^2\Phi : S_n \times S_n \to S_n$. Following our previous work [30], we denote by

$$\partial^2_\epsilon \Phi(A, B) := \frac{\partial_\epsilon \Phi(A) - B}{\sqrt{2\epsilon}}$$

the so-called ϵ-second-order subdifferential of Φ at $(A, B) \in \mathrm{Gr}\ \partial\Phi$. In this section we are interested in the second-order subdifferential

$$\partial^2\Phi(A, B) := \lim_{\epsilon \to 0^+} \partial^2_\epsilon \Phi(A, B), \tag{11.1}$$

where the limit is understood in the sense of Kuratowski–Painlevé. An equivalent expression for $\partial^2\Phi(A, B)$ in terms of Rockafellar's second-order epiderivative concept can be found in [30, Theorem 3.1] or [24, Theorem 1.1]. For the sake of convenience, we split (11.1) into the upper- and lower-limits

$$\begin{cases} \overline{\partial}^2\Phi(A, B) = \limsup\limits_{\epsilon \to 0^+}\ \partial^2_\epsilon \Phi(A, B),\\ \underline{\partial}^2\Phi(A, B) = \liminf\limits_{\epsilon \to 0^+}\ \partial^2_\epsilon \Phi(A, B). \end{cases} \tag{11.2}$$

If Φ is spectrally defined in terms of $f \in E(R^n)$, then it is possible to obtain estimates for $\overline{\partial}^2\Phi$ and $\underline{\partial}^2\Phi$ in terms of $\overline{\partial}^2 f$ and $\underline{\partial}^2 f$, respectively. The next theorem is a result in that direction.

Recall that each eigenvalue function $A \in S_n \mapsto \lambda_i(A)$ is directionally differentiable. Formulas for computing the directional derivative

$$H \in S_n \mapsto \lambda'_i(A; H) := \lim_{t \to 0^+} \frac{\lambda_i(A + tH) - \lambda_i(A)}{t}$$

can be found in works by Overton and Womersley [25] and Hiriart-Urruty and Ye [14, Theorem 3.12]. So, in principle, the computation of the vector

$$\lambda'(A; H) := (\lambda'_1(A; H), \ldots, \lambda'_n(A; H))^T$$

does not constitute a major difficulty.

THEOREM 11.1. *Let $f \in E(R^n)$ and $B \in \partial\Phi_f(A)$. Then*

$$\underline{\partial}^2\Phi_f(A, B) \subset \{C \in S_n : \lambda'(B; C) \in \underline{\partial}^2 f(\lambda(A), \lambda(B))\}.$$

Proof. Let $C \in \underline{\partial}^2\Phi_f(A,B)$, and take any sequence $\{\epsilon_k\}_{k\in N} \to 0^+$. Thus, there is a sequence $\{C_k\}_{k\in N} \subset S_n$ which converges to C and is such that

$$C_k \in \partial^2_{\epsilon_k}\Phi_f(A,B) \qquad \text{for all } k \in N$$

or, equivalently,

$$B + \sqrt{2\epsilon_k}C_k \in \partial_{\epsilon_k}\Phi(A).$$

According to Theorem 3.1, the above inclusion can be expressed in the form

$$\begin{cases} \alpha_k := \epsilon_k + \langle A, B + \sqrt{2\epsilon_k}C_k\rangle - \langle \lambda(A), \lambda(B + \sqrt{2\epsilon_k}C_k)\rangle \geq 0, \\ \lambda(B + \sqrt{2\epsilon_k}C_k) \in \partial_{\alpha_k} f(\lambda(A)). \end{cases}$$

Since $\alpha_k \leq \epsilon_k$, one gets, in particular,

$$\lambda(B + \sqrt{2\epsilon_k}C_k) \in \partial_{\epsilon_k} f(\lambda(A))$$

or, equivalently,

$$z_k := \frac{\lambda(B + \sqrt{2\epsilon_k}C_k) - \lambda(B)}{\sqrt{2\epsilon_k}} \in \partial^2_{\epsilon_k} f(\lambda(A), \lambda(B)).$$

Now, it suffices to observe that $\{z_k\}_{k\in N}$ converges to $\lambda'(B;C)$. □

Similarly, one has the following theorem.

THEOREM 11.2. *Let $f \in E(R^n)$ and $B \in \partial\Phi_f(A)$. Then*

$$\overline{\partial}^2\Phi_f(A,B) \subset \{C \in S_n : \lambda'(B;C) \in \overline{\partial}^2 f(\lambda(A), \lambda(B))\}.$$

Proof. The proof is analogous to the proof of Theorem 11.1. □

In most cases in practice, the lower limit $\underline{\partial}^2 f(\lambda(A), \lambda(B))$ coincides with the upper limit $\overline{\partial}^2 f(\lambda(A), \lambda(B))$. The common limit $\partial^2 f(\lambda(A), \lambda(B))$ can be computed by using calculus rules found in [13], [28], [29]. For instance, if f is twice differentiable at $\lambda(A)$, then $\lambda(B)$ is necessarily equal to the gradient $\bigtriangledown f(\lambda(A))$, and

$$\partial^2 f(\lambda(A), \lambda(B)) = \{z \in R^n : \langle z, h\rangle \leq [\langle h, \bigtriangledown^2 f(\lambda(A))h\rangle]^{1/2} \qquad \text{for all } h \in R^n\}$$

is an ellipsoid associated to the Hessian matrix $\bigtriangledown^2 f(\lambda(A))$.

It must be mentioned, however, that the spectral function Φ_f does not inherit the smoothness of f. This is due to the fact that the matrix A may have repeated eigenvalues. Whether or not it is possible to write the converse inclusions in Theorems 11.1 and 11.2 is a matter which requires further investigation.

Appendix. We record below a general property concerning the ϵ-subdifferential of a symmetric function.

LEMMA A. *Let $f \in E(R^n)$ and $\epsilon \geq 0$. Then, for all $(x,y) \in R^n \times R^n$, one has the implication*

$$y \in \partial_\epsilon f(x) \Rightarrow \tilde{y} \in \partial_\epsilon f(\tilde{x}),$$

where $\tilde{z} \in R^n$ has the same components as $z \in R^n$ but in a nondecreasing order.

Proof. Suppose $y \in \partial_\epsilon f(x)$, i.e.,

$$f^*(y) + f(x) \leq \langle y, x\rangle + \epsilon.$$

Let P and Q be $n \times n$ permutation matrices such that

$$x = P\tilde{x} \quad \text{and} \quad y = Q\tilde{y}.$$

Hence,

$$f^*(Q\tilde{y}) + f(P\tilde{x}) \leq \langle \tilde{y}, Q^T P\tilde{x}\rangle + \epsilon.$$

Since f and f^* are symmetric, one has

$$f^*(\tilde{y}) + f(\tilde{x}) \leq \langle \tilde{y}, Q^T P\tilde{x}\rangle + \epsilon.$$

Since $Q^T P$ is a permutation matrix, one can apply [11, Theorem 368] (see also [17, Lemma 2.1]) to obtain

$$\langle \tilde{y}, Q^T P\tilde{x}\rangle \leq \langle \tilde{y}, \tilde{x}\rangle.$$

In this way one gets, finally,

$$f^*(\tilde{y}) + f(\tilde{x}) \leq \langle \tilde{y}, \tilde{x}\rangle + \epsilon. \qquad \square$$

Acknowledgments. After completing this work, I received some additional papers by A. Lewis [18], [19], [20], [21] and J.E. Martinez-Legaz [22] dealing with the analysis of spectral functions and related questions. I thank both of them for sending me their unpublished manuscripts. Useful remarks by A. Lewis and an anonymous referee are also appreciated.

REFERENCES

[1] A. Auslender and J. P. Crouzeix, *Well behaved asymptotical convex functions*, in Analyse non Linéaire, Gauthier–Villars, Paris, 1989, pp. 101–122.

[2] A. Auslender, R. Cominetti, and J. P. Crouzeix, *Convex functions with unbounded level sets and applications to duality theory*, SIAM J. Optim., 3 (1993), pp. 669–687.

[3] H. Attouch, *Variational Convergence for Functions and Operators*, Pitman, Boston, MA, 1984.

[4] A. Barbara and J. P. Crouzeix, *Concave gauge functions and applications*, ZOR–Math. Methods Oper. Res., 40 (1994), pp. 43–74.

[5] G. Beer, *Lipschitz regularization and the convergence of convex functions*, Numer. Funct. Anal. Optim., 15 (1994), pp. 31–96.

[6] J. Benoist and J. B. Hiriart-Urruty, *Quel est le sous–différentiel de l'enveloppe convexe fermée d'une fonction?*, C.R. Acad. Sci. Paris, t. 316, Série I, (1993), pp. 233–237.

[7] G. Birkhoff, *Tres observaciones sobre el algebra lineal*, Revista Fac. de Ciencias Exactas, Puras y Aplicadas, Universita Nacional de Tucuman, Serie A, 5 (1946), pp. 147–151.

[8] A. Curnier, Q. C. He, and P. Zysset, *Conewise linear elastic materials*, J. Elasticity, 37 (1995), pp. 1–38.

[9] R. Fletcher, *Semidefinite matrix constraints in optimization*, SIAM J. Control Optim., 23 (1985), pp. 493–513.

[10] R. Fletcher, *A new variational result for quasi-Newton formulae*, SIAM J. Optim., 1 (1991), pp. 18–21.

[11] G. H. Hardy, J. E. Littlewood, and G. Polya, *Inequalities*, Cambridge University Press, Cambridge, United Kingdom, 1952.

[12] J. B. Hiriart-Urruty, *Lipschitz r-continuity of the approximate subdifferential of a convex function*, Math. Scand., 47 (1980), pp. 123–134.

[13] J. B. Hiriart-Urruty and A. Seeger, *Calculus rules on a new set-valued second-order derivative for convex functions*, Nonlinear Analysis, Th. Meth. Appl., 13 (1989), pp. 721–738.

[14] J. B. Hiriart-Urruty and D. Ye, *Sensitivity analysis of all eigenvalues of a symmetric matrix*, Numer. Math., 70 (1995), pp. 45–72.

[15] T. Hoang and A. Seeger, *On conjugate functions, subgradients, and directional derivatives of a class of optimality criteria in experimental design*, Statistics, 22 (1991), pp. 349–368.

[16] P. J. Laurent, *Approximation et optimisation*, Hermann, Paris, France, 1972.

[17] A. S. Lewis, *Convex Analysis on the Hermitian Matrices*, Combinatorics and Optimization Research report 93–33, University of Waterloo, Ontario, Canada, December 1993.

[18] A. S. Lewis, *Group invariance and convex matrix analysis*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 927–949.

[19] A. S. Lewis, *Derivatives of spectral functions*, Math. Oper. Res., to appear.

[20] A. S. Lewis, *The convex analysis of unitarily invariant matrix norms*, J. Convex Anal., 2 (1995), pp. 173–183.

[21] A. S. Lewis, *Von Neumann's lemma and a Chevalley-type theorem for convex functions on Cartan subspaces*, Combinatorics and Optimization Research report 95-19, University of Waterloo, Ontario, Canada, July 1995.

[22] J. E. Martinez-Legaz, *On Convex and Quasi-Convex Spectral Functions*, Technical report, Universitat Autónoma de Barcelona, Barcelona, Spain, 1995.

[23] Y. E. Nesterov and A. S. Nemirovskii, *Optimization Over Positive Semidefinite Matrices: Mathematical Background and User's Manual*, USSR Acad. Sci. Center Econ. & Math. Inst., Moscow, 1990.

[24] M. Moussaoui and A. Seeger, *Analyse du second ordre de fonctionnelles intégrales: le cas convex non différentiable*, C.R. Acad. Sci. Paris, t. 318, Série I, (1994), pp. 613–618.

[25] M. L. Overton and R. S. Womersley, *Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices*, Math. Programming, 62 (1993), pp. 321–357.

[26] R. T. Rockafellar, *Convex analysis*, Princeton University Press, Princeton, NJ, 1970.

[27] A. Pazman, *Foundations of optimum experimental design*, in Mathematics and its Applications, East European Series, D. Reidel, Boston, MA, 1986.

[28] A. Seeger, *Analyse du second ordre de problémes non différentiables*, Ph.D. thesis, Univ. Paul Sabatier, Toulouse, France, 1986.

[29] A. Seeger, *Complément de Schur et sous–différentiabilité du second ordre d'une fonction convexe*, Aquationes Mathematicae, 42 (1991), pp. 47–71.

[30] A. Seeger, *Limiting behavior of the approximate second-order subdifferential of a convex function*, J. Optim. Theory Appl., 74 (1992), pp. 527–544.

[31] A. Seeger, *Smoothing a Nondifferentiable Convex Function: The Technique of the Rolling Ball*, Technical report 165, Department of Mathematical Sciences, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, October 1994.

[32] A. Seeger, *Degree of pointedness of a convex function*, Bull. Austral. Math. Soc., 53 (1996), pp. 159–167.

[33] C. M. Theobald, *An inequality for the trace of the product of two symmetric matrices*, Math. Proc. Cambridge Philos. Soc., 77 (1975), p. 265.

[34] H. Wolkowicz, *Measures for symmetric rank–one updates*, Math. Oper. Res., 19 (1994), pp. 815–830.

SIAM J. OPTIM.
Vol. 7, No. 3, pp. 697–716, August 1997

ANALYSIS OF A CUTTING PLANE METHOD THAT USES WEIGHTED ANALYTIC CENTER AND MULTIPLE CUTS*

ZHI-QUAN LUO[†]

Abstract. We consider the analytic center cutting plane (or column generation) algorithm for solving general convex problems defined by a separation oracle. The oracle is called at an approximate analytic center of a polytope which contains the solution set and is given by the intersection of the linear inequalities previously generated from the oracle. If the approximate center is not in the solution set, separating hyperplanes will be placed through the approximate center, and a new approximate analytic center will be found for the shrunken polytope. In this paper, we consider using approximate weighted analytic centers in the cutting plane method and show that the method, with multiple cuts added in each step, has a complexity of $O(\eta m^2/\epsilon^2)$, where η is the maximum number of cuts that can be added in each step and m is the dimension of the problem.

Key words. convex feasibility problem, analytic center, potential reduction, column generation, cutting planes

AMS subject classifications. 90C25, 90C26, 90C60

PII. S105262349427652X

1. Introduction.

Consider the problem of finding a point in a convex set Γ, where $\Gamma \subset \mathbb{R}^m$ is contained in the m-dimensional cube $\Omega^0 = \{y \in \mathbb{R}^m : 0 \leq y \leq e\} = [0,1]^m$. Suppose Γ has a nonempty interior and contains a full dimensional closed ball with an ϵ ($< \frac{1}{2}$) radius. The set Γ is defined implicitly by a separating oracle, which for every $\bar{y} \in \mathbb{R}^m$ either answers $\bar{y} \in \Gamma$ or generates a separating hyperplane $\{y \in \mathbb{R}^m : a^T y \leq a^T \bar{y}\} \supset \Gamma$. Without loss of generality, we assume that a is normalized so that $\|a\| = 1$.

A popular method for solving the above convex feasibility problem is the analytic center cutting plane or column generation algorithm, which, in practice, has demonstrated superior performance (see [4, 5, 6] and references therein). Roughly speaking, the algorithm starts with Ω^0 as an initial polytope that contains Γ. Then, in each step the oracle is called at an approximate analytic center of the containing polytope given by the intersection of the linear inequalities previously generated from the oracle. If the center is not in Γ yet, separating hyperplanes will be placed through the center to further trim down the containing polytope. A new approximate center is computed for the shrunken polytope and the process is continued.

Several complexity results have been established for this cutting plane algorithm when one cut is used in each step; see Atkinson and Vaidya [2], Goffin, Luo, and Ye [6], Nesterov [7], and Altman and Kiwiel [1]. In practice, adding multiple cuts is a key to the algorithm's performance (see [4, 5]); unfortunately, this also presents technical difficulties in analyzing all cutting plane methods (including the ellipsoid method). Among the issues to be resolved are (i) how "close" does each approximate analytic center have to be from the exact center, (ii) can we still in $O(1)$ Newton iterations obtain a new approximate center after multiple cuts are added in each step, and (iii) how many steps are needed for the algorithm to find an ϵ-feasible solution? Recently,

[*] Received by the editors November 24, 1994; accepted for publication (in revised form) March 15, 1996. This research was supported by Natural Sciences and Engineering Research Council of Canada grant OPG0090391.

http://www.siam.org/journals/siopt/7-3/27652.html

[†] Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario L8S 4L7, Canada (luozq@mcmail.cis.mcmaster.ca).

Ye [10] analyzed the complexity of the algorithm when multiple cuts are added in each step (at the exact analytic center), and he obtained a bound of $O(\eta^2 m^2/\epsilon^2)$ on the total number of Newton iterations needed to find an ϵ-feasible solution. Here η is the maximum number of cuts that can be added in each step and m is the dimension of the problem. In this paper, we consider using approximate-weighted analytic centers in the cutting plane method and show that the method, with multiple cuts added in each step, has an improved complexity of $O(\eta m^2/\epsilon^2)$. Moreover, η need not be smaller than m as was stipulated in [10]. Also, our analysis shows that a new approximate analytic center can still be obtained in $O(1)$ Newton iterations, just like the cutting plane method which uses a single cut in each step [6].

2. Preliminaries. Let Ω be a (bounded) polytope in $\mathbb{R}^m$ defined by n ($> m$) linear inequalities, i.e.,

$$\Omega = \{y \in \mathbb{R}^m : c - A^T y \geq 0\}.$$

Recall that for y in the interior of Ω and for some weight vector $w > 0$, the potential function is defined as

$$\phi(y) := \sum_{j=1}^{n} w_j \log s_j,$$

where

$$s = c - A^T y.$$

The gradient and the Hessian of $\phi(y)$ are given by

$$\nabla\phi(y) = -\sum_{j=1}^{n} \frac{w_j}{s_j} = -AS^{-1}w$$

and

$$H = -\nabla^2\phi(y) = \sum_{j=1}^{n} \frac{w_j a_j a_j^T}{s_j^2} = AWS^{-2}A^T.$$

Here and throughout this paper, we shall use the convention that the capitalized letters W, S, etc. will denote the diagonal matrices whose (j,j)th entry is equal to w_j, s_j, respectively. The max-potential of Ω is defined as

$$P(\Omega) := \max_{y \in \Omega} \phi(y),$$

and the point at which this maximum is attained is called the w-analytic center of Ω (see [8]).

We shall consider a scaled version of s. Specifically, we let

$$d_j = \frac{c_j - a_j^T y}{\sqrt{w_j}} = \frac{s_j}{\sqrt{w_j}}, \qquad j = 1, \ldots, n,$$

so that

$$D = \operatorname{diag}(d_1, \ldots, d_n) = W^{-1/2} S.$$

Also, let

$$\theta = (\sqrt{w_1}, \ldots, \sqrt{w_n})^T.$$

It is easy to see that the gradient and the Hessian of $\phi(y)$ can be written as

$$\nabla\phi = -AD^{-1}\theta$$

and

$$H = AD^{-2}A^T.$$

Consider the following normalized gradient vector of ϕ at y:

$$\Delta s(y) = D^{-1}A^T(AD^{-2}A^T)^{-1}\nabla\phi(y) = -D^{-1}A^T(AD^{-2}A^T)^{-1}AD^{-1}\theta$$

and

$$\delta(y) = \|\Delta s(y)\| = \|D^{-1}A^T(AD^{-2}A^T)^{-1}AD^{-1}\theta\|,$$

or

$$\begin{aligned}\delta(y)^2 &= \theta^T D^{-1}A^T(AD^{-2}A^T)^{-1}AD^{-1}\theta \\ &= (\nabla\phi)^T H^{-1}\nabla\phi.\end{aligned}$$

Note that the matrix H is symmetric positive semidefinite. Thus, this quantity $\delta(y)$ is a norm of the gradient vector $\nabla\phi$. Let

$$x(y) = D^{-1}(I - D^{-1}A^T(AD^{-2}A^T)^{-1}AD^{-1})\theta.$$

It can be checked that

$$\Delta s(y) = Dx(y) - \theta \quad \text{and} \quad \delta(y) = \|Dx(y) - \theta\|.$$

Clearly, if $\Delta s(y) = 0$, then we have $Sx(y) = w$, implying that y is the w-analytic center of Ω.

The following results are well known; see Atkinson [3, pp. 19–30].

LEMMA 2.1. *Let (y, s) be an interior point in Ω and let $\bar{y}$ be the w-analytic center of Ω.*

(i)*If $\mu \le 0.16$ and $\delta(y) \le \sqrt{2}\mu$, then*

$$\left| \frac{a_j^T(y - \bar{y})}{a_j^T y - c_j} \right| \le \sqrt{\mu}$$

and

$$\frac{1}{2}(1 - \sqrt{\mu})^2\delta(y)^2 \le \phi(\bar{y}) - \phi(y) \le \frac{1}{2}(1 + \sqrt{\mu})^2\delta(y)^2.$$

(ii)*If $\nu \le 0.008$ and $\phi(\bar{y}) - \phi(y) \le \nu$, then*

$$\frac{1}{2}(1 - 5\sqrt{\nu})^2\delta(y)^2 \le \phi(\bar{y}) - \phi(y) \le \frac{1}{2}(1 + 5\sqrt{\nu})^2\delta(y)^2.$$

(iii) *Suppose* $\phi(\bar{y}) - \phi(y) \leq 0.008$. *Define* $\Delta(y) = \nabla^2\phi(y)^{-1}\nabla\phi(y)$ *and let* $y^+ = y + 0.75\Delta(y)$. *Then*

$$\phi(\bar{y}) - \phi(y^+) \leq 0.68(\phi(\bar{y}) - \phi(y)).$$

The next lemma is crucial to the convergence analysis to be given later.

Lemma 2.2. *Suppose* $\mu \leq 0.16$ *and* $\delta(y) \leq \sqrt{2}\mu$. *Then*

$$\|\bar{D}^{-1}s - \theta\| = \left(\sum_{j=1}^{n} w_j(s_j/\bar{s}_j - 1)^2\right)^{1/2} \leq \left(\frac{1+\sqrt{\mu}}{1-\sqrt{\mu}}\right)\delta(y),$$

where $\bar{s}$ *and* $\bar{D}$ *are computed at the* w*-weighted analytic center* $\bar{y}$.

Proof. By part (i) of Lemma 2.1, we have

$$\left|\frac{a_j^T(y - \bar{y})}{a_j^T y - c_j}\right| \leq \sqrt{\mu}.$$

Then it follows that

$$\frac{1}{1+\sqrt{\mu}} \leq \frac{a_j^T y - c_j}{a_j^T \bar{y} - c_j} \leq \frac{1}{1-\sqrt{\mu}}.$$

This further implies that

$$\frac{1}{1+\sqrt{\mu}} \leq \frac{a_j^T y' - c_j}{a_j^T \bar{y} - c_j} \leq \frac{1}{1-\sqrt{\mu}} \tag{2.1}$$

for any y' lying in the line segment joining y and $\bar{y}$.

Since $\nabla\phi(\bar{y}) = 0$, it follows from Taylor expansion that

$$\phi(\bar{y}) - \phi(y) = -\frac{1}{2}(y - \bar{y})^T\nabla^2\phi(y')(y - \bar{y})$$

for some y' in the line segment joining y and $\bar{y}$. By estimate (2.1), we have

$$-\nabla^2\phi(y') = AW(S')^{-2}A^T \geq (1-\sqrt{\mu})^2 AW\bar{S}^{-2}A^T.$$

It then follows that

$$\begin{aligned}\phi(\bar{y}) - \phi(y) &\geq \frac{1}{2}(1-\sqrt{\mu})^2(y-\bar{y})^T AW\bar{S}^{-2}A^T(y-\bar{y})\\ &= \frac{1}{2}(1-\sqrt{\mu})^2\sum_{j=1}^{n} w_j(s_j/\bar{s}_j - 1)^2.\end{aligned}$$

Combining this with Lemma 2.1 (i) yields

$$\frac{1}{2}(1+\sqrt{\mu})^2\delta(y)^2 \geq \frac{1}{2}(1-\sqrt{\mu})^2\sum_{j=1}^{n} w_j(s_j/\bar{s}_j - 1)^2,$$

which implies the desired result. □

Finally, we establish an important lemma which will be used later to bound the potential reduction process. This lemma is an extension of a result by Ye [9] who considered the special case of $w = e$.

LEMMA 2.3. *Let $w \in \mathbb{R}^n$ be a positive weight vector such that*

$$\sum_{j=1}^{n} w_j = k, \qquad w_{\min} := \min_j w_j \leq 1, \tag{2.2}$$

where $k \geq 1$ is a positive scalar. Suppose α^* is a maximizer of the following maximization problem:

$$\begin{array}{ll} \textit{maximize} & f(\alpha) = \|W^{1/2}(\alpha - e)\| \prod_{j=1}^{n} \alpha_j^{w_j} \\ \textit{subject to} & w^T\alpha = k, \quad \alpha > 0. \end{array} \tag{2.3}$$

Then, $f(\alpha^*) \leq \gamma_1/\sqrt{w_{\min}}$, where γ_1 is an absolute constant.

Proof. For convenience, we consider the maximization problem

$$\begin{array}{ll} \text{maximize} & f^2(\alpha) = \left(\sum_{j=1}^{n} w_j(\alpha_j - 1)^2\right) \prod_{j=1}^{n} \alpha_j^{2w_j} \\ \text{subject to} & w^T\alpha = k, \quad \alpha > 0. \end{array} \tag{2.4}$$

Notice that

$$\begin{aligned} \sum_{j=1}^{n} w_j(\alpha_j - 1)^2 &= \sum_{j=1}^{n} w_j\alpha_j^2 - 2\sum_{j=1}^{n} w_j\alpha_j + \sum_{j=1}^{n} w_j \\ &= \sum_{j=1}^{n} w_j\alpha_j^2 - 2k + k \\ &= \sum_{j=1}^{n} w_j\alpha_j^2 - k. \end{aligned}$$

Taking the logarithm of $f^2(\alpha)$, the maximization problem (2.4) can be written equivalently as

$$\begin{array}{ll} \text{maximize} & \log f^2(\alpha) = \log\left(\sum_{j=1}^{n} w_j\alpha_j^2 - k\right) + \sum_{j=1}^{n} 2w_j \log \alpha_j \\ \text{subject to} & w^T\alpha = k, \quad \alpha > 0. \end{array} \tag{2.5}$$

The Kuhn–Tucker condition for (2.5) is, after simplification, given by

$$\begin{aligned} \frac{\alpha_j}{\rho} + \frac{1}{\alpha_j} &= \tau \qquad \forall j, \\ \sum_{j=1}^{n} w_j\alpha_j^2 - k &= \rho, \\ \sum_{j=1}^{n} w_j\alpha_j &= k, \end{aligned} \tag{2.6}$$

where τ is some dual multiplier. The quadratic equation (2.6) shows that the variables α_j can only take on two possible values, say, β and $\tilde{\beta}$, with $\beta\tilde{\beta} = \rho$. Let J denote the set of indices j such that $\alpha_j = \beta$ and $\tilde{J}$ denote its complement in $\{1, \ldots, n\}$. Thus, $\alpha_j = \tilde{\beta}$ for $j \in \tilde{J}$. Let us denote $w_J = \sum_{j\in J} w_j$ and, likewise, $w_{\tilde{J}} = \sum_{j\in\tilde{J}} w_j$. Then the above Kuhn–Tucker conditions can be further simplified as

$$\frac{\beta}{\rho} + \frac{1}{\beta} = \tau, \tag{2.7}$$

$$\frac{\tilde{\beta}}{\rho} + \frac{1}{\tilde{\beta}} = \tau, \tag{2.8}$$

$$w_J\beta^2 + w_{\tilde{J}}\tilde{\beta}^2 - k = \beta\tilde{\beta}, \tag{2.9}$$

$$w_J\beta + w_{\tilde{J}}\tilde{\beta} = k. \tag{2.10}$$

Let us evaluate the objective value at any point α satisfying (2.7)–(2.10):

$$\begin{aligned}
\log f^2(\alpha) &= \log\left(w_J\beta^2 + w_{\tilde{J}}\tilde{\beta}^2 - k\right) + 2w_J\log\beta + 2w_{\tilde{J}}\log\tilde{\beta} \\
&= \log\left(\beta\tilde{\beta}\right) + 2w_J\log\beta + 2w_{\tilde{J}}\log\tilde{\beta} \\
&= (2w_J+1)\log\beta + (2w_{\tilde{J}}+1)\log\tilde{\beta} \\
&= (2w_J+1)\log\left(\frac{w_J\beta}{2w_J+1}\right) + (2w_{\tilde{J}}+1)\log\left(\frac{w_{\tilde{J}}\tilde{\beta}}{2w_{\tilde{J}}+1}\right) \\
&\quad -(2w_J+1)\log\left(\frac{w_J}{2w_J+1}\right) - (2w_{\tilde{J}}+1)\log\left(\frac{w_{\tilde{J}}}{2w_{\tilde{J}}+1}\right).
\end{aligned}$$

Since $(2w_J+1)+(2w_{\tilde{J}}+1) = 2k+2$, it follows from (2.10) and the concavity of log that

$$\begin{aligned}
\log f^2(\alpha) \le{}& (2k+2)\log\frac{k}{2k+2} \\
&-(2w_J+1)\log\left(\frac{w_J}{2w_J+1}\right) - (2w_{\tilde{J}}+1)\log\left(\frac{w_{\tilde{J}}}{2w_{\tilde{J}}+1}\right).
\end{aligned}$$

Recall that $w_J + w_{\tilde{J}} = k$ and J is a subset of $\{1, \ldots, n\}$; it follows from elementary calculus that

$$(2w_J+1)\log\left(\frac{w_J}{2w_J+1}\right) + (2w_{\tilde{J}}+1)\log\left(\frac{w_{\tilde{J}}}{2w_{\tilde{J}}+1}\right)$$

attains its minimum with $w_{\tilde{J}} = w_{\min}$. With such a choice of $\tilde{J}$ (which is a singleton), we have

$$\begin{aligned}
\log f^2(\alpha) &\le (2k+2)\log\frac{2k}{2k+2} - (2w_J+1)\log\left(\frac{2w_J}{2w_J+1}\right) \\
&\quad -(2k+2)\log 2 + (2w_J+1)\log 2 - (2w_{\min}+1)\log\left(\frac{w_{\min}}{2w_{\min}+1}\right) \\
&= (2k+2)\log\frac{2k}{2k+2} - (2w_J+1)\log\left(\frac{2w_J}{2w_J+1}\right) \\
&\quad -(2w_{\min}+1)\log 2 - (2w_{\min}+1)\log\left(\frac{w_{\min}}{2w_{\min}+1}\right),
\end{aligned}$$

where we have used $w_J + w_{\min} = k$ in the last step. Since $w_{\min} \leq \min\{w_J, 1\}$ and $w_J \geq k/2 \geq 1/2$, we can see that the first three terms in the right-hand side of the above inequality are bounded for all k; the last term is of order $\log w_{\min}$. Therefore, there exists a $\gamma_1 > 0$ such that

$$\log f^2(\alpha) \leq 2 \log \gamma_1 - \log w_{\min},$$

implying $f(\alpha) \leq \gamma_1 / \sqrt{w_{\min}}$. This completes the proof. □

3. Max-potential reduction. Let

$$\Omega = \{y \in \mathbb{R}^m : \; c - A^T y \geq 0\},$$

defined by n $(> m)$ linear inequalities, be bounded and have a nonempty interior. Let the w-analytic center of Ω be $\bar{y}$. Define

$$\bar{s} = c - A^T \bar{y}, \qquad \bar{d} = \bar{S}\theta.$$

Suppose we have an approximate w-center y^k in the sense

$$\delta(y^k) \leq \sqrt{2}\mu \qquad \text{with } \mu \leq 0.16. \tag{3.1}$$

Let us place $\eta \geq 1$ cuts at y^k, that is, add η new inequalities $a_{n+i}^T y \leq a_{n+i}^T y^k + \beta r_{n+i}, i = 1, \ldots, \eta$, to Ω, and consider the new set

$$\Omega^+ = \{y : A^T y \leq c, \quad a_{n+i}^T y \leq c_{n+i}, \quad i = 1, \ldots, \eta\},$$

where

$$c_{n+i} := a_{n+i}^T y^k + \beta r_{n+i} \tag{3.2}$$

with $\beta > 0$ a constant (to be determined later, see Theorem 6.2) and

$$r_{n+i} = \sqrt{a_{n+i}^T (A(D^k)^{-2} A^T)^{-1} a_{n+i}}, \qquad i = 1, \ldots, \eta.$$

Again, we assume

$$\|a_{n+i}\| = 1, \qquad i = 1, \ldots, \eta.$$

Define $w_{n+i} = 1/\eta$ for $i = 1, \ldots, \eta$ and let

$$P(\Omega^+) = \max_{y \in \Omega^+} \sum_{j=1}^{n+\eta} w_j \log(c - A^T y)_j.$$

Also, we denote

$$\bar{r}_{n+i} = \sqrt{a_{n+i}^T (A\bar{D}^{-2} A^T)^{-1} a_{n+i}}, \qquad i = 1, \ldots, \eta.$$

Then we have the following result, which is an extension of Theorem 2 of Ye [9].

THEOREM 3.1. *Suppose $w \in \mathbb{R}^n$ is a positive weight vector satisfying* (2.2) *and define $w_{n+i} = 1/\eta$ for $i = 1, \ldots, \eta$. Let y^k be an approximate center as defined by* (3.1) *and let y^+ be the w-analytic center of Ω^+ and*

$$s_j^+ = (c - A^T y^+)_j, \quad s_{n+i}^+ = a_{n+i}^T y^k - a_{n+i}^T y^+ + \beta r_{n+i},$$

where $j = 1, \ldots, n$ and $i = 1, \ldots, \eta$. Then,

$$P(\Omega^+) = \sum_{j=1}^{n+\eta} w_j \log s_j^+ \le P(\Omega) + \frac{1}{\eta}\sum_{j=1}^{\eta} \log(\bar{r}_{n+i}) + \log\left(\frac{\gamma_1}{\sqrt{w_{\min}}} + \gamma_2 + \frac{5}{3}\beta\right),$$

where $w_{\min} := \min_j w_j \le 1$, and γ_1, γ_2 are some absolute constants independent of w, n or k.

Proof. Note that for $i = 1, \ldots, \eta$ we have

$$\begin{aligned} s_{n+i}^+ &= a_{n+i}^T(y^k - y^+) + \beta r_{n+i} \\ &= a_{n+i}^T(A\bar{D}^{-2}A^T)^{-1}(A\bar{D}^{-2}A^T)(y^k - y^+) + \beta r_{n+i} \\ &= a_{n+i}^T(A\bar{D}^{-2}A^T)^{-1}A\bar{D}^{-2}(A^T y^k - A^T y^+) + \beta r_{n+i} \\ &= a_{n+i}^T(A\bar{D}^{-2}A^T)^{-1}A\bar{D}^{-2}(-c + A^T y^k + c - A^T y^+) + \beta r_{n+i} \\ &= a_{n+i}^T(A\bar{D}^{-2}A^T)^{-1}A\bar{D}^{-2}(s^+ - \bar{s} + \bar{s} - s^k) + \beta r_{n+i} \\ &= a_{n+i}^T(A\bar{D}^{-2}A^T)^{-1}A\bar{D}^{-1}(\bar{D}^{-1}s^+ - \theta + \theta - \bar{D}^{-1}s^k) + \beta r_{n+i} \\ &\le \|a_{n+i}^T(A\bar{D}^{-2}A^T)^{-1}A\bar{D}^{-1}\| \left(\|\bar{D}^{-1}s^+ - \theta\| + \|\bar{D}^{-1}s^k - \theta\|\right) + \beta r_{n+i} \\ &= \bar{r}_{n+i}\left(\|\bar{D}^{-1}s^+ - \theta\| + \|\bar{D}^{-1}s^k - \theta\|\right) + \beta r_{n+i}. \end{aligned}$$

We shall bound r_{n+i} and $\|\bar{D}^{-1}s^k - \theta\|$ separately. First, by Lemma 2.1 (i), we have

$$\left|\frac{a_j^T(y^k - \bar{y})}{a_j^T y^k - c_j}\right| \le \sqrt{\mu},$$

which further implies

$$1 - \sqrt{\mu} \le \left|\frac{a_j^T\bar{y} - c_j}{a_j^T y^k - c_j}\right| \le 1 + \sqrt{\mu}.$$

Consequently, we obtain

$$\begin{aligned} r_{n+i} &= \sqrt{a_{n+i}^T(AW(S^k)^{-2}A^T)^{-1}a_{n+i}} \\ &\le \frac{1}{1-\sqrt{\mu}}\sqrt{a_{n+i}^T(AW(\bar{S})^{-2}A^T)^{-1}a_{n+i}} \\ &= \frac{\bar{r}_{n+i}}{1-\sqrt{\mu}} \\ &\le \frac{5}{3}\bar{r}_{n+i}, \quad i = 1, \ldots, \eta. \end{aligned} \tag{3.3}$$

On the other hand, since $\delta(y^k) \le \sqrt{2}\mu$ with $\mu \le 0.16$, we have from Lemma 2.2 that

$$\|\bar{D}^{-1}s^k - \theta\| \le \left(\frac{1+\sqrt{\mu}}{1-\sqrt{\mu}}\right)\delta(y^k) \le 0.528.$$

Thus, we obtain

$$s_{n+i}^+ \le \bar{r}_{n+i}\left(\|\bar{D}^{-1}s^+ - \theta\| + 0.528 + \frac{5}{3}\beta\right), \quad i = 1, \ldots, \eta.$$

Consider the following:

$$\begin{aligned}\frac{\exp P(\Omega^+)}{(\prod_1^\eta \bar{r}_{n+i})^{1/\eta}\exp P(\Omega)} &= \prod_{i=1}^{\eta}\left(\frac{s_{n+i}^+}{\bar{r}_{n+i}}\right)^{1/\eta}\prod_{j=1}^{n}\left(\frac{s_j^+}{\bar{s}_j}\right)^{w_j}\\ &\le \left(\|W^{1/2}(\bar{S}^{-1}s^+ - e)\| + 0.528 + \frac{5}{3}\beta\right)\prod_{j=1}^{n}\left(\frac{s_j^+}{\bar{s}_j}\right)^{w_j}.\end{aligned}\tag{3.4}$$

Note also that we have

$$w^T\bar{S}^{-1}s^+ = k.$$

This can be seen as follows:

$$\begin{aligned}w^T\bar{S}^{-1}s^+ &= w^T\bar{X}Ws^+\\ &= w^T\bar{X}W(c - A^Ty^+)\\ &= w^T\bar{X}Wc\\ &= w^T\bar{X}W(c - A^T\bar{y})\\ &= w^T\bar{X}W\bar{s}\\ &= w^Te = k.\end{aligned}$$

Let $\alpha = \bar{S}^{-1}s^+ \in \mathbb{R}^n$. Then, to bound the quantity of (3.3), we face the following maximization problem:

$$\begin{aligned}&\text{maximize} \qquad f(\alpha) = \left(\|W^{1/2}(\alpha - e)\| + 0.528 + \frac{5}{3}\beta\right)\prod_{j=1}^{n}\alpha_j^{w_j}\\ &\text{subject to} \qquad w^T\alpha = k, \quad \alpha > 0.\end{aligned}$$

By Lemma 2.3 and by

$$\prod_{j=1}^{n}\alpha_j^{w_j} \le \left(\frac{w_1\alpha_1 + \cdots + w_n\alpha_n}{k}\right)^k = 1,$$

this maximum value is bounded above by

$$\frac{\gamma_1}{\sqrt{w_{\min}}} + 0.528 + \frac{5}{3}\beta,$$

where γ_1 is a constant independent of w, n, or k. Thus, we have established

$$\frac{\exp P(\Omega^+)}{(\prod_1^\eta \bar{r}_{n+i})^{1/\eta}\exp P(\Omega)} \le \frac{\gamma_1}{\sqrt{w_{\min}}} + 0.528 + \frac{5}{3}\beta.$$

Setting $\gamma_2 = 0.528$ and taking logarithm completes the proof of the theorem. □

4. The weighted analytic center cutting plane algorithm with multiple cuts. Recall that $\Omega^0 = \{y \in \mathbb{R}^m : 0 \le y \le e\}$. Suppose there exists an oracle which for every $z \in \mathbb{R}^m$ either returns $z \in \Gamma$ or generates separating hyperplanes, $\{y : a_i^Ty \le a_i^Tz\} \supset \Gamma$, with $\|a_i\| = 1$. The weighted analytic center cutting plane algorithm is as follows:

- **Step 0 (Initialization)**
  Fix a constant $\beta > 0$ and let

$$A^0 = (I, -I) \in \mathbb{R}^{m \times 2m},$$

$$c^0 = \begin{pmatrix} e \\ 0 \end{pmatrix} \in \mathbb{R}^{2m},$$

$$y^0 = \frac{1}{2} e \in \mathbb{R}^m,$$

$$s^0 = c^0 - (A^0)^T y^0 = \frac{1}{2} e \in \mathbb{R}^{2m},$$

$$x^0 = 2e \in \mathbb{R}^{2m}$$

and

$$k = 0, \quad w^0 = e \in \mathbb{R}^{2m} \quad \text{and} \quad n_0 = 0.$$

- **Step 1 (Checking for Termination/Generating Cuts)**
  Let y^k be an approximate analytic center of $\Omega^k = \{y \in \mathbb{R}^m : c^k - (A^k)^T y \geq 0\}$, $s^k = c^k - (A^k)^T y^k > 0$ such that $\delta(y^k) < 0.06$. Query the oracle to see if $y^k \in \Gamma$ or not.
  If yes, stop; otherwise generate hyperplanes, $i = 1, \ldots, \eta_k, \{y : a_{n_k+i}^T y \leq a_{n_k+i}^T y^k + \beta r_{n_k+i}\} \supset \Gamma$ with $\|a_{n_k+i}\| = 1$, where

$$r_{n_k+i} = \sqrt{a_{n_k+i}^T (A^k W^k (S^k)^{-2} (A^k)^T)^{-1} a_{n_k+i}}$$

and $s^k = c^k - (A^k)^T y^k$. Let

$$\Omega^{k+1} = \{y \in \mathbb{R}^m : c^{k+1} - (A^{k+1})^T y \geq 0\}$$

where

$$A^{k+1} = \left(A^k, a_{n_k+1}, \ldots, a_{n_k+\eta_k}\right)$$

and

$$c^{k+1} = \begin{bmatrix} c^k \\ a_{n_k+1}^T y^k + \beta r_{n_k+1} \\ \vdots \\ a_{n_k+\eta_k}^T y^k + \beta r_{n_k+\eta_k} \end{bmatrix}. \tag{4.1}$$

- **Step 2 (Recentering)**
Define

$$w^{k+1} = \begin{bmatrix} w^k \\ \frac{1}{\eta_k} \\ \vdots \\ \frac{1}{\eta_k} \end{bmatrix}$$

and

$$\phi^{k+1}(y) = \sum_{j=1}^{n_k+\eta_k} w_j^{k+1} \log s_j^{k+1}(y)$$

with $s^{k+1}(y) = c^{k+1} - (A^{k+1})^T y$. Take dual Newton iterations

$$y^+ := y + 0.75(\nabla^2 \phi^{k+1}(y))^{-1} \nabla \phi^{k+1}(y) \tag{4.2}$$

starting from y^k, until a new approximate analytic center y^{k+1} is obtained with $\delta(y^{k+1}) < 0.06$. Set $n_{k+1} = n_k + \eta_k$, $k := k+1$, and return to Step 1.

The selection of the parameter $\beta > 0$ is usually done by the user. Its choice will affect the number of total cuts used by the algorithm as well as the number of Newton iterations (4.2) required to compute a new approximate analytic center in Step 2 of the algorithm. Theorem 6.2 gives one particular choice of β which ensures that seven Newton iterations are sufficient in the recentering step of the algorithm.

5. Bound on the total number of steps. Let $\{y^k\}$ be a sequence generated by the analytic center cutting plane method described in the previous section. We shall establish an upper bound for the total number of steps needed for the algorithm to find a solution in Γ. Throughout, we assume that the maximum number of planes that can enter at each step is bounded by η (≥ 1).

We first make two simple observations. First, since $w_j^k = 1/\eta_i$ for some i, we have $1/\sqrt{w^k{}_{\min}} \leq \sqrt{\eta}$. Second, since $\delta(y^k) \leq \sqrt{2}\mu$ with $\mu \leq 0.16$, it follows from Lemma 2.1 (i) (and using an argument similar to the one used for proving (3.3)) that

$$\bar{r}_{n_k+i} \leq (1+\sqrt{\mu}) r_{n_k+i} \leq 1.4 r_{n_k+i}.$$

With these two observations, Theorem 3.1 implies that the following relations, provided that termination has not occurred, hold for all $k \geq 0$:

$$\Gamma \supset \Omega^k \tag{5.1}$$

and

$$P(\Omega^{k+1}) \leq P(\Omega^k) + \frac{1}{\eta_k} \sum_{i=1}^{\eta_k} \log(r_{n_k+i}) + \log\left(\gamma_1\sqrt{\eta} + \gamma_2 + \frac{7}{3}\beta\right), \tag{5.2}$$

where γ_1 and γ_2 are some absolute constants independent of w, η, or k. (γ_1, γ_2 can be chosen to be 1.4 times the respective constants in Theorem 3.1.)

We need several lemmas to bound $P(\Omega^k)$.

LEMMA 5.1 (see [6]). *For all $k \geq 0$,*

$$P(\Omega^k) \geq (2m+k)\log \epsilon.$$

Proof. From (5.1), $\Gamma \subset \Omega^k$. Thus, Ω^k contains a full dimensional ball with radius ϵ. Let the center of this ball be $\bar{y}$; then $c^k - (A^k)^T \bar{y} \geq \epsilon e$. Thus,

$$\begin{aligned} P(\Omega^k) &= \sum_{j=1}^{2m+n_k} w_j^k \log(c^k - (A^k)^T y^k)_j \\ &\geq \sum_{j=1}^{2m+n_k} w^k \log(c^k - (A^k)^T \bar{y})_j \\ &\geq \sum_{j=1}^{2m+n_k} w_j^k \log \epsilon = (2m+k)\log\epsilon, \end{aligned}$$

as desired. □

LEMMA 5.2. *Let $s = c^k - (A^k)^T y \geq 0$ for any $y \in \Omega^k$. Then*

1. $0 \leq s_j \leq 1, \quad j = 1, \ldots, 2m,$
2. $0 \leq s_j \leq \sqrt{m} + \beta, \quad j = 2m+1, \ldots, n_k.$

Proof. The proof closely resembles the one in [6]. By feasibility, we have $s \geq 0$, so we only need to argue the upper bounds. For $j = 1, \ldots, m$, $s_j = 1 - y_j$; since $0 \leq y_j \leq 1$, it follows that $0 \leq s_j \leq 1$. Similarly, for $j = m+1, \ldots, 2m$, $s_j = y_{j-m}$; since $0 \leq y_{j-m} \leq 1$, we get $0 \leq s_j \leq 1$. For $j = 2m+1, \ldots, n_k$, it follows from the updating rule (4.1) that

$$\begin{aligned} s_j &= c_j^k - a_j^T y \\ &= a_j^T y^\ell + \beta r_{n_\ell + i} - a_j^T y \\ &\leq \|a_j\| \cdot \|y^\ell - y\| + \beta r_{n_\ell + i} \\ &\leq \|y^\ell - y\| + \beta r_{n_\ell + i} \qquad \text{(by } \|a_j\| = 1) \\ &\leq \sqrt{m} + \beta r_{n_\ell + i} \end{aligned}$$

for some $y^\ell \in \Omega^0$ and some $1 \leq i \leq \eta_\ell$, where $0 \leq \ell < k$. Note that the last inequality is due to the fact that for all pairs y^ℓ, $y \in \Omega^0$,

$$-e \leq y^\ell - y \leq e.$$

It remains to bound the term $r_{n_\ell + i}$. Recall that

$$\begin{aligned} A^\ell (S^\ell)^{-2} W^\ell (A^\ell)^T &= (Y^\ell)^{-2} + (I - Y^\ell)^{-2} + \sum_{j=2m+1}^{n_\ell} w_j^\ell \frac{a_j a_j^T}{(s_j^\ell)^2} \\ &\succeq (Y^\ell)^{-2} + (I - Y^\ell)^{-2} \\ &\succeq 8I, \end{aligned}$$

where the notation $U \succeq V$ means $U - V$ is positive semidefinite. Therefore, we have

$$r_{n_\ell + i} = \sqrt{a_{n_\ell + i}^T (A^\ell (S^\ell)^{-2} W^\ell (A^\ell)^T)^{-1} a_{n_\ell + i}} \leq \frac{1}{2\sqrt{2}}.$$

This implies that

$$s_j \leq \sqrt{m} + \beta r_{n_\ell+i} \leq \sqrt{m} + \beta,$$

which completes the proof. □

LEMMA 5.3. *Let $B^0 = 8I$ and*

$$B^{k+1} = B^k + \frac{1}{(\sqrt{m}+\beta)^2 \eta_k} \sum_{i=1}^{\eta_k} a_{n_k+i} a_{n_k+i}^T.$$

Then

$$A^k (S^k)^{-2} W^k (A^k)^T \succeq B^k.$$

That is,

$$A^k (S^k)^{-2} W^k (A^k)^T - B^k$$

is positive semidefinite.

Proof. Let $Y^k = \operatorname{diag}(y^k)$. Then

$$\begin{aligned} A^k (S^k)^{-2} W^k (A^k)^T &= (Y^k)^{-2} + (I - Y^k)^{-2} + \sum_{j=2m+1}^{n_k} w_j^k \frac{a_j a_j^T}{(s_j^k)^2} \\ &\succeq (Y^k)^{-2} + (I - Y^k)^{-2} + \frac{1}{(\sqrt{m}+\beta)^2} \sum_{j=2m+1}^{n_k} w_j^k a_j a_j^T \qquad \text{(by Lemma 5.2)} \\ &\succeq 8I + \frac{1}{(\sqrt{m}+\beta)^2} \sum_{j=2m+1}^{n_k} w_j^k a_j a_j^T \quad \text{(as } 0 \leq y^k \leq e\text{)} \\ &= B^k. \end{aligned}$$

This completes the proof of the lemma. □

The above lemma leads to the following lemma.

LEMMA 5.4. *Let y^k be the analytic center y^k of Ω^k, $s^k = c^k - (A^k)^T y^k$, and $(\omega_{n_k+i})^2 = a_{n_k+i}^T (B^k)^{-1} a_{n_k+i}$ for $i = 1, \ldots, \eta_k$. Then, for $i = 1, \ldots, \eta_k$,*

$$(\omega_{n_k+i})^2 \geq a_{n_k+i}^T (A^k (S^k)^{-2} W^k (A^k)^T)^{-1} a_{n_k+i} = (r_{n_k+i})^2. \tag{5.3}$$

This lemma implies that any upper bound on the sequence $\{\omega_j^2\}$ will lead to the same bound on the sequence $\{r_j^2\}$. The following lemma is the key to establish our main result; its proof is modelled after that of Ye [10, Lemma 3.5].

LEMMA 5.5. *For all $k \geq 1$, there holds*

$$\sum_{j=2m+1}^{n_{k+1}} w_j^{k+1} (\omega_j)^2 \leq \frac{18m(\sqrt{m}+\beta)^2}{15} \log\left(1 + \frac{k+1}{8m(\sqrt{m}+\beta)^2}\right).$$

Proof. Notice that

$$\begin{aligned} \det B^{k+1} &= \det\left(B^k + \frac{1}{\eta_k m} \sum_{i=1}^{\eta_k} a_{n_k+i} a_{n_k+i}^T\right) \\ &= \left(1 + \frac{\omega^2}{\eta_k (\sqrt{m}+\beta)^2}\right) \det\left(B^k + \frac{1}{\eta_k (\sqrt{m}+\beta)^2} \sum_{i=2}^{\eta_k} a_{n_k+i} a_{n_k+i}^T\right), \end{aligned}$$

where

$$\omega^2 = a_{n_k+1}^T \left(B^k + \frac{1}{\eta_k(\sqrt{m}+\beta)^2} \sum_{i=2}^{\eta_k} a_{n_k+i} a_{n_k+i}^T \right)^{-1} a_{n_k+1}.$$

Clearly,

$$\omega^2 \le a_{n_k+1}^T (B^k)^{-1} a_{n_k+1} = (\omega_{n_k+1})^2.$$

On the other hand, consider the matrix

$$I + \frac{1}{\eta_k(\sqrt{m}+\beta)^2} \sum_{i=2}^{\eta_k} (B^k)^{-1/2} a_{n_k+i} a_{n_k+i}^T (B^k)^{-1/2}.$$

We claim that its largest eigenvalue is at most 9/8. This is because for any $y \in \mathbb{R}^m$ and $\|y\| = 1$, we have

$$\begin{aligned} y^T & \left(I + \frac{1}{\eta_k(\sqrt{m}+\beta)^2} \sum_{i=2}^{\eta_k} (B^k)^{-1/2} a_{n_k+i} a_{n_k+i}^T (B^k)^{-1/2} \right) y \\ & = \|y\|^2 + \frac{1}{\eta_k(\sqrt{m}+\beta)^2} \sum_{i=2}^{\eta_k} (y^T (B^k)^{-1/2} a_{n_k+i})^2 \\ & \le \|y\|^2 + \frac{1}{\eta_k(\sqrt{m}+\beta)^2} \sum_{i=2}^{\eta_k} \|y\|^2 \|(B^k)^{-1/2} a_{n_k+i}\|^2 \\ & = 1 + \frac{1}{\eta_k(\sqrt{m}+\beta)^2} \sum_{i=2}^{\eta_k} a_{n_k+i}^T (B^k)^{-1} a_{n_k+i} \\ & \le 1 + \frac{1}{\eta_k(\sqrt{m}+\beta)^2} \sum_{i=2}^{\eta_k} \|a_{n_k+i}\|^2/8 \qquad (\text{by } B^k \succeq 8I) \\ & \le 1 + \frac{\eta_k - 1}{8\eta_k m(\sqrt{m}+\beta)^2} < 9/8. \end{aligned}$$

Thus,

$$\begin{aligned} \omega^2 & = a_{n_k+1}^T \left(B^k + \frac{1}{\eta_k(\sqrt{m}+\beta)^2} \sum_{i=2}^{\eta_k} a_{n_k+i} a_{n_k+i}^T \right)^{-1} a_{n_k+1} \\ & = a_{n_k+1}^T (B^k)^{-1/2} \left(I + \frac{\sum_{i=2}^{\eta_k} (B^k)^{-1/2} a_{n_k+i} a_{n_k+i}^T (B^k)^{-1/2}}{\eta_k(\sqrt{m}+\beta)^2} \right)^{-1} (B^k)^{-1/2} a_{n_k+1} \\ & \ge a_{n_k+1}^T (B^k)^{-1/2} \left(\frac{8}{9} I \right) (B^k)^{-1/2} a_{n_k+1} \\ & = \frac{8}{9} (\omega_{n_k+1})^2. \end{aligned}$$

This shows that

$$\begin{aligned} \log\det B^{k+1} & = \log\det \left(B_k + \frac{1}{\eta_k(\sqrt{m}+\beta)^2} \sum_{i=2}^{\eta_k} a_{n_k+i} a_{n_k+i}^T \right) + \log \left(1 + \frac{\omega^2}{\eta_k(\sqrt{m}+\beta)^2} \right) \\ & \ge \log\det \left(B_k + \frac{1}{\eta_k(\sqrt{m}+\beta)^2} \sum_{i=2}^{\eta_k} a_{n_k+i} a_{n_k+i}^T \right) + \log \left(1 + \frac{8(\omega_{n_k+1})^2}{9\eta_k(\sqrt{m}+\beta)^2} \right). \end{aligned}$$

Continuing this process for $i = 2, \ldots, \eta_k$, we have

$$\log\det B^{k+1} \geq \log\det B^k + \sum_{i=1}^{\eta_k} \log\left(1 + \frac{8(\omega_{n_k+i})^2}{9\eta_k(\sqrt{m}+\beta)^2}\right).$$

But, for $i = 1, \ldots, \eta_k$, we have from $B^k \succeq B^0 \succeq 8I$ that

$$\frac{8(\omega_{n_k+i})^2}{9\eta_k(\sqrt{m}+\beta)^2} \leq \frac{8}{9\eta_k(\sqrt{m}+\beta)^2}\frac{1}{8}\|a_{n_k+i}\|^2 \leq \frac{1}{9},$$

hence

$$\begin{aligned}\log\left(1 + \frac{8(\omega_{n_k+i})^2}{9\eta_k(\sqrt{m}+\beta)^2}\right) &\geq \frac{8(\omega_{n_k+i})^2}{9\eta_k(\sqrt{m}+\beta)^2} - \frac{\left(\frac{8(\omega_{n_k+i})^2}{9\eta_k(\sqrt{m}+\beta)^2}\right)^2}{2\left(1 - \frac{8(\omega_{n_k+i})^2}{9\eta_k(\sqrt{m}+\beta)^2}\right)}\\ &\geq \frac{15(\omega_{n_k+i})^2}{18\eta_k(\sqrt{m}+\beta)^2}.\end{aligned}$$

Thus, we have

$$\log\det B^{k+1} \geq \log\det B^k + \sum_{i=1}^{\eta_k} \frac{15(\omega_{n_k+i})^2}{18\eta_k(\sqrt{m}+\beta)^2}.$$

Using induction on k and noting that $w_j^{k+1} = 1/\eta_k$ for $j = n_k+1, \ldots, n_k+\eta_k$, we have

$$\begin{aligned}\log\det B^{k+1} &\geq \log\det B^0 + \sum_{j=2m+1}^{n_{k+1}} w_j^{k+1} \frac{15(\omega_j)^2}{18(\sqrt{m}+\beta)^2}\\ &= m\log 8 + \sum_{j=2m+1}^{n_{k+1}} w_j^{k+1} \frac{15(\omega_j)^2}{18(\sqrt{m}+\beta)^2}.\end{aligned}$$

However, by using the arithmetic–geometric inequality and Lemma 5.3, we have

$$\frac{1}{m}\log\det B^{k+1} \leq \log\frac{\operatorname{trace}(B^{k+1})}{m} = \log\left(8 + \frac{k+1}{m(\sqrt{m}+\beta)^2}\right).$$

Thus,

$$\sum_{j=2m+1}^{n_{k+1}} w_j^{k+1} \frac{15(\omega_j)^2}{18(\sqrt{m}+\beta)^2} \leq m\log\left(8 + \frac{k+1}{m(\sqrt{m}+\beta)^2}\right) - m\log 8.$$

This shows that

$$\sum_{j=2m+1}^{n_{k+1}} w_j^{k+1}(\omega_j)^2 \leq \frac{18m(\sqrt{m}+\beta)^2}{15}\log\left(1 + \frac{k+1}{8m(\sqrt{m}+\beta)^2}\right),$$

which completes the proof. □

Now we present our main result.

THEOREM 5.6. *Let the number of cuts generated in each step of the analytic center cutting plane algorithm be between* 1 *and* η. *Then, the algorithm stops with a solution in* Γ *as soon as* k *satisfies*

$$\frac{\epsilon^2}{(\gamma_1\sqrt{\eta}+\gamma_2+7\beta/3)^2} > \frac{\frac{m}{2}+\frac{18m(\sqrt{m}+\beta)^2}{15}\log\left(1+\frac{k+1}{8m(\sqrt{m}+\beta)^2}\right)}{2m+k+1},$$

where $\gamma_1 > 0$, $\gamma_2 > 0$ *are some absolute constants given by* (5.2).

Proof. If the algorithm does not terminate, then we have the following from relation (5.2) and Lemma 5.1:

$$\begin{aligned}(2m+k+1)\log\epsilon &\le P(\Omega^{k+1})\\ &\le P(\Omega^k)+\frac{1}{2\eta_k}\sum_{i=1}^{\eta_k}\log(r_{n_k+i})^2+\log\left(\gamma_1\sqrt{\eta}+\gamma_2+\frac{7}{3}\beta\right)\\ &\le P(\Omega^0)+\frac{1}{2}\sum_{j=2m+1}^{n_{k+1}}w_j^{k+1}\log(r_j)^2+(k+1)\log\left(\gamma_1\sqrt{\eta}+\gamma_2+\frac{7}{3}\beta\right)\\ &= 2m\log\frac{1}{2}+\frac{1}{2}\sum_{j=2m+1}^{n_{k+1}}w_j^{k+1}\log(r_j)^2+(k+1)\log\left(\gamma_1\sqrt{\eta}+\gamma_2+\frac{7}{3}\beta\right).\end{aligned}$$

Thus,

$$\begin{aligned}\log\epsilon-\log\left(\gamma_1\sqrt{\eta}+\gamma_2+\frac{7}{3}\beta\right) &\le \frac{1}{2(2m+k+1)}\left[2m\log\frac{1}{4}+\sum_{j=2m+1}^{n_{k+1}}w_j^{k+1}\log(r_j)^2\right]\\ &\le \frac{1}{2}\log\frac{2m\frac{1}{4}+\sum_{j=2m+1}^{n_{k+1}}w_j^{k+1}(r_j)^2}{2m+k+1}\\ &\qquad\text{(by the concavity of log and by } \textstyle\sum_j w_j^{k+1}=k+1)\\ &\le \frac{1}{2}\log\frac{\frac{m}{2}+\sum_{j=2m+1}^{n_{k+1}}w_j^{k+1}(\omega_j)^2}{2m+k+1}\\ &\le \frac{1}{2}\log\frac{\frac{m}{2}+\frac{18m(\sqrt{m}+\beta)^2}{15}\log\left(1+\frac{k+1}{8m(\sqrt{m}+\beta)^2}\right)}{2m+k+1},\end{aligned}$$

where the second-to-last step is due to Lemma 5.4 and the last step follows from Lemma 5.5. Equivalently, we have

$$\frac{\epsilon^2}{(\gamma_1\sqrt{\eta}+\gamma_2+7\beta/3)^2} \le \frac{\frac{m}{2}+\frac{18m(\sqrt{m}+\beta)^2}{15}\log\left(1+\frac{k+1}{8m(\sqrt{m}+\beta)^2}\right)}{2m+k+1}. \tag{5.4}$$

This proves the theorem. □

Theorem 5.1 implies that the complexity of the analytic center cutting plane algorithm, counted by the total number of calls to the oracle, is $O^*(\frac{\eta m(\sqrt{m}+\beta)^2}{\epsilon^2})$; the notation O^* means that lower-order terms are ignored. As we shall see in Theorem 6.2, the parameter β can be chosen as a constant (say, 1860). Thus, the total number of oracle calls is $O^*(\frac{\eta m^2}{\epsilon^2})$. This result improves the recent result of Ye [10] which has complexity $O^*(\frac{\eta^2 m^2}{\epsilon^2})$ and requires $\eta \leq m$.

We remark that one can also use the arbitrary weighting scheme

$$w^{k+1} = \begin{bmatrix} w^k \\ w^{k+1}_{n_k+1} \\ \vdots \\ w^{k+1}_{n_k+\eta_k} \end{bmatrix}$$

as long as

$$\sum_{i=1}^{\eta_k} w^{k+1}_{n_k+i} = 1.$$

The total complexity will be $O^*(\frac{m^2}{w_{\min}\epsilon^2})$, where $w_{\min}$ denotes the minimum weight placed on a cut.

6. Updating to a new center. In each step of the analytic center cutting plane algorithm, we need to compute an (approximate) weighted analytic center y^{k+1} of Ω^{k+1}. In this section, we show that y^{k+1} can be computed by the dual Newton procedure (4.2) starting from y^k in no more than seven iterations.

Throughout this section, all the slacks are evaluated at y^k, and therefore, for simplicity, we denote $s^{k+1}(y^k)$, $s^k(y^k)$, and $s_{n_k+i}(y^k)$ by s^{k+1}, s^k, and s_{n_k+i}, respectively. Furthermore, we denote

$$H_+ = A^{k+1}(S^{k+1})^{-2}W^{k+1}(A^{k+1})^T \quad \text{and} \quad H = A^k(S^k)^{-2}W^k(A^k)^T$$

and

$$g_+ = A^{k+1}(S^{k+1})^{-1}W^{k+1}e \quad \text{and} \quad g = A^k(S^k)^{-1}W^k e.$$

Clearly, we have

$$H_+ = H + \frac{1}{\eta_k}\sum_{i=1}^{\eta_k}\frac{a_{n_k+i}a^T_{n_k+i}}{(s^k_{n_k+i})^2}$$

and

$$g_+ = g + \frac{1}{\eta_k}\sum_{i=1}^{\eta_k}\frac{a_{n_k+i}}{s_{n_k+i}}.$$

We have the following lemma.

Lemma 6.1. *Let*

$$\delta(y^k) = \left(g^T H^{-1} g\right)^{1/2}.$$

Then

$$\delta^+(y^k) \equiv \left(g_+^T H_+^{-1} g_+\right)^{1/2} \leq 1.65\delta(y^k) + \frac{1.86}{\beta},$$

where β is the constant used in the definition of the new cuts (3.2).

Proof. Let

$$g_0 = g, \quad g_i = g + \frac{1}{\eta_k}\left(\frac{a_{n_k+1}}{s_{n_k+1}} + \cdots + \frac{a_{n_k+i}}{s_{n_k+i}}\right), \quad i = 1, \ldots, \eta_k.$$

Similarly, we let

$$H_0 = H, \quad H_i = H_0 + \frac{1}{\eta_k}\left(\frac{a_{n_k+1}a_{n_k+1}^T}{(s_{n_k+1}^k)^2} + \cdots + \frac{a_{n_k+i}a_{n_k+i}^T}{(s_{n_k+i}^k)^2}\right), \quad i = 1, \ldots, \eta_k.$$

Clearly, $g_{\eta_k} = g_+$ and $H_{\eta_k} = H_+$.

Since

$$H_i = H_{i-1} + \frac{1}{\eta_k}\frac{a_{n_k+i}a_{n_k+i}^T}{(s_{n_k+i}^k)^2},$$

it follows that $H^{-1} \succeq H_{i-1}^{-1} \succeq H_i^{-1}$ for $i = 1, \ldots, \eta_k$. Consider

$$\begin{aligned} g_i^T H_i^{-1} g_i &\le g_i^T H_{i-1}^{-1} g_i \\ &= \left(g_{i-1} + \frac{1}{\eta_k}\frac{a_{n_k+i}}{s_{n_k+i}}\right)^T H_{i-1}^{-1}\left(g_{i-1} + \frac{1}{\eta_k}\frac{a_{n_k+i}}{s_{n_k+i}}\right) \\ &= g_{i-1}^T H_{i-1}^{-1} g_{i-1} + 2\frac{g_{i-1}^T H_{i-1}^{-1} a_{n_k+i}}{\eta_k s_{n_k+i}} + \frac{a_{n_k+i}^T H_{i-1}^{-1} a_{n_k+i}}{\eta_k^2 s_{n_k+i}^2}. \end{aligned}$$

Notice that

$$\begin{aligned} \frac{a_{n_k+i}^T H_{i-1}^{-1} a_{n_k+i}}{s_{n_k+i}^2} &\le \frac{a_{n_k+i}^T H^{-1} a_{n_k+i}}{s_{n_k+i}^2} \\ &= \frac{r_{n_k+i}^2}{\beta^2 r_{n_k+i}^2} = \frac{1}{\beta^2} \end{aligned}$$

and

$$\begin{aligned} \frac{g_{i-1}^T H_{i-1}^{-1} a_{n_k+i}}{s_{n_k+i}} &\le \left(\sqrt{g_{i-1}^T H_{i-1}^{-1} g_{i-1}}\right)\left(\frac{\sqrt{a_{n_k+i}^T H_{i-1}^{-1} a_{n_k+i}}}{s_{n_k+i}}\right) \\ &\le \frac{1}{2}\left(g_{i-1}^T H_{i-1}^{-1} g_{i-1} + \frac{a_{n_k+i}^T H_{i-1}^{-1} a_{n_k+i}}{s_{n_k+i}^2}\right) \\ &\le \frac{1}{2}\left(g_{i-1}^T H_{i-1}^{-1} g_{i-1} + \frac{1}{\beta^2}\right). \end{aligned}$$

Therefore, we have

$$g_i^T H_i^{-1} g_i \le \left(1 + \frac{1}{\eta_k}\right) g_{i-1}^T H_{i-1}^{-1} g_{i-1} + \frac{1}{\beta^2}\left(\frac{1}{\eta_k} + \frac{1}{\eta_k^2}\right)$$

for $i = 1, \ldots, \eta_k$. Consequently, we see that

$$\begin{aligned} g_+^T H_+^{-1} g_+ &= g_{\eta_k}^T H_{\eta_k}^{-1} g_{\eta_k} \\ &\leq \left(1 + \frac{1}{\eta_k}\right)^{\eta_k} g^T H^{-1} g + \frac{1}{\beta^2 \eta_k} \sum_{i=1}^{\eta_k} \left(1 + \frac{1}{\eta_k}\right)^i \\ &= \left(1 + \frac{1}{\eta_k}\right)^{\eta_k} g^T H^{-1} g + \frac{1}{\beta^2} \left(1 + \frac{1}{\eta_k}\right) \left(\left(1 + \frac{1}{\eta_k}\right)^{\eta_k} - 1\right) \\ &\leq 2.72 g^T H^{-1} g + \frac{3.44}{\beta^2}. \end{aligned}$$

Thus, we obtain

$$\delta^+(y^k) = \sqrt{g_+^T H_+^{-1} g_+} \leq \sqrt{2.72 g^T H^{-1} g + \frac{3.44}{\beta^2}} \leq 1.65 \delta(y^k) + \frac{1.86}{\beta},$$

as desired. □

We are now ready to establish the main result of this section.

THEOREM 6.2. *Suppose $\beta = 1860$ and $\mu \leq 0.0424$. Then the dual Newton procedure* (4.2)*, when initialized at y^k satisfying $\delta(y^k) \leq \sqrt{2}\mu \leq 0.06$, will generate an approximate analytic center y^{k+1} for Ω^{k+1} with $\delta(y^{k+1}) \leq 0.06$ in* seven *iterations.*

Proof. Let

$$\phi^+(y) = \sum_{j=1}^{n_{k+1}} w_j^{k+1} \log(c^{k+1} - (A^{k+1})^T y)_j$$

and let $\bar{y}^+$ denote its maximizer (the analytic center of Ω^{k+1}). Since $\delta(y^k) \leq \sqrt{2}\mu \leq 0.06$ and $\beta \geq 1860$, it follows from Lemma 6.1 that

$$\delta^+(y^k) \leq 1.65 \times 0.06 + \frac{1.86}{1860} \leq 0.1.$$

By Lemma 2.1 (i) we get

$$\phi^+(\bar{y}^+) - \phi^+(y^k) \leq \frac{1}{2} \left(1 + \sqrt{\delta^+(y^k)/2^{1/2}}\right)^2 \delta^+(y^k)^2 \leq 0.008.$$

Let y^{k+1} denote the iterate obtained after performing seven dual Newton iterations. Then, we have from Lemma 2.1 (iii) that

$$\phi^+(\bar{y}^+) - \phi^+(y^{k+1}) \leq (0.68)^7 \left(\phi^+(\bar{y}^+) - \phi^+(y^k)\right) \leq (0.68)^7 \times 0.008 = 5.3784 \times 10^{-4}.$$

Finally, we use Lemma 2.1 (ii) to bound

$$\delta^+(y^{k+1}) \leq \left(\frac{\phi^+(\bar{y}^+) - \phi^+(y^{k+1})}{0.5(1 - 5\sqrt{0.008})^2}\right)^{1/2} \leq 0.06.$$

Thus, y^{k+1} is sufficiently close to the new analytic center $\bar{y}^+$. □

Combining Theorems 5.1 and 6.1, we see that the analytic center cutting plane algorithm (with multiple cuts added at each iteration) will terminate in at most

$O^*(\frac{\eta m^2}{\epsilon^2})$ Newton steps. This is a factor of η faster than Ye's algorithm (see [10]) which uses the unweighted analytic center (i.e., $w_i^k = 1$ for all k and i). It should be pointed out that the constants in our analysis have not been fully optimized, and it is quite likely that a much smaller value of β may work in Theorem 6.2. Finally, the above complexity bound in the multiple cut case is worse (by a factor of η) than that in the single cut case which has a complexity of only $O^*(\frac{m^2}{\epsilon^2})$. It will be interesting to close this gap.

Acknowledgment. It is a pleasure to thank both Professor Y. Ye for his encouragement in this work and a referee for pointing out a mistake in the proof of Lemma 5.2.

REFERENCES

[1] A. Altman and K. C. Kiwiel, *A Note on Some Analytic Center Cutting Plane Methods for Convex Feasibility and Minimization Problems*, Technical report, Systems Research Institute, Newelska 6, 01-447, Warsaw, Poland, June 1994.

[2] D. S. Atkinson and P. M. Vaidya, *A cutting plane algorithm for convex programming that uses analytic centers*, Math. Programming B, 69 (1995), pp. 1–44.

[3] D. S. Atkinson, *Scaling and Interior Point Methods in Optimization*, Ph.D. dissertation, Coordinated Science Laboratory, College of Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, 1992.

[4] O. Bahn, O. Du Merle, J.-L. Goffin, and J.-P. Vial, *Experimental behavior of an interior point cutting plane algorithm for convex programming: An application to geometric programming*, Discrete Appl. Math., 49 (1994), pp. 3–23.

[5] O. Bahn, O. Du Merle, J.-L. Goffin, and J.-P. Vial, *A cutting plane method from analytic centers for stochastic programming*, Math. Programming B, 69 (1995), pp. 45–74.

[6] J.-L. Goffin, Z.-Q. Luo, and Y. Ye, *Complexity analysis of an interior cutting plane method for convex feasibility problems*, SIAM J. Optim., 6 (1996), pp. 638–652.

[7] Y. Nesterov, *Cutting plane algorithms from analytic centers: Efficiency estimates*, Math. Programming B, 69 (1995), pp. 149–176.

[8] G. Sonnevend, *New algorithms in convex programming based on a notion of 'centre' (for systems of analytic inequalities) and on rational extrapolation*, in Trends in Mathematical Optimization: Proceedings of the 4th French-German Conference on Optimization in Irsee, K. H. Hoffmann, J. B. Hiriat-Urruty, C. Lemarechal, and J. Zowe, eds., West Germany, April 1986.

[9] Y. Ye, *A potential reduction algorithm allowing column generation*, SIAM J. Optim., 2 (1992), pp. 7–20.

[10] Y. Ye, *Complexity analysis of the analytic center cutting plane method that uses multiple cuts*, Math. Programming, to appear.

SIAM J. Optim.
Vol. 7, No. 3, pp. 717–731, August 1997

A TRUST REGION INTERIOR POINT ALGORITHM FOR LINEARLY CONSTRAINED OPTIMIZATION[*]

J. FRÉDÉRIC BONNANS[†] AND CECILIA POLA[‡]

Abstract. We present an extension, for nonlinear optimization under linear constraints, of an algorithm for quadratic programming using a trust region idea introduced by Ye and Tse [Math. Programming, 44 (1989), pp. 157–179] and extended by Bonnans and Bouhtou [RAIRO Rech. Opér., 29 (1995), pp. 195–217]. Due to the nonlinearity of the cost, we use a linesearch in order to reduce the step if necessary. We prove that, under suitable hypotheses, the algorithm converges to a point satisfying the first-order optimality system, and we analyze under which conditions the unit stepsize will be asymptotically accepted.

Key words. trust region, quadratic model, linesearch, interior points

AMS subject classifications. 90C30, 65K05, 49M40

PII. S1052623493250639

1. Introduction.

In this paper, we study an algorithm for minimizing a nonlinear cost under linear constraints. We consider problems with linear equality constraints and nonnegative variables. At each step, a direction is computed by minimizing a convex quadratic model over an ellipsoidal trust region, and then a linesearch of Armijo type is performed in this direction. At each iteration, the ellipsoid of the quadratic problem is so small that it forces the nonnegativity constraints to be satisfied. However, the ellipsoid is not necessarily contained in the set of feasible points.

In the case of linear programming (LP) or convex quadratic programming (QP), we may assume the quadratic model to be equal to the cost function. Then the unit step will be accepted by the linesearch. In the case of LP, the algorithm is then reduced to the celebrated Dikin's algorithm [10] (see also Tsuchiya [26]). Ye and Tse [27] have extended this algorithm to convex quadratic programming using the trust region idea. This problem was also considered by Sun [25]. Bonnans and Bouhtou [2] studied such methods for nonconvex quadratic problems by taking a variable size for the trust region. An early extension of trust region algorithms to nonlinear costs is done in Dikin and Zorkalcev [11]. Among the related work, we quote Gonzaga and Carlos [13]. Interior point algorithms for the solution of constrained convex optimization problems have been studied by many other researchers; see, for instance, Den Hertog, Roos, and Terlaky [8], Jarre [15], Mehrotra and Sun [19], McCormick [18], Monteiro and Adler [20], Dennis, Heinkenschloss, and Vicente [9], and Coleman and Li [7]. Gonzaga [14] explores the shape of the trust regions to generate ellipsoidal regions adapted to the shape of the feasible set. The resulting algorithm generates sequences of points in the interior of the feasible set.

In this paper, we obtain some results of global convergence, comparable to those obtained in [2] for QP; by global convergence we only mean that the limit points of the sequence generated by the algorithm satisfy the first-order optimality system. The main novelty of the paper, however, is in the local analysis in the vicinity of

[*] Received by the editors June 17, 1993; accepted for publication (in revised form) January 19, 1996. This research was partially supported by DGICYT (Madrid).
http://www.siam.org/journals/siopt/7-3/25063.html

[†] INRIA, Domaine de Voluceau, 78153 Rocquencourt, France (frederic.bonnans@inria.fr).

[‡] Dpto Mat., Estad. y Comp.-Facultad de Ciencias, 39005 Santander, Spain (pola@matsun3.unican.es).

a local solution satisfying some strong second-order sufficient conditions. We check that if such a point is a limit point of the sequence computed by the algorithm and is under a "sufficient curvature" condition satisfied by the Hessian of the quadratic approximation, then the sequence actually converges to this point and the unit step is asymptotically accepted. Unfortunately, the acceptance of the unit step is not by itself a guarantee of a rapid convergence (the convergence might be linear at a very poor rate). The interest of the result lies in the fact that in the case of convex QP, this type of algorithm converges reasonably well in practice, although the convergence rate is only linear (see, e.g., the numerical results reported in Bonnans and Bouhtou [2] and Bouhtou [5]). Therefore, the question is to know to which extent the features of Dikin's type algorithms may be kept when dealing with nonlinear cost functions. In particular, we do not expect the rate of convergence of the cost to be superlinear, as this is not the case for quadratic programs.

The paper is organized as follows. In section 2 we present the algorithm and give a result of global convergence in the sense that under some convenient hypotheses, the sequence computed by the algorithm converges towards a point satisfying the first-order optimality system. Then, in section 3 we perform the local analysis: we check that if the sequence computed by the algorithm has some regular limit point $\bar{x}$ and if a condition of "sufficient curvature" holds, then the sequence converges to this point and the unit step is asymptotically accepted.

2. The algorithm. We consider the following problem:

$$\text{(P)} \qquad \min f(x); Ax = b; x \geq 0,$$

where f is a smooth mapping from $\mathbb{R}^n$ in $\mathbb{R}$, not necessarily convex; A is a $p \times n$ matrix; and $b \in \mathbb{R}^p$. We define the following sets:

$$F := \{x \in \mathbb{R}^n; Ax = b, x \geq 0\},$$

$$\overset{\circ}{F} := \{x \in \mathbb{R}^n; Ax = b; x > 0\},$$

so that F is the set of feasible points and $\overset{\circ}{F}$ is the set of "strictly feasible" points. In the sequel, we assume that F is bounded and $\overset{\circ}{F}$ is nonempty.

The algorithm will use two matrices at each iteration. The first is $X_k :=$ diag (x^k), where $\{x^k\}$ is the current feasible point. This is a scaling matrix that takes care of the positivity constraints. The second matrix is M_k, a symmetric approximation of the Hessian of the cost function. We assume M_k to be positive semidefinite (i.e., $d^t M_k d \geq 0$ for all d in $\mathbb{R}^n$). We consider the following algorithm.

Algorithm 1.

0) *Choose* $x^\circ \in \overset{\circ}{F}, \delta \in (0,1), \beta \in (0,1), \gamma \in (0,1); k \leftarrow 0.$

1) *Choose an* $n \times n$ *symmetric matrix* M_k. *Compute* δ_k *in* $(\delta, 1/\delta)$ *such that the point* d^k *that solves*

$$\text{(SP)} \qquad \min_d \varphi_k(d) := f(x^k) + \nabla f(x^k)^t d + \frac{1}{2} d^t M_k d;\ Ad = 0;\ d^t X_k^{-2} d \leq \delta_k^2$$

satisfies $x^k + d^k > 0.$

2) *If* $\varphi_k(d^k) = f(x^k)$, *stop.*

3) *Linesearch: Compute* $\rho^k = \beta^{\ell_k}$, *with* ℓ_k *the smallest nonnegative integer such that*

$$f(x^k) - f(x^k + \beta^{\ell_k} d^k) \geq \gamma \beta^{\ell_k} (f(x^k) - \varphi_k(d^k)). \tag{1}$$

4) $x^{k+1} = x^k + \rho^k d^k$; $k \leftarrow k+1$. *Go to 1.*

Some comments are needed to clarify the description of the algorithm. First, let us note that the stopping criterion of step 2 is, of course, unrealistic. The algorithm will typically never stop. This is convenient for studying the asymptotic properties of the sequence generated by the algorithm. A practical stopping criterion might require that we stop when $\varphi_k(d^k)$ is close enough to $f(x^k)$. Because the cost function may be nonconvex, there is, of course, no guarantee that the limit points are close to a global or even local solution (our results below deal with the optimality system at the limit points).

Our second comment deals with the fact that we allow δ_k to be greater or equal to 1. If we specify a value of δ_k smaller than 1, then we automatically have $x^k + d^k > 0$. What is the meaning of allowing $\delta_k \geq 1$? In order to understand that, let us observe that the trust region problem (SP) cannot be solved directly because of the nonlinear constraint (see, e.g., Moré [21], Sorensen [24]). Instead, one typically solves a sequence of equality constrained quadratic problems of type

$$\min_d f(x^k) + \nabla f(x^k)^t d + \frac{1}{2} d^t M_k d + \frac{\nu}{2} d^t X_k^{-2} d; \; Ad = 0,$$

where $\nu \geq 0$ is an estimate of ν_k (the Lagrange multiplier associated with the nonlinear constraint). As M_k is semidefinite positive for any $\nu > 0$, this problem has a unique solution $d = d(\nu)$, and the mapping $\nu \rightarrow d^t(\nu) X_k^{-2} d(\nu)$ is strictly decreasing. Let us say that ν is *too small* if either $d^t(\nu) X_k^{-2} d(\nu) > 1/\delta$ or $\min_i x_i^k + d_i(\nu) < 0$ and *too large* if $d^t(\nu) X_k^{-2} d(\nu) < \delta$ (as $\delta < 1$, it follows in that case that $\min_i x_i^k + d_i(\nu) > 0$). A dichotomic procedure based on these notions of "too large" and "too small" allows us to compute a solution of (SP) with $\delta_k \in (\delta, 1/\delta)$ in a finite number of steps; this is associated with a value of δ_k that may be greater than 1. It was observed already in [2] that to allow the possibility that $\delta_k \geq 1$ may speed up the convergence, and therefore it is worth taking this possibility into account in the analysis.

We note that if the algorithm stops at iteration k, then x^k satisfies the first-order optimality condition of (P). To see this, we need the following lemma, which states the optimality system of (SP). This is a simple extension of the known result for problems without equality constraints; see [6].

LEMMA 2.1. *The point* d^k *that solves* (SP) *is characterized by the existence of* λ^{k+1} *in* $\mathbb{R}^p$, $\nu_k \geq 0$ *such that*

$$\nabla f(x^k) + M_k d^k + A^t \lambda^{k+1} + \nu_k X_k^{-2} d^k = 0, \tag{2}$$

$$Ad^k = 0, \tag{3}$$

$$\nu_k \geq 0, \; (d^k)^t X_k^{-2} d^k \leq \delta_k^2, \; \nu_k [(d^k)^t X_k^{-2} d^k - \delta_k^2] = 0. \tag{4}$$

We now come back to the discussion of step 2 of the algorithm. Using (2), we deduce that

$$\begin{aligned} f(x^k) - \varphi_k(d^k) &= -\nabla f(x^k)^t d^k - \frac{1}{2}(d^k)^t M_k d^k, \\ &= (\lambda^{k+1})^t A d^k + \nu_k (d^k)^t X_k^{-2} d^k + \frac{1}{2}(d^k)^t M_k d^k. \end{aligned}$$

Using (3) and (4), we get

$$f(x^k) - \varphi_k(d^k) = \nu_k \delta_k^2 + \frac{1}{2}(d^k)^t M_k d^k. \tag{5}$$

So, if $f(x^k) = \varphi_k(d^k)$, as M_k is a positive semidefinite matrix, then each of the nonnegative terms on the right-hand side is equal to 0. We deduce that $\nu_k = 0$ and $M_k^{1/2} d^k = 0$, so $M_k d^k = 0$, where $(M_k)^{1/2}$ is the square root of the symmetric positive semidefinite matrix M_k. That is,

$$(M_k)^{1/2} = \sum_{i=1}^{n} (\lambda_i)^{1/2} u^i (u^i)^t,$$

where $\{\lambda_i, u^i\}$, $i = 1$ to n, are the eigenvalues and associated orthonormal eigenvectors of M_k. Hence, again using (2), we get

$$\begin{aligned} &\nabla f(x^k) + A^t \lambda^{k+1} = 0, \\ &A x^k = b, \ x^k > 0. \end{aligned}$$

So, x^k satisfies the first-order optimality condition of (P).

In the sequel, when studying the convergence of the algorithm, we will assume that it generates an infinite sequence of iterates.

Remark 2.1. From Lemma 2.1 it follows that the convex quadratic function

$$\psi_k(x) := \varphi_k(x - x^k) + \nu_k (x - x^k)^t X_k^{-2} (x - x^k)/2$$

attains its minimum on $\{x \in \mathbb{R}^n; Ax = b\}$ at $x^k + d^k$.

In step 3, we see that the linesearch is of Armijo type [1], i.e., it consists simply of testing the unit step, then reducing the step by a factor $\beta < 1$ until a convenient point is found. We note that this linesearch is well defined because, as M_k is positive semidefinite, the function φ_k is convex. It follows that

$$\nabla f(x^k)^t d^k = \nabla \varphi_k(0)^t d^k \leq \varphi_k(d^k) - \varphi_k(0) = \varphi_k(d^k) - f(x^k),$$

hence, for $\rho > 0$ small enough,

$$\begin{aligned} f(x^k) - f(x^k + \rho d^k) &= -\rho \nabla f(x^k)^t d^k + o(\rho), \\ &\geq \rho [f(x^k) - \varphi_k(d^k)] + o(\rho). \end{aligned}$$

As $\gamma \in (0, 1)$ and $f(x^k) > \varphi_k(d^k)$, condition (1) is satisfied whenever ℓ_k is large enough.

For the statement of the result of global convergence, we need some definitions. Given $x \in F$, we denote the set of active constraints by

$$I(x) := \{i \in \{1, \ldots, n\}; x_i = 0\}.$$

To any $I \subset \{1, \ldots, n\}$ we associate the optimization problem

$$(\mathrm{P})_\mathrm{I} \qquad \min f(x); Ax = b; x_I = 0.$$

The first-order optimality system associated to $(P)_I$ is

$$(\mathrm{OS})_\mathrm{I} \qquad \begin{cases} \nabla f(x) + A^t\lambda - \mu = 0, \\ Ax = b, \\ x_I = 0;\ \mu_i = 0, i \notin I. \end{cases}$$

We will use the following hypotheses:

(H1) For all $I \subset \{1, \ldots, n\}$, system $(\mathrm{OS})_\mathrm{I}$ has no nonisolated solutions.

(H1)$'$ For all $I \subset \{1, \ldots, n\}$, system $(\mathrm{OS})_\mathrm{I}$ has at most one solution.

(H2) There exists $\alpha > 0$; $(d^k)^t(M_k + 2\nu_k X_k^{-2})d^k \geq \alpha\|d^k\|^2$.

$$(\mathrm{H3}) \qquad \begin{cases} \text{The constraints of (P) are qualified in the sense that} \\ (A^t\lambda)_i = 0, \forall\, i \notin I(\bar{x}) \text{ implies that } \lambda = 0. \end{cases}$$

We briefly discuss these hypotheses. If f is strictly convex, then the optimality system $(\mathrm{OS})_\mathrm{I}$, which characterizes the minima of F over the feasible set of $(\mathrm{P})_\mathrm{I}$, has at most one primal solution; therefore, if (H3) is satisfied in addition, then (H1)$'$ will be satisfied. (H1) is a weaker condition that may be useful especially for nonconvex problems. Hypothesis (H2) is a means that allows control of the decrease of the cost function at each iteration. Indeed, from (5) it follows easily that (H2) is equivalent to

$$\text{there exists } \alpha > 0 \text{ ; } f(x^k) - \varphi_k(d^k) \geq \frac{\alpha}{2}\|d^k\|^2.$$

We have no control on the value of ν_k, except that it is nonnegative. Still, we may observe that (H2) will be satisfied if M_k is uniformly positive definite in the sense that

$$\text{there exists } \alpha > 0 \text{ ; } (d^k)^t M_k d^k \geq \alpha\|d^k\|^2.$$

In particular, (H2) is satisfied if M_k is close to the Hessian of f and f satisfies a strong convexity condition of the type

$$\forall x \in F,\ \exists \alpha > 0 \text{ ; } (d^k)^t\nabla^2 f(x)d^k \geq \alpha\|d^k\|^2\ \forall d \in \mathbb{R}^n;\ Ad = 0.$$

Also, (H3) is no more than the hypothesis of linear independence of the gradients of active constraints.

THEOREM 2.2. *Let $\{x^k\}$ be computed by Algorithm* 1. *We assume that $\{M_k\}$ is bounded. Then,*

(i) *any limit point $\bar{x}$ of $\{x^k\}$ is a solution of* $(\mathrm{OS})_{\mathrm{I}(\bar{x})}$;

(ii) *if either* (H1)$'$ *or* (H1) *and* (H2) *hold, then $\{x^k\}$ converges. If, in addition,* (H3) *holds then $\bar{x}$ satisfies the first-order optimality system of* (P); *i.e.,*

$$(\mathrm{OS}) \qquad \begin{cases} \nabla f(\bar{x}) + A^t\bar{\lambda} - \bar{\mu} = 0, \\ A\bar{x} = b, \\ \bar{x} \geq 0, \bar{\mu} \geq 0, \bar{x}^t\bar{\mu} = 0. \end{cases}$$

The proof of the theorem uses the following lemma.

LEMMA 2.3. *The sequence $\{x^k\}$ generated by Algorithm* 1 *satisfies the following conditions:*

(i) $\sum_k (f(x^k) - \varphi_k(d^k))^2 < \infty$.

(ii) $\nu_k \to 0$.

(iii) $(M_k)^{1/2} d^k \to 0$.

(iv) *If, in addition, $\{M_k\}$ is bounded, then*

$$X_k[\nabla f(x^k) + A^t \lambda^{k+1}] \to 0.$$

Proof. (i) As F is bounded, $\{x^k\}$ and $\{d^k\}$ are bounded too. We deduce that for some $c_1 > 0$,

$$f(x^k) - f(x^k + \rho d^k) \geq -\rho \nabla f(x^k)^t d^k - c_1 \rho^2.$$

Using the convexity of φ_k, we get

$$-\nabla f(x^k)^t d^k \geq f(x^k) - \varphi_k(d^k),$$

so that

$$f(x^k) - f(x^k + \rho d^k) \geq \rho[f(x^k) - \varphi_k(d^k)] - c_1 \rho^2.$$

It follows after some algebra that the linesearch test is satisfied whenever

$$\rho \leq \hat{\rho}_k := \min\left\{1, \frac{1-\gamma}{c_1}[f(x^k) - \varphi_k(d^k)]\right\}.$$

This implies that $\rho^k \geq \beta \hat{\rho}_k$. Plugging this in the linesearch test and using the fact that as F is bounded, $\{f(x^k)\}$ is bounded from below, we deduce that necessarily $(f(x^k) - \varphi_k(d^k))$ vanishes and, for k large enough,

$$f(x^k) - f(x^{k+1}) \geq \gamma \beta \frac{1-\gamma}{c_1}(f(x^k) - \varphi_k(d^k))^2.$$

Relation (i) follows.

(ii), (iii) By (i), we get that the left-hand side of (5) goes to 0. Then each of the nonnegative terms on the right-hand side must go to 0, and that proves (ii) and (iii).

(iv) From (2) we deduce

$$X_k[\nabla f(x^k) + A^t \lambda^{k+1}] = -\nu_k X_k^{-1} d^k - X_k M_k d^k. \tag{6}$$

From (4) we have that $\|\nu_k X_k^{-1} d^k\|_2 = \nu_k \delta_k$. So, using Lemma 2.3 (ii) and the boundedness of $\{\delta_k\}$, it follows that $\|\nu_k X_k^{-1} d^k\| \to 0$. If, in addition, $\{M_k\}$ is bounded, we get that $X_k M_k d^k = X_k (M_k)^{1/2} (M_k)^{1/2} d^k \to 0$ by using the boundedness of $\{X_k\}$ and Lemma 2.3 (iii). Henceforth, the left-hand side of (6) goes to 0. □

Proof of Theorem 2.1. (i) Let us denote by $R(.)$ the range of an operator. Define

$$\bar{I} := \{1, \ldots, n\} - I(\bar{x}).$$

From point (iv) of Lemma 2.3 it follows that

$$[\nabla f(x^k) + A^t \lambda^{k+1}]_{\bar{I}} \to 0.$$

Since $R\,(A^t)_{\bar{I}}$ is closed, we deduce that $\nabla f(\bar{x})_{\bar{I}} \in \ R\,(A^t)_{\bar{I}}$, i.e., $(\nabla f(\bar{x}) + A^t \bar{\lambda})_{\bar{I}} = 0$ for some $\bar{\lambda} \in \mathbb{R}^p$; system $(\mathrm{OS})_{\mathrm{I}(\bar{x})}$ follows.

(ii) We first discuss the convergence of $\{x^k\}$. Note that $x_i^{k+1} = x_i^k(1 + \rho^k d_i^k / x_i^k)$; hence,

$$x_i^{k+1} \leq (1 + 1/\delta) x_i^k.$$

It follows that if $(x^k, x^{k+1}) \to (\bar{x}, \hat{x})$ for a subsequence, then $I(\bar{x}) \subset I(\hat{x})$.

If (H1)$'$ holds, using point (i) we deduce that $\bar{x} = \hat{x}$ and, in particular, $\|x^{k+1} - x^k\| \to 0$; hence, the set of limit points of $\{x^k\}$ is connected. Using (H1)$'$ again, it follows that the set of limit points is finite. Hence, the entire sequence converges towards the same point.

Now let us analyze the case when (H1) and (H2) hold. We know by Lemma 2.3 (i) that $f(x^k) - \varphi_k(d^k) \to 0$. With (5) and (H2), this implies that $d^k \to 0$. As $\|x^{k+1} - x^k\| = \rho^k \|d^k\|$ and $\rho_k \leq 1$, the set of limit points of $\{x^k\}$ is connected. By (i) and (H1) each of them is isolated. It follows that the sequence converges.

We now prove that (OS) is satisfied under the additional assumption (H3). If $x^k \to \bar{x}$ then there exists $(\bar{\lambda}, \bar{\mu})$ such that $(\bar{x}, \bar{\lambda}, \bar{\mu})$ verifies the first-order optimality system of $(\mathrm{P})_{\mathrm{I}(\bar{x})}$ by (i). We have to show that $\bar{\mu}_{I(\bar{x})} \geq 0$. With Lemma 2.3 (iv) and (H3), we deduce that $\{\lambda^k\}$ converges to $\bar{\lambda}$; hence, by (2) we have $\mu^{k+1} := -\nu_k X_k^{-2} d^k$ converges to $\bar{\mu}$. Let $i \in I(\bar{x})$ be such that $\bar{\mu}_i < 0$; then $d_i^k = -(x_i^k)^2 \mu_i^{k+1} / \nu_k > 0$ for k large enough, and this contradicts the fact that $x_i^k \to \bar{x}_i = 0$. □

3. Acceptance of the unit stepsize.

In this section we perform a local analysis around some point $\bar{x}$, local solution of (P). We seek conditions implying that if $\bar{x}$ is a limit point of $\{x^k\}$, the sequence $\{x^k\}$ converges to $\bar{x}$ and $\rho^k = 1$ is accepted. We note that the rate of convergence of the cost will not be better than linear, as this is the case in LP. Hence, the interest in obtaining a unit stepsize might be questionable. Our motivation is the following. We know that for QP problems, the solution can be computed with a good precision in a small number of iterates by using the exact Hessian for M_k (see [2] and [5]). Hence, we try to reproduce, for problems with a nonquadratic cost, this behavior. What we may prove, by a theoretical study, is that provided that M_k approximates the Hessian of the cost in a certain sense, the stepsize 1 is accepted; we then may hope that the contribution of the "nonquadratic part" of the cost is asymptotically negligible so that the rapid (although linear) convergence still occurs.

It might be argued that the need for M_k to be both positive semidefinite and an approximation of the Hessian in a certain sense makes the theory applicable only in the case of a convex f. This is not so. The situation is comparable to the one for sequential QP algorithms that use a positive definite approximation of the Hessian. The key property is that the Hessian of the cost is positive definite in the tangent space under some natural second-order assumptions, whereas the approximation in the normal space plays no role. This allows approximation in an effective way of a possibly undefinite Hessian by a positive semidefinite matrix.

We need a few definitions. Assuming that $\bar{x}$ satisfies (H3), it follows that $\bar{x}$ is associated with a unique pair $(\bar{\lambda}, \bar{\mu})$ such that (OS) holds. Define the set of strictly active constraints as

$$J(\bar{x}) := \{i \in \{1, \dots, n\} \; ; \; \bar{\mu}_i > 0\}$$

and the extended critical cone as

$$T := \{d \in \mathbb{R}^n \; ; \; Ad = 0 \; ; \; d_i = 0, \; i \in J(\bar{x})\}.$$

We say that $\bar{x}$ satisfies the strong second-order condition (see Robinson [23]) whenever

$$\text{(SSOC)} \qquad \exists \alpha_1 > 0 \; ; \; d^t \nabla^2 f(\bar{x}) d \geq \alpha_1 \|d\|^2 \; \forall \; d \in T.$$

This is a sufficient condition for the strong regularity, as defined in [23], of the associated optimality system. It has proven useful in sensitivity analysis as well as in the study of convergence properties of algorithms (see, e.g., [16], [4], and [3]).

Given d in $\mathcal{N}(A)$, the null space of A, we now define d_T, d_N as the orthogonal projection (in $\mathcal{N}(A)$) of d onto T and N, where N is the orthogonal complement of T in $\mathcal{N}(A)$, i.e.,

$$N = \{z \in \mathcal{N}(A) \; ; \; z^t d = 0 \; \forall \; d \in T\},$$

of course, $d = d_T + d_N$ and $\|d\|^2 = \|d_T\|^2 + \|d_N\|^2$. Similarly, we associate d^k with d_T^k and d_N^k. Last, but not least, we define the sufficient curvature condition as

$$\text{(SCC)} \qquad \left\{ \begin{array}{l} \exists \; \varepsilon_0 > 0, \text{ if } \|x^k - \bar{x}\| \leq \varepsilon_0 \text{ then} \\ \\ (d_T^k)^t M_k d_T^k \geq \dfrac{1}{2-\gamma} \, (d_T^k)^t \nabla^2 f(\bar{x}) d_T^k + \varepsilon_0 \|d_T^k\|^2. \end{array} \right.$$

We briefly discuss this condition. Specifically, we check that if M_k satisfies the inequality below and condition (SSOC) holds, then (SCC) is satisfied. We consider the following condition:

$$(7) \qquad (d_T^k)^t M_k d_T^k \geq (d_T^k)^t \nabla^2 f(\bar{x}) d_T^k + o(\|d_T^k\|^2).$$

To see that (7) implies (SCC), note that $1/(2-\gamma) \in (0, 1)$ and $(d_T^k)^t \nabla^2 f(\bar{x}) d_T^k \geq \alpha_1 \|d_T^k\|^2$ by (SSOC). This and (7) imply that

$$(d_T^k)^t M_k d_T^k \geq \frac{1}{2-\gamma} (d_T^k)^t \nabla^2 f(\bar{x}) d_T^k + \alpha_1 \left(1 - \frac{1}{2-\gamma}\right) \|d_T^k\|^2 + o(\|d_T^k\|^2),$$

from which (SCC) follows. In particular, (SCC) is satisfied if (SSOC) holds and $M_k = \nabla^2 f(x^k)$ (which, of course, is possible only if f is convex).

Condition (SCC) is similar to a condition recently used in the analysis of successive QP algorithms [3]. It is checked in [3] that in the case of unconstrained optimization (then actually d_T^k and d^k coincide), this condition is very weak in the following sense: assuming that the second-order sufficient optimality condition hold for $(\nabla^2 f(\bar{x}) > 0)$, a necessary condition for the acceptance of the unit step for x^k close to $\bar{x}$ is

$$(d_T^k)^t M_k d_T^k \geq \frac{1}{2-\gamma} (d_T^k)^t \nabla^2 f(\bar{x}) d_T^k + o(\|d_T^k\|^2).$$

THEOREM 3.1. *Assume that* $\{M_k\}$ *is bounded,* $\bar{x}$ *satisfies* (H3) *and* (SSOC), *and* (SCC) *is satisfied for* x^k *close enough to* $\bar{x}$*. Then, there exists* $\varepsilon > 0$; *if, for some* k_0, $\|x^{k_0} - \bar{x}\| < \varepsilon$ *then* $d^k \to 0$, $\rho^k = 1$ *for all* $k \geq k_0$*, and* $x^k \to \bar{x}$.

We need a few lemmas (Lemma 3.2 is stated in [3]; we give its proof for the reader's convenience).

LEMMA 3.2. *Given* $\varepsilon > 0$ *and an* $n \times n$ *symmetric matrix* M*, define*

$$K(\varepsilon, M) := \|M\|(1 + \|M\|/\varepsilon).$$

The two inequalities below then hold:

$$d_T^t M d_T \geq d^t M d - \varepsilon\|d_T\|^2 - K(\varepsilon, M)\|d_N\|^2, \tag{8}$$

$$d^t M d \geq d_T^t M d_T - \varepsilon\|d_T\|^2 - K(\varepsilon, M)\|d_N\|^2. \tag{9}$$

Proof. Since $d = d_T + d_N$, it follows that

$$d^t M d = d_T^t M d_T + 2 d_T^t M d_N + d_N^t M d_N.$$

Hence,

$$|d^t M d - d_T^t M d_T| = |2 d_T^t M d_N + d_N^t M d_N| \leq \|M\|(2\|d_T\|.\|d_N\| + \|d_N\|^2).$$

Using the inequality $2ab \leq a^2 + b^2$ with $a = \sqrt{\varepsilon}\|d_T\|$ and $b = \|M\|\|d_N\|/\sqrt{\varepsilon}$, we get

$$|d^t M d - d_T^t M d_T| \leq \varepsilon\|d_T\|^2 + \|M\|(1 + \|M\|/\varepsilon)\|d_N\|^2,$$

from which the conclusion follows. □

LEMMA 3.3. *There exists* $c_1 > 0$ *such that*

$$\|z_N\| \leq c_1 \sum_{i \in J(\bar{x})} |z_i| \ \forall\, z \in \ker A.$$

Proof. We have $z_N = z - z_T$ and $(z_T)_i = 0, i \in J(\bar{x})$. Henceforth, $z_i = (z_N)_i, i \in J(\bar{x})$, and it suffices to prove that

$$\|z\| \leq c_1 \sum_{i \in J(\bar{x})} |z_i| \ \forall\, z \in N.$$

Since both sides are positively homogeneous, it suffices to establish the inequality when $\|z\| = 1$. Then, the existence of c_1 amounts to saying that the problem

$$\min \sum_{i \in J(\bar{x})} |z_i| \ ; \ z \in N, \ \|z\| = 1$$

has a positive infimum. If this were not the case, there would exist $z \in N$, $\|z\| = 1$, with $z_i = 0$, $i \in J(\bar{x})$ because this problem has a solution by compactness arguments; hence, $z \in T$ (by definition of T), i.e., $z \in T \cap N = \{0\}$, a contradiction. □

LEMMA 3.4. *Assume that* $\{M_k\}$ *is bounded and* $\bar{x}$ *satisfies* (H3)*. Given* $K \geq 0$*, if* x^k *is sufficiently close to* $\bar{x}$*, the following relation holds:*

$$\nu_k (d^k)^t X_k^{-2} d^k > K\|d_N^k\|^2. \tag{10}$$

Proof. Denote $\mu^{k+1} := -\nu_k X_k^{-2} d^k$. From (H3) and Lemma 2.3 (iv), we deduce that for any subsequence of $\{x^k\}$ converging to $\bar{x}$, the associated subsequence λ^k converges. Combining this with (2), the boundedness of M_k, and Lemma 2.3 (iii), we deduce that the associated subsequence of $\{\mu^k\}$ converges to $\bar{\mu}$. Hence, if x^k is close enough to $\bar{x}$, one has $d_i^k < 0$ and $\mu_i^k > \bar{\mu}_i/2$, $i \in J(\bar{x})$. Denote

$$\theta := \min\{\bar{\mu}_i/2,\ i \in J(\bar{x})\}.$$

It follows that

$$\nu_k (d^k)^t X_k^{-2} d^k \geq \nu_k \sum_{i \in J(\bar{x})} (d_i^k / x_i^k)^2 \geq \frac{1}{2} \sum_{i \in J(\bar{x})} -\bar{\mu}_i d_i^k \geq \theta \sum_{i \in J(\bar{x})} |d_i^k|. \tag{11}$$

Also, since $|d_i^k| \leq |x_i^k|/\delta$, it follows that $|d_i^k|$, $i \in J(\bar{x})$ can be made arbitrarily small by taking x^k close to $\bar{x}$. It follows with (11) that

$$\nu_k (d^k)^t X_k^{-2} d^k \Big/ \left(\sum_{i \in J(\bar{x})} |d_i^k| \right)^2 \to \infty. \tag{12}$$

We conclude with Lemma 3.3. □

Lemma 3.5. *Let* $\alpha_1 > 0$ *be given by* (SSOC). *Given* $K > 0$, *under the hypotheses of Theorem* 3.1, *if* x^k *is sufficiently close to* $\bar{x}$ *then*

$$(d^k)^t (M_k + 2\nu_k X_k^{-2}) d^k \geq \frac{\alpha_1}{2} \|d^k\|^2 + K \|d_N^k\|^2. \tag{13}$$

Proof. Define

$$K(\varepsilon) := \sup_{k \in N} K(\varepsilon, M_k).$$

Because $\{M_k\}$ is bounded, we have that $K(\varepsilon) < \dot{\infty}$. Apply Lemma 3.2, with $\varepsilon = \varepsilon_0$, where $\varepsilon_0 > 0$ is such that (SCC) holds. We obtain that if x^k is close to $\bar{x}$, then

$$(d^k)^t M_k d^k \geq \frac{1}{2-\gamma} (d_T^k)^t \nabla^2 f(\bar{x}) d_T^k - K(\varepsilon_0) \|d_N^k\|^2.$$

Since $1/(2-\gamma) \geq 1/2$, by using (SSOC) we get

$$\begin{aligned} (d^k)^t M_k d^k &\geq \frac{\alpha_1}{2} \|d_T^k\|^2 - K(\varepsilon_0) \|d_N^k\|^2, \\ &= \frac{\alpha_1}{2} \|d^k\|^2 - \left(K(\varepsilon_0) + \frac{\alpha_1}{2} \right) \|d_N^k\|^2. \end{aligned}$$

The conclusion is obtained with Lemma 3.4. □

Proof of Theorem 3.1. (a) We first prove that $x^k \to \bar{x}$. We use the fact that $\|d^k\|$ is small whenever x^k is close to $\bar{x}$, k is large enough as a consequence of Lemma 3.5 and (5), and $\bar{x}$ satisfies (SSOC). The last fact implies that $\bar{x}$ is an isolated critical point of (P) (see [23]). As (H3) necessarily holds in a neighborhood of $\bar{x}$, it follows by Theorem 2.2 that $\bar{x}$ is the only limit point of $\{x^k\}$ in some neighborhood $\mathcal{V}$ of $\bar{x}$.

We now just have to prove that x^k remains in $\mathcal{V}$ for k large enough. We can take $\mathcal{V}$ of the form

$$\mathcal{V}_\varepsilon := \{x \in F \ ; \ \|x - \bar{x}\| \leq \varepsilon\}.$$

Note that $\|d^k\| < \varepsilon/2$ whenever $x^k \in \mathcal{V}_{\varepsilon_1}$ for some $\varepsilon_1 > 0$ small enough. We may assume that $\varepsilon_1 < \varepsilon/2$. It follows that if $x^k \in \mathcal{V}_{\varepsilon_1}$, then $\|x^{k+1} - \bar{x}\| \leq \|x^k - \bar{x}\| + \|d^k\| \leq \varepsilon$. In other words, x^{k+1} is in $\mathcal{V}_\varepsilon$ whenever x^k is in $\mathcal{V}_{\varepsilon_1}$.

On the other hand, we also know that $f(x^{k+1}) \leq f(x^k)$. So, let us define

$$\hat{f} := \inf\{f(x) \ ; \ x \in \mathcal{V}_\varepsilon - \mathcal{V}_{\varepsilon_1}\}.$$

Because $\bar{x}$ is a strict local minimum of (P), we may assume that $\hat{f} > f(\bar{x})$, reducing ε and ε_1 if necessary. Now, assuming that $f(x^k) \leq \hat{f}$ and $x^k \in \mathcal{V}_{\varepsilon_1}$, it follows that $f(x^{k+1}) < \hat{f}$ and $x^{k+1} \in \mathcal{V}_\varepsilon$; using the definition of $\hat{f}$, we find that x^{k+1} is in $\mathcal{V}_{\varepsilon_1}$ again. This implies that the sequence $\{x^k\}$ remains in $\mathcal{V}_{\varepsilon_1}$, hence, that $x^k \to \bar{x}$.

(b) We now check that $\rho^k = 1$ for k large enough. Define

$$H_k := 2\int_0^1 (1-\sigma)\nabla^2 f(x^k + \sigma d^k) d\sigma.$$

Then,

$$f(x^k) - f(x^k + d^k) = -\nabla f(x^k)^t d^k - \frac{1}{2}(d^k)^t H_k d^k.$$

If x^k is close enough to $\bar{x}$, d^k is then close to 0 as was already observed; hence, H_k is close to $\nabla^2 f(\bar{x})$. We deduce that

$$-(d^k)^t H_k d^k \geq -(d^k)^t \nabla^2 f(\bar{x}) d^k - \frac{\varepsilon_0}{2}\|d^k\|^2,$$

with ε_0 given by (SCC). As a consequence,

$$\begin{aligned} f(x^k) - f(x^k + d^k) &\geq -\nabla f(x^k)^t d^k - \frac{1}{2}(d^k)^t \nabla^2 f(\bar{x}) d^k - \frac{\varepsilon_0}{4}\|d^k\|^2, \\ &= f(x^k) - \varphi_k(d^k) + \frac{1}{2}(d^k)^t (M_k - \nabla^2 f(\bar{x})) d^k - \frac{\varepsilon_0}{4}\|d^k\|^2. \end{aligned}$$

So, by (1), the unit step will be accepted if

$$(14) \qquad (1-\gamma)(f(x^k) - \varphi_k(d^k)) + \frac{1}{2}(d^k)^t (M_k - \nabla^2 f(\bar{x})) d^k - \frac{\varepsilon_0}{4}\|d^k\|^2 \geq 0.$$

Using (5), Lemma 3.2 with $\varepsilon = \varepsilon_0/2$ (where ε_0 is given by (SCC)), (SCC), and Lemma 3.4, we get

$$\begin{aligned} f(x^k) - \varphi_k(d^k) &= \frac{1}{2}(d^k)^t M_k d^k + \nu_k \delta_k^2, \\ &\geq \frac{1}{2}(d_T^k)^t M_k d_T^k - \frac{\varepsilon_0}{4}\|d_T^k\|^2 - \frac{K(\varepsilon_0)}{2}\|d_N^k\|^2 + \nu_k \delta_k^2, \\ &\geq \frac{1}{2(2-\gamma)}(d_T^k)^t \nabla^2 f(\bar{x}) d_T^k - \frac{K(\varepsilon_0)}{2}\|d_N^k\|^2 + \frac{\varepsilon_0}{4}\|d_T^k\|^2 + \nu_k \delta_k^2, \\ &\geq \frac{1}{2(2-\gamma)}(d_T^k)^t \nabla^2 f(\bar{x}) d_T^k + \frac{\nu_k}{2}\delta_k^2. \end{aligned}$$

Similarly, by defining $K'(\varepsilon) := \sup_{k\in\mathbb{N}} K(\varepsilon, M_k - \nabla^2 f(\bar{x}))$, we obtain

$$\begin{aligned}\frac{1}{2}(d^k)^t(M_k - \nabla^2 f(\bar{x}))d^k &\geq \frac{1}{2}(d_T^k)^t(M_k - \nabla^2 f(\bar{x}))d_T^k - \frac{\varepsilon_0}{4}\|d_T^k\|^2 - K'(\varepsilon_0)\|d_N\|^2,\\ &\geq \frac{\gamma-1}{2(2-\gamma)}(d_T^k)^t\nabla^2 f(\bar{x})d_T^k - K'(\varepsilon_0)\|d_N^k\|^2 + \frac{\varepsilon_0}{4}\|d_T^k\|^2.\end{aligned}$$

Combining these inequalities and using Lemma 3.4 again, we get

$$\begin{aligned}(1-\gamma)(f(x^k) - \varphi_k(d^k)) + \frac{1}{2}(d^k)^t(M_k - \nabla^2 f(\bar{x}))d^k - \frac{\varepsilon_0}{4}\|d^k\|^2&\\ \geq (1-\gamma)\frac{\nu_k}{2}\delta_k^2 - (K'(\varepsilon_0) + \frac{\varepsilon_0}{4})\|d_N^k\|^2 \geq 0.&\end{aligned}$$

We have proven (14) as required. It follows that the unit step is accepted, hence, d^k vanishes, as was to be proved. □

We now check that if M_k is close to $\nabla^2 f(\bar{x})$ in a very weak sense (see (16) below), then the following holds:

$$\sum_k (\|x^k - \bar{x}\| + \|\lambda^k - \bar{\lambda}\| + \|\mu^k - \bar{\mu}\|) < \infty. \tag{15}$$

THEOREM 3.6. *Assume that the hypotheses of Theorem* 3.1 *hold and, in addition, that $\bar{x}$ satisfies the strict complementarity condition. If $x^k \to \bar{x}$ (hence, $\rho^k = 1$ by Theorem* 3.1*) then there exists $\varepsilon_1 > 0$ such that*

$$\|(M_k - \nabla^2 f(\bar{x}))d_T^k\| \leq \varepsilon_1 \|d^k\| \tag{16}$$

implies (15)*.*

Let us note that Newton's method satisfies (16). Note that if we assume $M_k \longrightarrow \nabla^2 f(\bar{x})$, then we may violate the positive definiteness requirement on M_k since $\nabla^2 f(\bar{x})$ need not be positive definite.

Proof. Denote

$$I := I(\bar{x}),\ \bar{I} := \{1, \ldots, n\} - I.$$

The proof is based on the mapping

$$\psi(x, \lambda) := \begin{cases} (\nabla f(x) + A^t\lambda)_{\bar{I}},\\ Ax - b,\\ x_I.\end{cases}$$

It follows easily from (SSOC) and (H3) that $\psi(x,\lambda)$ has an invertible derivative at $(\bar{x}, \bar{\lambda})$; hence, there exists some $a_1 > 0$ such that

$$\|x^{k+1} - \bar{x}\| + \|\lambda^{k+1} - \bar{\lambda}\| \leq a_1\|\psi(x^{k+1}, \lambda^{k+1})\|. \tag{17}$$

(a) Let us prove that

$$\exists\ K_1, K_4\ ; \|\psi(x^{k+1}, \lambda^{k+1})\| \leq nK_1\nu_{k+1} + nK_4\nu_k + \|x^{k+1} - x^k\|/(4a_1). \tag{18}$$

Indeed, from the convergence of $\{x^k\}$ to $\bar{x}$ and (H3) and by using Lemma 2.3 (iv) and (2), it follows that $(\lambda^k, \mu^k) \to (\bar{\lambda}, \bar{\mu})$. Now, multiplying (2) by X_k and recalling that $\nu_k \|X_k^{-1} d^k\| = \nu_k \delta_k$, we get

$$\|X_k(\nabla f(x^k) + M_k d^k + A^t \lambda^{k+1})\| = \nu_k \delta_k.$$

Using the strict complementarity hypothesis and the relation $|z_i| \leq \|z\|$, we obtain, for some $K_1 > 0$,

$$x_i^k \leq K_1 \nu_k, \quad i \in I, \tag{19}$$

$$|(\nabla f(x^k) + M_k d^k + A^t \lambda^{k+1})_i| \leq K_1 \nu_k, \quad i \notin I. \tag{20}$$

Now choose ε_1 in (16) as $\varepsilon_1 = 1/(8a_1 n)$.

We have

$$\nabla f(x^k) + M_k d^k = \nabla f(x^k) + \nabla^2 f(\bar{x}) d^k + (M_k - \nabla^2 f(\bar{x})) d^k,$$

$$= \nabla f(x^{k+1}) + r^k + (M_k - \nabla^2 f(\bar{x})) d^k, \tag{21}$$

where the term r^k for x^k close to $\bar{x}$ satisfies

$$\|r^k\| \leq \|d^k\|/(8a_1 n). \tag{22}$$

Also, by (16) and as $\{M_k\}$ is bounded, we get for some $K_2 > 0$

$$\|(M_k - \nabla^2 f(\bar{x})) d^k\| \leq \|(M_k - \nabla^2 f(\bar{x})) d_T^k\| + K_2 \|d_N^k\|,$$

$$\leq \|d^k\|/(8a_1 n) + K_2 \|d_N^k\|. \tag{23}$$

Using (21), (22), (23), and Lemma 3.3, we obtain for some $K_3 > 0$

$$\|\nabla f(x^k) + M_k d^k - \nabla f(x^{k+1})\| \leq \|d^k\|/(4a_1 n) + K_3 \sum_{j \in I} |d_j^k|. \tag{24}$$

Now we prove (18). As $\bar{\mu} = -\lim_{k \to +\infty} \nu_k X_k^{-2} d^k$, using the strict complementarity hypothesis for k large enough and for all $j \in I$, we get $d_j^k < 0$, hence, $|d_j^k| \leq x_j^k$. So, combining this with (19), (20), and (24), we get for some $K_4 > 0$

$$|(\nabla f(x^{k+1}) + A^t \lambda^{k+1})_i| \leq K_4 \nu_k + \|d^k\|/(4a_1 n), \quad i \notin I. \tag{25}$$

So, by (19) and (25), we get (18).

(b) On the other hand, by (5), the linesearch rule, and the fact that $\rho^k = 1$, we have

$$f(x^k) - f(x^{k+1}) \geq \gamma(f(x^k) - \varphi_k(d^k)) \geq \gamma \nu_k \delta_k^2.$$

Hence, as $\delta_k \geq \delta > 0$,

$$\nu := \sum_{k=1}^{\infty} \nu_k < \infty. \tag{26}$$

Hence, using (17) and (18), we get

$$\sum_{k=k_0}^{\bar{k}} (\|x^{k+1} - \bar{x}\| + \|\lambda^{k+1} - \bar{\lambda}\|) \leq a_1 n(K_1 + K_4)\nu + \frac{1}{4} \sum_{k=k_0}^{\bar{k}} \|x^{k+1} - x^k\|.$$

Now, by using

$$\frac{1}{4} \sum_{k=k_0}^{\bar{k}} \|x^{k+1} - x^k\| \leq \frac{1}{4} \sum_{k=k_0}^{\bar{k}} (\|x^{k+1} - \bar{x}\| + \|x^k - \bar{x}\|) \leq \frac{1}{2} \sum_{k=k_0}^{\bar{k}} \|x^{k+1} - \bar{x}\| + \frac{1}{4} \|x^{k_0} - \bar{x}\|,$$

we deduce that

$$\sum_{k=k_0}^{\bar{k}} (\|x^{k+1} - \bar{x}\| + \|\lambda^{k+1} - \bar{\lambda}\|) \leq 2a_1 n(K_1 + K_4)\nu + \frac{1}{2} \|x^{k_0} - \bar{x}\|.$$

Finally, we obtain (15), noticing that by (2)

$$\begin{aligned} \mu^{k+1} - \bar{\mu} &= O(\|x^k - \bar{x}\| + \|\lambda^{k+1} - \bar{\lambda}\| + \|d^k\|), \\ &= O(\|x^{k+1} - \bar{x}\| + \|x^k - \bar{x}\| + \|\lambda^{k+1} - \bar{\lambda}\|). \qquad \square \end{aligned}$$

Acknowledgment. The authors would like to thank the referees for their useful remarks.

REFERENCES

[1] L. Armijo, *Minimization of functions having Lipschitz continuous first partial derivatives*, Pacific J. Math., 16 (1966), pp. 1–3.

[2] J. F. Bonnans and M. Bouhtou, *The trust region affine interior point algorithm for nonconvex quadratic programming*, RAIRO Rech. Opér., 29 (1995), pp. 195–217.

[3] J. F. Bonnans and G. Launay, *An implicit trust region algorithm for constrained optimization*, SIAM J. Optim., 5 (1995), pp. 792–812.

[4] J. F. Bonnans and A. Sulem, *Pseudopower expansion of generalized equations and constrained optimization problems*, Math. Programming, 70 (1995), pp. 123–148.

[5] M. Bouhtou, *Méthodes de points intérieurs pour l'optimisation des systèmes de grande taille*, Ph.D. dissertation, Université de Paris-IX Dauphine, Paris, France, 1993.

[6] J. P. Bulteau and J. P. Vial, *A restricted trust region algorithm for unconstrained optimization*, J. Optim. Theory Appl., 47 (1985), pp. 413–435.

[7] T. F. Coleman and Y. Li, *An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds*, Technical report TR93-1342, Computer Science Department, Cornell University, Ithaca, NY, 1993.

[8] D. Den Hertog, C. Roos, and T. Terlaky, *On the classical logarithmic barrier function method for a class of smooth convex programming problems*, J. Optim. Theory Appl., 73 (1992), pp. 1–25.

[9] J. E. Dennis, M. Heinkenschloss, and L. N. Vicente, *Trust-Region Interior Point Algorithms for a Class of Nonlinear Programming Problems*, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1994.

[10] I. I. Dikin, *Iterative solutions of problems of linear and quadratic programming*, Soviet Math. Dokl., 8 (1967), pp. 674–675.

[11] I. I. Dikin and V. I. Zorkalcev, *Iterative Solutions of Mathematical Programming Problems (Algorithms of Interior Point Methods)*, Nauka Publishers, Novosibirsk, 1980 (in Russian).

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

[13] C. C. Gonzaga and L. A. Carlos, *A primal affine-scaling algorithm for linearly constrained convex programs*, Technical report ES-238/90, Universidade Federal do Rio de Janeiro, Brazil, 1990.

[14] C. C. Gonzaga, *An interior trust region method for linearly constrained optimization*, COAL Bull., 19 (1991), pp. 55–65.

[15] F. Jarre, *On the method of analytic centers for solving smooth convex programs*, in Lecture Notes in Mathematics 1405, S. Dolecki, ed., Springer-Verlag, Berlin, New York, 1989, pp. 69–85.

[16] K. Jittorntrum, *Solution point differentiability without strict complementarity in nonlinear programming*, Math. Programming Study, 21 (1984), pp. 127–138.

[17] N. Karmarkar, *A new polynomial time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[18] G. McCormick, *The projective SUMT method for convex programming problems*, Math. Oper. Res., 14 (1989), pp. 203–223.

[19] S. Mehrotra and J. Sun, *An interior point algorithm for solving smooth convex programs based on Newton's method*, in Mathematical Developments Arising from Linear Programming, Contemporary Mathematics 114, J. C. Lagarias and M. J. Todd, eds., AMS, Providence, RI, 1990, pp. 265–284.

[20] R. D. C. Monteiro and I. Adler, *An extension of Karmarkar–type algorithms to a class of convex separable programming problems with global rate of convergence*, Math. Oper. Res., 15 (1990), pp. 408–422.

[21] J. J. Moré, *Recent developments in algorithms and software for trust region method*, in Mathematical Programming, the State of the Art, A. Bachem, M. Grötschel, and B. Korte, eds., Springer-Verlag, Berlin, New York, 1983, pp. 258–287.

[22] M. J. D. Powell, *Algorithms for nonlinear constraints that use Lagrangian functions*, Math. Programming, 15 (1978), pp. 224–248.

[23] S. M. Robinson, *Strongly regular generalized equations*, Math. Oper. Res., 5 (1980), pp. 43–62.

[24] D. C. Sorensen, *Newton's method with a model trust region modification*, SIAM J. Numer. Anal., 19 (1982), pp. 409–426.

[25] J. Sun, *A convergence proof for an affine-scaling algorithm for convex quadratic programming without nondegeneracy assumptions*, Math. Programming, 60 (1993), pp. 69–79.

[26] T. Tsuchiya, *Global Convergence of the Affine Scaling Algorithm for Primal Degenerate Strictly Convex Quadratic Programming Problems*, Math. Oper. Res., 47 (1993), pp. 509–539.

[27] Y. Ye and E. Tse, *An extension of Karmarkar's projective algorithm for convex quadratic programming*, Math. Programming, 44 (1989), pp. 157–179.

SIAM J. OPTIM.
Vol. 7, No. 3, pp. 732–756, August 1997

TENSOR METHODS FOR LARGE, SPARSE UNCONSTRAINED OPTIMIZATION*

ALI BOUARICHA†

Abstract. Tensor methods for unconstrained optimization were first introduced by Schnabel and Chow [*SIAM J. Optim.*, 1 (1991), pp. 293–315], who described these methods for small- to moderate-sized problems. The major contribution of this paper is the extension of these methods to large, sparse unconstrained optimization problems. This extension requires an entirely new way of solving the tensor model that makes the methods suitable for solving large, sparse optimization problems efficiently. We present test results for sets of problems where the Hessian at the minimizer is nonsingular and where it is singular. These results show that tensor methods are significantly more efficient and more reliable than standard methods based on Newton's method.

Key words. tensor methods, unconstrained optimization, sparse problems, large-scale optimization, singular problems

AMS subject classification. 65K

PII. S1052623494267723

1. Introduction.

In this paper we describe tensor methods for solving the unconstrained optimization problem

$$\text{given } f : \Re^n \to \Re, \text{ find } x_* \in \Re^n \text{ such that } f(x_*) \leq f(x) \text{ for all } x \in D, \tag{1.1}$$

where D is some open set containing x_*, and f is convex on D. We assume that f is at least twice continuously differentiable and n is large.

Tensor methods for unconstrained optimization are general-purpose methods primarily intended to improve upon the performance of standard methods, especially on problems where $\nabla^2 f(x_*)$ has a small rank deficiency. They are also intended to be at least as efficient as standard methods on problems where $\nabla^2 f(x_*)$ is nonsingular.

Tensor methods for unconstrained optimization base each iteration upon the fourth-order model of the objective function $f(x)$,

$$M_T(x_c + d) = f(x_c) + \nabla f(x_c) \cdot d + \frac{1}{2}\nabla^2 f(x_c) \cdot d^2 + \frac{1}{6}T_c \cdot d^3 + \frac{1}{24}V_c \cdot d^4, \tag{1.2}$$

where $d \in \Re^n$, x_c is the current iterate, $\nabla f(x_c)$ and $\nabla^2 f(x_c)$ are the first and second analytic derivatives of f at x_c, or finite-difference approximations to them, and the tensor terms at x_c, $T_c \in \Re^{n\times n\times n}$, and $V_c \in \Re^{n\times n\times n\times n}$ are symmetric. (We use the notation $\nabla f(x_c) \cdot d$ for $\nabla f(x_c)^T d$, and $\nabla^2 f(x_c) \cdot d^2$ for $d^T \nabla^2 f(x_c) d$ to be consistent with the tensor notation $T_c \cdot d^3$ and $V_c \cdot d^4$. Also, for simplicity, we abbreviate terms of the form dd, ddd, and $dddd$ by d^2, d^3, and d^4, respectively.) Before proceeding, we define the tensor notation used above.

[*] Received by the editors May 13, 1994; accepted for publication (in revised form) March 15, 1996. Part of this work was performed while the author was a research associate at CERFACS (Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique). This work was supported in part by the Office of Scientific Computing, U. S. Department of Energy, under contract W-31-109-Eng-38.

http://www.siam.org/journals/siopt/7-3/26772.html

[†] Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439 (bouarich@mcs.anl.gov).

Definition 1.1. *Let $T \in \Re^{n \times n \times n}$. Then for $u, v, w \in \Re^n$, $T \cdot uvw \in \Re$, $T \cdot vw \in \Re^n$, with*

$$T \cdot uvw = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} T(i,j,k)u(i)v(j)w(k),$$

$$(T \cdot vw)(i) = \sum_{j=1}^{n} \sum_{k=1}^{n} T(i,j,k)v(j)w(k), \quad i = 1, \ldots, n.$$

Definition 1.2. *Let $V \in \Re^{n \times n \times n \times n}$. Then for $r, u, v, w \in \Re^n, V \cdot ruvw \in \Re, V \cdot uvw \in \Re^n$ with*

$$V \cdot ruvw = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} V(i,j,k,l)r(i)u(j)v(k)w(l),$$

$$(V \cdot uvw)(i) = \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} V(i,j,k,l)u(j)v(k)w(l), \quad i = 1, \ldots, n.$$

The tensor terms are selected so that the model interpolates a small number of function and gradient values from previous iterations. This results in T_c and V_c being low-rank tensors, which is crucial for the efficiency of the tensor method. The tensor method requires no more function or derivative evaluations per iteration and hardly more storage or arithmetic operations than does a standard method based on Newton's method.

Standard methods for solving unconstrained optimization problems are widely described in the literature; general references on this topic include Dennis and Schnabel [9], Fletcher [12], and Gill, Murray, and Wright [14]. In this paper, we propose extensions to standard methods that use analytic or finite-difference gradients and Hessians.

The standard method for unconstrained optimization, Newton's method, bases each iteration upon the quadratic model of $f(x)$,

$$M_N(x_c + d) \ = \ f(x_c) \ + \ \nabla f(x_c) \cdot d + \frac{1}{2} \nabla^2 f(x_c) \cdot d^2. \tag{1.3}$$

This method is defined when $\nabla^2 f(x_c)$ is nonsingular and consists of setting the next iterate x_+ to the minimizer of (1.3), namely,

$$x_+ = x_c - \nabla^2 f(x_c)^{-1} \nabla f(x_c). \tag{1.4}$$

A distinguishing feature of Newton's method is that if $\nabla^2 f(x_c)$ is nonsingular at a local minimizer x_*, then the sequence of iterates produced by (1.4) converges locally quadratically to x_*. However, Newton's method is generally linearly convergent at best if $\nabla^2 f(x_*)$ is singular [15].

Methods based on (1.2) have been shown to be more reliable and more efficient than standard methods on small- to moderate-sized problems [19]. In the test results obtained for both nonsingular and singular problems, the improvement by the tensor method over Newton's method is substantial, ranging from 30% to 50% in iterations and in function and derivative evaluations. Furthermore, the tensor method solves several problems that Newton's method fails to solve.

The tensor algorithms described in [19] are QR-based algorithms involving orthogonal transformations of the variable space. These algorithms are very effective for minimizing the tensor model when the Hessian is dense because they are very stable numerically, especially when the Hessian is singular. They are not efficient for sparse problems, however, because they destroy the sparsity of the Hessian due to the orthogonal transformation of the variable space. To preserve the sparsity of the Hessian, we have developed an entirely new way of solving the tensor model that employs a sparse variant of the Cholesky decomposition. This makes our new algorithms very well suited for sparse problems.

The remainder of this paper is organized as follows. In section 2 we briefly review the techniques introduced by Schnabel and Chow [19] to form the tensor model. In section 3 we describe efficient algorithms for minimizing the tensor model when the Hessian is sparse. In sections 4 and 5 we discuss the globally convergent modifications for tensor methods for large, sparse unconstrained optimization. These consist of line search backtracking and model trust region techniques. A high-level implementation of the tensor method is given in section 6. In section 7 we describe comparative testing for an implementation based on the tensor method versus an implementation based on Newton's method, and we present summary statistics of the test results. Finally, in section 8, we give a summary of our work and a discussion of future research.

2. Forming the tensor model.

In this section, we briefly review the techniques that were introduced in [19] for forming the tensor model for unconstrained optimization.

As was stated in the preceding section, the tensor method for unconstrained optimization bases each iteration upon the fourth-order model of the nonlinear function $f(x)$ given by (1.2).

The choices of T_c and V_c in (1.2) cause the third-order term $T_c \cdot d^3$ and the fourth-order term $V_c \cdot d^4$ to have simple and useful forms. These tensor terms are selected so that the tensor model interpolates function and gradient information at a set of p not necessarily consecutive past iterates $x_{-1}, \ldots, x_{-p}$.

In the remainder of this paper, we restrict our attention to $p = 1$. The reasons for this choice are that the performance of the tensor version that allows $p \geq 1$ is similar overall to that constraining p to be 1, and that the method is simpler and less expensive to implement in this case. (The derivation of the third- and fourth-order tensor terms for $p \geq 1$ is explained in detail in [19].)

The interpolation conditions at the past point x_{-1} are given by

$$f(x_{-1}) \;=\; f(x_c) \;+\; \nabla f(x_c) \cdot s \;+\; \frac{1}{2}\nabla^2 f(x_c) \cdot s^2 + \frac{1}{6}T_c \cdot s^3 + \frac{1}{24}V_c \cdot s^4 \tag{2.1}$$

and

$$\nabla f(x_{-1}) \;=\; \nabla f(x_c) \;+\; \nabla^2 f(x_c) \cdot s \;+\; \frac{1}{2}T_c \cdot s^2 \;+\; \frac{1}{6}V_c \cdot s^3, \tag{2.2}$$

where

$$s \;=\; x_{-1} - x_c.$$

Schnabel and Chow [19] choose T_c and V_c to satisfy (2.1) and (2.2). They first show that the interpolation conditions (2.1) and (2.2) uniquely determine $T_c \cdot s^3$ and $V_c \cdot s^4$. Multiplying (2.2) by s yields

$$\nabla f(x_{-1}) \cdot s \;=\; \nabla f(x_c) \cdot s \;+\; \nabla^2 f(x_c) \cdot s^2 \;+\; \frac{1}{2} T_c \cdot s^3 \;+\; \frac{1}{6} V_c \cdot s^4. \tag{2.3}$$

Let α, $\beta \in \Re$ be defined by

$$\alpha = T_c \cdot s^3,$$

$$\beta = V_c \cdot s^4.$$

Then from (2.1) and (2.3) they obtain the following system of two linear equations in the two unknowns α and β:

$$\frac{1}{2}\alpha \;+\; \frac{1}{6}\beta \;=\; q_1, \tag{2.4}$$

$$\frac{1}{6}\alpha \;+\; \frac{1}{24}\beta \;=\; q_2, \tag{2.5}$$

where q_1, $q_2 \in \Re$ are defined by

$$q_1 \;=\; \nabla f(x_{-1}) \cdot s \;-\; \nabla f(x_c) \cdot s \;-\; \nabla^2 f(x_c) \cdot s^2,$$

$$q_2 \;=\; f(x_{-1}) \;-\; f(x_c) \;-\; \nabla f(x_c) \cdot s \;-\; \frac{1}{2} \nabla^2 f(x_c) \cdot s^2.$$

The system (2.4)–(2.5) is nonsingular; therefore, the values of α and β are uniquely determined. Hence, the interpolation conditions uniquely determine $T_c \cdot s^3$ and $V_c \cdot s^4$. Since these are the only interpolation conditions, the choice of T_c and V_c is vastly underdetermined.

Schnabel and Chow [19] choose T_c and V_c by first selecting the smallest symmetric V_c, in the Frobenius norm, for which

$$V_c \cdot s^4 \;=\; \beta,$$

where β is determined by (2.4)–(2.5). Then they substitute this value of V_c into (2.2), obtaining

$$T_c \cdot s^2 = a, \tag{2.6}$$

where

$$a \;=\; 2\left(\nabla f(x_{-1}) \;-\; \nabla f(x_c) \;-\; \nabla^2 f(x_c) \cdot s \;-\; \frac{1}{6} V_c \cdot s^3 \right). \tag{2.7}$$

This is a set of n linear equations in n^3 unknowns $T_c(i, j, k)$, $1 \leq i, j, k \leq n$. More precisely, Schnabel and Chow [19] choose the smallest symmetric T_c and V_c, in the Frobenius norm, that satisfy the equations (2.6)–(2.7). That is,

$$\min_{V_c \in \Re^{n\times n\times n\times n}} \; \| V_c \|_F \tag{2.8}$$

$$\text{subject to } V_c \cdot s^4 \;=\; \beta, \text{ and } V_c \text{ is symmetric,}$$

and

$$\min_{T_c \in \Re^{n\times n\times n}} \; \| T_c \|_F \tag{2.9}$$

$$\text{subject to } T_c \cdot s^2 \;=\; a, \text{ and } T_c \text{ is symmetric.}$$

The solution to (2.8) is

$$V_c \;=\; \gamma\,(s \otimes s \otimes s \otimes s), \quad \gamma = \frac{\beta}{(s^T s)^4},$$

where the tensor $V_c = s \otimes s \otimes s \otimes s \in \Re^{n\times n\times n\times n}$ is called a fourth-order rank-one tensor for which $V_c(i,j,k,l) = s(i)s(j)s(k)s(l),\ 1 \le i,j,k,l \le n$. (We use the notation $\otimes$ to be consistent with [19].)

The solution to (2.9) is

$$T_c \;=\; b \otimes s \otimes s + s \otimes b \otimes s + s \otimes s \otimes b, \tag{2.10}$$

where the notation $T = u \otimes v \otimes w,\ u,v,w \in \Re^n,\ T \in \Re^{n\times n\times n}$, is called a third-order rank-one tensor for which $T(i,j,k) = u(i)v(j)w(k)$. Here $b \in \Re^n$ is the unique vector for which (2.10) satisfies (2.6). It is given by

$$b \;=\; \frac{3a(s^T s) \;-\; 2s(s^T a)}{3(s^T s)^3}.$$

T_c and V_c determined by the minimum norm problems (2.9) and (2.8) have rank 2 and 1, respectively. This is the key to forming, storing, and solving the tensor model efficiently. The whole process of forming the tensor model requires only $O(n^2)$ arithmetic operations. The storage needed for forming and storing the tensor model is only a total of $6n$.

For further information, we refer to [19].

3. Solving the tensor model when the Hessian is sparse.

In this section we give algorithms for finding a minimizer of the tensor model (1.2) efficiently when the Hessian is sparse.

The substitution of the values of T_c and V_c into (1.2) results in the tensor model

$$\begin{aligned} M_T(x_c + d) \;&=\; f(x_c) + \nabla f(x_c) \cdot d + \frac{1}{2}\nabla^2 f(x_c) \cdot d^2 \\ &\quad + \frac{1}{2}(b^T d)(s^T d)^2 + \frac{\gamma}{24}(s^T d)^4. \end{aligned} \tag{3.1}$$

As we stated in section 2, we only consider the case $p = 1$ where the tensor model interpolates $f(x)$ and $\nabla f(x)$ at the previous iterate. The generalization for $p \ge 1$ is fairly straightforward. This constraint is mainly motivated by our computational results. When we allow $p \ge 1$, our test results show almost no improvement over the case where $p = 1$. The tensor method is therefore considerably simpler, as well as cheaper in terms of storage and cost per iteration.

3.1. Case 1: The Hessian is nonsingular. We show that the minimization of (3.1) can be reduced to the solution of a third-order polynomial in one unknown, plus the solution of three systems of linear equations that all involve the same coefficient matrix $\nabla^2 f(x_c)$. For conciseness, we use the notation $g = \nabla f(x_c)$ and $H = \nabla^2 f(x_c)$.

A necessary condition for d to be a local minimizer of (3.1) is that the derivative of the tensor model with respect to d must be zero. That is,

$$\nabla M_T(x_c + d) = g + Hd + (b^T d)(s^T d)s + \frac{1}{2}(s^T d)^2 b + \frac{\gamma}{6}(s^T d)^3 s = 0,$$

which yields

$$d = -H^{-1}\left(g + (b^T d)(s^T d)s + \frac{1}{2}(s^T d)^2 b + \frac{\gamma}{6}(s^T d)^3 s\right). \tag{3.2}$$

If we first premultiply the equation (3.2) by s^T on both sides, we obtain a cubic equation in the unknowns $\beta = s^T d$ and $\theta = b^T d$,

$$s^T H^{-1} g + \beta + s^T H^{-1} s\theta\beta + \frac{1}{2} s^T H^{-1} b\beta^2 + \frac{\gamma}{6} s^T H^{-1} s\beta^3 = 0. \tag{3.3}$$

If we then premultiply the equation (3.2) by b^T on both sides, we obtain another cubic equation in the unknowns β and θ,

$$b^T H^{-1} g + \theta + b^T H^{-1} s\theta\beta + \frac{1}{2} b^T H^{-1} b\beta^2 + \frac{\gamma}{6} b^T H^{-1} s\beta^3 = 0. \tag{3.4}$$

Thus, we obtain a system of two cubic equations in the two unknowns β and θ which can be solved analytically.

We now show how to compute the solutions of this system of two cubic equations in two unknowns by computing the solutions of a single cubic equation in the unknown β. Let $u = s^T H^{-1} g$, $v = s^T H^{-1} b$, $w = s^T H^{-1} s$, $y = b^T H^{-1} g$, and $z = b^T H^{-1} b$. We first calculate the value of θ as a function of β using the equation (3.3):

$$\theta = -\frac{\left(u + \beta + \frac{1}{2} v\beta^2 + \frac{\gamma}{6} w\beta^3\right)}{w\beta}. \tag{3.5}$$

Note that the denominator of (3.5) is equal to zero if either $\beta = 0$ or $w = 0$. We assume that $\beta \neq 0$; otherwise the tensor model would be reduced to the Newton model. Now, if $w = 0$, then (3.3) would be quadratic in β and

$$\beta = \frac{-1 \pm \sqrt{1 - 2uv}}{2}.$$

In this case, real-valued minimizers of the tensor model (3.1) may exist only if $1 - 2uv \geq 0$. It is easy to check that in order for θ to have a defined value, $1 + v\beta$ cannot be zero.

If $\beta \neq 0$ and $w \neq 0$, we substitute the expression for θ into (3.4) and obtain

$$-u + (yw - uv - 1)\beta - \frac{3}{2} v\beta^2 + \left(\frac{1}{2} wz - \frac{\gamma}{6} w - \frac{1}{2} v^2\right)\beta^3 = 0, \tag{3.6}$$

which is a third-order polynomial in the one unknown β. The roots of (3.6) are computed analytically. We substitute the values of β into (3.5) to calculate the values of θ. Then we simply substitute the values of β and θ into (3.2) to obtain the values of d. The major cost in this whole process is the calculation of $H^{-1}g$, $H^{-1}b$, and $H^{-1}s$.

After we compute the values of d, we determine which of them are potential minimizers. Our criterion is to select those values of d that guarantee that there is a descent path from x_c to $x_c + d$ for the model $M_T(x_c + d)$. Then, among the selected steps, we choose the one that is closest to the current iterate x_c in the Euclidean norm sense. If the tensor model has no minimizer, we use the standard Newton step as the step direction for the current iteration.

3.2. Case 2: The Hessian is rank deficient. If the Hessian matrix is rank deficient, we transform the tensor model given in (3.1) by the following procedure. Let $d = \hat{d} + \delta$ for a fixed $\hat{d}$, where δ is the new unknown. Substituting this expression for d into (3.1) yields the following tensor model, which is a function of δ:

$$
\begin{aligned}
M_T(x_c + d) \;=\; & f(x_c) + \nabla f(x_c) \cdot \hat{d} + \frac{1}{2}\nabla^2 f(x_c) \cdot \hat{d}^2 + \frac{1}{2}(b^T\hat{d})(s^T\hat{d})^2 \\
& + \frac{\gamma}{24}(s^T\hat{d})^4 + \Big(\nabla f(x_c) + \nabla^2 f(x_c)\hat{d} + (b^T\hat{d})(s^T\hat{d})s \\
& + \frac{1}{2}(s^T\hat{d})^2 b + \frac{\gamma}{24}(s^T\hat{d})^3 s\Big) \cdot \delta + \frac{1}{2}\Big(\nabla^2 f(x_c) \\
& + \Big(b^T\hat{d} + \frac{\gamma}{2}\Big) ss^T\Big) \cdot \delta^2 + (s^T\hat{d})(b^T\delta)(s^T\delta) + \frac{1}{2}(b^T\delta)(s^T\delta)^2 \\
& + \frac{\gamma}{6}(s^T\hat{d})(s^T\delta)^3 + \frac{\gamma}{24}(s^T\delta)^4.
\end{aligned} \tag{3.7}
$$

If we let $\hat{\beta} = s^T\hat{d}$, $\hat{\theta} = b^T\hat{d}$, $\hat{g} = \nabla f(x_c) + \nabla^2 f(x_c)\hat{d} + \hat{\theta}\hat{\beta}s + \frac{1}{2}\hat{\beta}^2 b + \frac{\gamma}{6}\hat{\beta}^3 s$, $c = b^T\hat{d} + \frac{\gamma}{2}$, and $\hat{H} = \nabla^2 f(x_c) + css^T$, then we obtain the modified tensor model

$$
\begin{aligned}
M_T(x_c + d) \;=\; & M_T(x_c + \hat{d}) + \hat{g} \cdot \delta + \frac{1}{2}\hat{H} \cdot \delta^2 + \hat{\beta}(b^T\delta)(s^T\delta) \\
& + \frac{1}{2}(b^T\delta)(s^T\delta)^2 + \frac{\gamma}{6}\hat{\beta}(s^T\delta)^3 + \frac{\gamma}{24}(s^T\delta)^4.
\end{aligned} \tag{3.8}
$$

The advantage of this transformation is that the matrix $\hat{H}$ is likely to be nonsingular if the rank of $\nabla^2 f(x_c)$ is at least $n - 1$. A necessary and sufficient condition for $\hat{H}$ to be nonsingular is given in the following lemma. Let g and H denote $\nabla f(x_c)$ and $\nabla^2 f(x_c)$, respectively.

LEMMA 3.1. *Let* $H \in \Re^{n \times n}$, $s \in \Re^n$.

$$
H + css^T \textit{ is nonsingular if and only if } M = \begin{bmatrix} H & cs \\ cs^T & -c \end{bmatrix} \textit{ is nonsingular.}
$$

(*Note that the* $\begin{bmatrix} s^T & -1 \end{bmatrix}$ *submatrix was premultiplied by the constant* c *to symmetrize the augmented matrix* M.)

Proof. We prove that there exists $v \in \Re^n, v \neq 0$, for which $(H + css^T)v = 0$, if and only if there exist $\bar{v} \in \Re^n, w \in \Re$, for which

$$\begin{bmatrix} H & cs \\ cs^T & -c \end{bmatrix} \begin{bmatrix} \bar{v} \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} \bar{v} \\ w \end{bmatrix} \neq \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{3.9}$$

Suppose first that $(H + css^T)v = 0, v \neq 0$. Then for $\bar{v} = v, w = s^T v$, $(\bar{v}, w)$ satisfies (3.9). Conversely, if there exists $(\bar{v}, w)$ satisfying (3.9), then $s^T \bar{v} = w$, so $(H + css^T)\bar{v} = 0$, and $\bar{v} \neq 0$; otherwise, $w = 0$, which contradicts (3.9). Thus $(H + css^T)$ is singular if and only if M is singular.

COROLLARY 3.2. *Let $H \in \Re^{n \times n}$, $s \in \Re^n$.*

If $H + css^T$ is nonsingular, then $\begin{bmatrix} H & cs \end{bmatrix}$ has full row rank.

Proof. The proof follows from Lemma 3.1.

LEMMA 3.3. *Let $H \in \Re^{n \times n}$, $\text{rank}(H) = n - 1$, $s \in \Re^n$.*

$H + css^T$ is nonsingular if and only if $\begin{bmatrix} H & cs \end{bmatrix}$ has full row rank.

Proof. The *only if* part follows from Corollary 3.2. Now assume $\begin{bmatrix} H & cs \end{bmatrix}$ has full row rank. Since H has rank $n - 1$, $H = H_1 H_2^T$, where $H_1, H_2 \in \Re^{n \times (n-1)}$ have full column rank. Since $\begin{bmatrix} H & cs \end{bmatrix}$ has full row rank,

$$(v^T H = 0 \text{ and } v^T s = 0) \Rightarrow v = 0. \tag{3.10}$$

From $H = H_1 {H_2}^T$ and the fact that H_2 has full column rank, (3.10) is equivalent to

$$(v^T H_1 = 0 \text{ and } v^T s = 0) \Rightarrow v = 0.$$

Thus the $n \times n$ matrix $\begin{bmatrix} H_1 & cs \end{bmatrix}$ is nonsingular. Analogously, the $n \times n$ matrix $\begin{bmatrix} H_2 & s \end{bmatrix}$ is nonsingular. Therefore

$$\begin{bmatrix} H_1 & cs \end{bmatrix} \begin{bmatrix} H_2^T \\ s^T \end{bmatrix} = H_1 {H_2}^T + css^T = H + css^T$$

is nonsingular. □

Even though $\hat{H}$ is not sparse in general, Lemma 3.1 will be used later on to exploit the sparsity of H when working with $\hat{H}$.

For δ to be a local minimizer of (3.8) the derivative of the tensor model (3.8) with respect to δ must be zero. That is,

$$\begin{aligned} \nabla M_T(x_c + \delta) \quad = \quad & \hat{g} + \hat{H}\delta + \hat{\beta}(s^T\delta)b + \hat{\beta}(b^T\delta)s + (s^T\delta)(b^T\delta)s \\ & + \left(\frac{1}{2}b + \frac{\gamma}{2}\hat{\beta}s\right)(s^T\delta)^2 + \frac{\gamma}{6}(s^T\delta)^3 s = 0, \end{aligned} \tag{3.11}$$

which yields

$$\begin{aligned} \delta \quad = \quad -\hat{H}^{-1}\Bigg(& \hat{g} + \hat{\beta}(s^T\delta)b + \hat{\beta}(b^T\delta)s + (s^T\delta)(b^T\delta)s \\ & + \left(\frac{1}{2}b + \frac{\gamma}{2}\hat{\beta}s\right)(s^T\delta)^2 + \frac{\gamma}{6}(s^T\delta)^3 s\Bigg). \end{aligned} \tag{3.12}$$

Premultiplying (3.12) by s^T on both sides results in a cubic equation (in β) in the two unknowns $\beta = s^T\delta$ and $\theta = b^T\delta$:

$$\begin{aligned} &s^T\hat{H}^{-1}\hat{g} + (1+\hat{\beta}s^T\hat{H}^{-1}b)\beta + \hat{\beta}s^T\hat{H}^{-1}s\theta + s^T\hat{H}^{-1}s\beta\theta \\ &+\left(\frac{1}{2}s^T\hat{H}^{-1}b + \frac{\gamma}{2}\hat{\beta}s^T\hat{H}^{-1}s\right)\beta^2 + \frac{\gamma}{6}s^T\hat{H}^{-1}s\beta^3 = 0. \end{aligned} \tag{3.13}$$

The premultiplication of (3.12) by b^T on both sides yields another cubic equation (in β) in the two unknowns β and θ:

$$\begin{aligned} &b^T\hat{H}^{-1}\hat{g} + (1+\hat{\beta}b^T\hat{H}^{-1}s)\theta + \hat{\beta}b^T\hat{H}^{-1}b\beta + b^T\hat{H}^{-1}s\beta\theta \\ &+\left(\frac{1}{2}b^T\hat{H}^{-1}b + \frac{\gamma}{2}\hat{\beta}b^T\hat{H}^{-1}s\right)\beta^2 + \frac{\gamma}{6}b^T\hat{H}^{-1}s\beta^3 = 0. \end{aligned} \tag{3.14}$$

Therefore, we obtain a system of two cubic equations in the two unknowns β and θ, which we can solve analytically.

Since (3.13) is linear in θ, we can compute θ as a function of β and then substitute its expression into (3.14) to obtain an equation in the one unknown β. Let $u = s^T\hat{H}^{-1}\hat{g}$, $v = s^T\hat{H}^{-1}b$, $w = s^T\hat{H}^{-1}s$, $y = b^T\hat{H}^{-1}\hat{g}$, and $z = b^T\hat{H}^{-1}b$. Equation (3.13) yields

$$\begin{aligned} \theta = \frac{1}{w(\hat{\beta}+\beta)}\Bigg(&yw\hat{\beta} - u - uv\hat{\beta} + (yw + zw\hat{\beta}^2 - 2v\hat{\beta} - v^2\hat{\beta}^2 - uv - 1)\beta \\ &+\left(\frac{3}{2}zw\hat{\beta} - \frac{\gamma}{2}w\hat{\beta} - \frac{3}{2}v - \frac{3}{2}v^2\hat{\beta}\right)\beta^2 + \left(\frac{1}{2}zw - \frac{\gamma}{6}w - \frac{v^2}{2}\right)\beta^3\Bigg). \end{aligned} \tag{3.15}$$

The denominator of (3.15) is equal to zero if either $\hat{\beta}+\beta = 0$ or $w = 0$. If $w = 0$, then (3.13) would be quadratic in β. Therefore

$$\beta = \frac{-(1+\hat{\beta}v) \pm \sqrt{(1+\hat{\beta}v)^2 - 2uv}}{v}.$$

Hence, real-valued minimizers of the tensor model (3.8) may exist only if $(1+\hat{\beta}v)^2 \geq 2uv$ and $v \neq 0$. It is straightforward to verify from (3.14) that for θ to be defined, $(\hat{\beta}+\beta)v$ cannot equal -1. Now, if $\hat{\beta}+\beta = 0$, then (3.13) reduces to the following cubic equation in β:

$$u + (1+\hat{\beta}v)\beta + \left(\frac{1}{2}v + \frac{\gamma}{2}w\hat{\beta}\right)\beta^2 + \frac{\gamma}{6}w\beta^3 = 0. \tag{3.16}$$

Once we calculate the expressions for β from (3.16), we substitute them into the following equation for θ obtained from (3.14):

$$\theta \ = -y - z\hat{\beta}\beta - \left(\frac{1}{2}z + \frac{\gamma}{2}v\hat{\beta}\right)\beta^2 - \frac{\gamma}{6}v\beta^3.$$

If neither $\hat{\beta}+\beta = 0$ nor $w = 0$, we substitute the expression (3.15) into (3.14) and obtain

$$\begin{aligned} &-(u + 2\hat{\beta}v + \hat{\beta}uv + \hat{\beta}^2v^2 + 1) + (yw + \hat{\beta}^2zw - \hat{\beta}v - v - uv)\beta \\ &+\left(\hat{\beta}^2zw + \frac{1}{2}\hat{\beta}zw - \frac{1}{2}v - \frac{\gamma}{2}\hat{\beta}w - \frac{1}{2}\hat{\beta}v^2\right)\beta^2 + \left(\frac{1}{2}zw - \frac{\gamma}{6}w - \frac{1}{2}v^2\right)\beta^3 = 0, \end{aligned} \tag{3.17}$$

which is a third-order polynomial in the one unknown β. The roots of (3.17) are then computed analytically. After we determine the values of β, we substitute them into (3.15) to calculate the corresponding values of θ. Then, we simply substitute the values of β and θ into (3.12) to obtain the values of δ. The dominant cost in this whole process is the computation of $\hat{H}^{-1}\hat{g}$, $\hat{H}^{-1}b$, and $\hat{H}^{-1}s$.

Similar to the nonsingular case, a minimizer δ is selected such that there exists a descent path from the current point x_c to $x_c + \delta$, and that δ is closest to x_c in the Euclidean norm sense.

To obtain the tensor step d, we set d to $\hat{d} + \delta$. An appropriate choice of $\hat{d}$ is the step used in the previous iteration simply because it has the right scale.

The above procedure is tailored to handle only the case where the Hessian matrix has rank $n - 1$. It has been shown in practice that when $\nabla^2 f(x_*)$ has rank $n - 1$ the convergence rate of the tensor method is better than the linear convergence of the standard Newton method [19] (also see section 7 for the ratios of the errors of successive iterates on the BRYBND problem with $\operatorname{rank}(\nabla^2 f(x_*)) = n - 1$). Tensor methods for nonlinear equations problems have been shown to have three-step Q-order 1.5 convergence on problems where the Jacobian has rank $n - 1$ at the solution [11], whereas Newton's method is linearly convergent with constant 1/2 on such problems. However, no attempt has been made yet to prove the convergence rate of tensor methods for unconstrained optimization problems where the Hessian at the solution has rank $n - 1$. On problems where $\operatorname{rank}(\nabla^2 f(x_*)) \leq n - 2$, tensor methods do not have enough information to prove a faster-than-linear convergence rate, since it usually uses $p = 1$. Consequently, when $\operatorname{rank}(\nabla^2 f(x_*)) \leq n - 2$ we simply use the modified Newton step (see section 6) as the step direction for the current iteration.

4. Line search backtracking techniques.

The line search global strategy we use in conjunction with our tensor method for large, sparse unconstrained optimization is similar to the one used for nonlinear equations [4, 6]. This strategy has been shown to be very successful for large, sparse systems of nonlinear equations. We also found that it is superior to the approach used by Schnabel and Chow [19]. The main difference between the two approaches is that ours always tries the full tensor step first. If this provides enough decrease in the objective function, then we terminate; otherwise we find acceptable next iterates in both the Newton and tensor directions and select the one with the lower function value as the next iterate. Schnabel and Chow, on the other hand, always find acceptable next iterates in both the Newton and tensor directions and choose the one with the lower function value as the next iterate. In practice, our approach almost always requires fewer function evaluations while retaining the same efficiency in iteration numbers. The global framework for line search methods for unconstrained minimization is given in Algorithm 4.1.

ALGORITHM 4.1. GLOBAL FRAMEWORK FOR LINE SEARCH METHODS FOR UNCONSTRAINED MINIMIZATION.

```
Let $x_c$ be the current iterate,
$d_t$ the tensor step,
$d_n$ is the Newton step,
$g = \nabla f(x_c)$,
$f_c = f(x_c)$,
$slope = g^T d_t$,
and $\alpha = 10^{-4}$.
    $x_+^t = x_c + d_t$
    $f_p = f(x_+^t)$
```

```
if (minimizer of the tensor model was found) then
    if $f_p < f_c + \alpha \cdot slope$ then
        $x_+ = x_+^t$
    else
        Find an acceptable $x_+^n$ in the Newton direction $d_n$
        using the line search given by Algorithm A6.3.1 [9, p. 325]
        Find an acceptable $x_+^t$ in the tensor direction $d_t$
        using the line search given by Algorithm A6.3.1 [9, p. 325]
        if $f(x_+^n) < f(x_+^t)$ then
            $x_+ = x_+^n$
        else
            $x_+ = x_+^t$
        endif
    endif
else
    Find an acceptable $x_+^n$ in the Newton direction $d_n$
    using the line search given by Algorithm A6.3.1 [9, p. 325]
    $x_+ = x_+^n$
endif
```

5. Model trust region techniques. The two computational methods—the locally constrained optimal (or "hook") step and the dogleg step—are generally used for approximately solving the trust region problem based on the standard model,

$$\text{minimize } f(x_c) + \nabla f(x_c) \cdot d + \frac{1}{2} \nabla^2 f(x_c) \cdot d^2 \qquad (5.1)$$

$$\text{subject to } \| \, d \, \|_2 \leq \delta_c,$$

where δ_c is the current trust region radius. When δ_c is shorter than the Newton step, the locally constrained optimal step [17] finds a μ_c such that $\| \, d(\mu_c) \, \|_2 \approx \delta_c$, where $d(\mu_c) = -(\nabla^2 f(x_c) + \mu I)^{-1} \nabla f(x_c)$. Then it takes $x_+ = x_c + d(\mu_c)$. The dogleg step is a modification of the trust region algorithm introduced by Powell [18]. However, rather than finding a point $x_+ = x_c + d(\mu_c)$ on the curve $d(\mu_c)$ such that $\| \, x_+ - x_c \, \| \approx \delta_c$, it approximates this curve by a piecewise linear function in the subspace spanned by the Newton step and the steepest descent direction $-\nabla f(x_c)$, and takes x_+ as the point on this approximation for which $\| \, x_+ - x_c \, \| = \delta_c$. (See, e.g., [9] for more details.)

Unfortunately, these two methods are hard to extend to the tensor model, which is a fourth-order model. Trust region algorithms based on (5.1) are well defined because it is always possible to find a unique point x_+ on the curve such that $\| \, x_+ - x_c \, \| = \delta_c$. Additionally, the value of $f(x_c) + \nabla f(x_c) \cdot d + \frac{1}{2} \nabla^2 f(x_c) \cdot d^2$ along the curve $d(\mu_c)$ is monotonically decreasing from x_c to x_+^n, where $x_+^n = x_c + d_n$, which makes the process reasonable. These properties do not extend to the tensor model, which is a fourth-order model that may not be convex. Furthermore, the analogous curve to $d(\mu_c)$ is more expensive to compute. For these reasons, we consider a different trust region approach for our tensor methods.

The trust region approach that is discussed in this section is a two-dimensional trust region step over the subspace spanned by the steepest descent direction and the tensor (or standard) step. The main reasons that lead us to adopt this approach are that it is easy to construct and closely related to dogleg-type algorithms over the same subspace. This step may be close to optimal trust region step algorithms in practice. Byrd, Schnabel, and Shultz [7] have shown that for unconstrained optimization using a standard quadratic model, the analogous two-dimensional minimization approach produces nearly as much decrease in the quadratic model as the optimal trust region step in almost all cases.

The two-dimensional trust region approach for the tensor model computes an approximate solution to

$$\text{minimize } f(x_c) + \nabla f(x_c) \cdot d + \frac{1}{2}\nabla^2 f(x_c) \cdot d^2 + \frac{1}{2}(b^T d)(s^T d)^2 + \frac{\gamma}{24}(s^T d)^4$$

$$\text{subject to } \|\, d \,\|_2 \leq \delta_c,$$

by performing a two-dimensional minimization,

$$\text{minimize } f(x_c) + \nabla f(x_c) \cdot d + \frac{1}{2}\nabla^2 f(x_c) \cdot d^2 + \frac{1}{2}(b^T d)(s^T d)^2 + \frac{\gamma}{24}(s^T d)^4 \tag{5.2}$$

$$\text{subject to } \|\, d \,\|_2 \leq \delta_c, \quad d \in [d_t, g_s],$$

where d_t and g_s are the tensor step and the steepest descent direction, respectively, and δ_c is the trust region radius. This approach will always produce a step that reduces the quadratic model by at least as much as a dogleg-type algorithm, which reduces d to a piecewise linear curve in the same subspace. At each iteration of the tensor algorithm, the trust region method either solves (5.2) or minimizes the standard linear model over the two-dimensional subspace spanned by the standard Newton step and the steepest descent direction. The decision of whether to use the tensor or standard model is made using the following criterion:

```
if ( (no minimizer of the tensor model was found)
    or $(\nabla f(x_c)^T d_t > -10^{-4}\|\, \nabla f(x_c) \,\|_2 \|\, d_t \,\|_2)$ ) then
  $x_+ = x_c + \alpha d_n - \beta g_s$; $\alpha$, $\beta$ selected by trust region algorithm
else
  $x_+ = x_c + \alpha d_t - \beta g_s$; $\alpha$, $\beta$ selected by trust region algorithm
endif
```

Before we define the two-dimensional trust region step for tensor methods, we show how to convert the problem

$$\text{minimize } f(x_c) + \nabla f(x_c) \cdot d + \frac{1}{2}\nabla^2 f(x_c) \cdot d^2 + \frac{1}{2}(b^T d)(s^T d)^2 + \frac{\gamma}{24}(s^T d)^4 \tag{5.3}$$

$$\text{subject to } \|\, d \,\|_2 = \delta_c, \quad d \in [\ d_t, g_s\ ],$$

to an unconstrained minimization problem.

First, we make g_s orthogonal to d_t by performing the Householder transformation:

$$\hat{g}_s = g_s - d_t \frac{g_s^T d_t}{d_t^T d_t}; \tag{5.4}$$

then, we normalize both $\hat{g}_s$ and d_t to obtain

$$\tilde{d}_t = \frac{d_t}{\|\, d_t \,\|_2}, \tag{5.5}$$

$$\tilde{g}_s = \frac{\hat{g}_s}{\|\ \hat{g}_s\ \|_2}. \tag{5.6}$$

Since d is in the subspace spanned by the tensor step $\tilde{d}_t$ and the steepest descent direction $\tilde{g}_s$, it can be written as

$$d = \alpha \tilde{d}_t + \beta \tilde{g}_s, \quad \alpha, \beta \in \Re. \tag{5.7}$$

If we square the l_2 norm of this expression for d and set it to δ^2, we obtain the following equation for β as a function of α:

$$\beta = \sqrt{\delta^2 - \alpha^2}.$$

Substituting this expression for β into (5.7) and then the resulting d into (5.3) yields the global minimization problem in the one variable α, given by (5.8) below. Thus, problems (5.8) and (5.3) are equivalent. Let $g_{hg} = \tilde{g}_s^T H \tilde{g}_s$, $d_{hd} = \tilde{d}_t^T H \tilde{d}_t$, $d_{hg} = \tilde{d}_t^T H \tilde{g}_s$, $b_t = b^T \tilde{d}_t$, $s_t = s^T \tilde{d}_t$, $b_g = b^T \tilde{g}_s$, and $s_g = s^T \tilde{g}_s$.

$$\begin{aligned}
\text{minimize} \quad f(x_c) \quad & + \frac{1}{2}\delta_c^2 g_{hg} + \frac{\gamma}{24}\delta_c^4 s_g^4 + (1 + \delta_c^2 b_g s_g^2)\sqrt{\delta_c^2 - \alpha^2} \\
& + (d_{hg} + \frac{\gamma}{6}\delta_c^2 s_t s_g^3)\alpha\sqrt{\delta_c^2 - \alpha^2} + (b_t s_g s_t + b_g s_t^2 + b_t s_t s_g \\
& - b_g s_g^2)\alpha^2\sqrt{\delta_c^2 - \alpha^2} + (\delta_c^2 b_g s_g s_t + \delta_c^2 b_t s_g^2 + \delta_c^2 b_g s_t s_g)\alpha \\
& + \left(\frac{1}{2}d_{hd} - \frac{1}{2}g_{hg} + \frac{1}{2}b_t s_t^2 + \frac{\gamma}{4}\delta_c^2 s_t^2 s_g^2 - \frac{\gamma}{12}\delta_c^2 s_g^4\right)\alpha^2 \\
& - (b_g s_g s_t + b_t s_g^2 + b_g s_t s_g)\alpha^3 + \left(\frac{\gamma}{24}s_t^4 - \frac{\gamma}{4}s_t^2 s_g^2 + \frac{\gamma}{24}s_g^4\right)\alpha^4 \\
& + \left(\frac{\gamma}{6}s_t^3 s_g - \frac{\gamma}{6}s_t s_g^3\right)\alpha^3\sqrt{\delta_c^2 - \alpha^2},
\end{aligned} \tag{5.8}$$

where $-\delta_c < \alpha < \delta_c$.

To transform the problem

$$\text{minimize } f(x_c) + \nabla f(x_c) \cdot d + \frac{1}{2}\nabla^2 f(x_c) \cdot d^2 \tag{5.9}$$

$$\text{subject to } \|\ d\ \|_2 = \delta_c, \quad d \in [\ d_n, g\ ]$$

to an unconstrained minimization problem, we use the same procedure described above to show that (5.9) is equivalent to the following global minimization problem in the one variable α:

$$\begin{aligned}
\text{minimize} \qquad & f(x_c) + \frac{1}{2}\delta_c^2 g_{hg} + \sqrt{\delta_c^2 - \alpha^2} \\
& + d_{hg}\alpha\sqrt{\delta_c^2 - \alpha^2} + \left(\frac{1}{2}d_{hd} - \frac{1}{2}g_{hg}\right)\alpha^2,
\end{aligned} \tag{5.10}$$

where $-\delta_c < \alpha < \delta_c$.

Algorithm 5.1. Two-Dimensional Trust Region for Tensor Methods. Let d_t be the tensor step,
d_n the standard step,
x_c the current iterate,

$f_c = f(x_c)$,
x_+ the next iterate,
$f_+ = f(x_+)$,
$g_s = -\nabla f(x_c)$, the steepest descent direction,
$g_c = \nabla f(x_c)$,
$H_c = \nabla^2 f(x_c)$,
and δ_c the current trust region radius.
$\tilde{d}_t, \tilde{g}_s$ are given by (5.5) and (5.6), respectively.
$\tilde{d}_n$ is obtained in an analogous way to $\tilde{d}_t$; by applying transformations (5.4) and (5.5) to it.

```
1. if tensor model selected then
      Solve problem (5.8) using the procedure described in Algorithm 3.4 [6]
   else {standard Newton model selected}
      Solve problem (5.10) using the procedure described in Algorithm 3.4 [6]
   endif
2. if tensor model selected then
      $d = \alpha_* \tilde{d}_t + \tilde{g}_s \sqrt{\delta_c^2 - \alpha_*^2}$
      where $\alpha_*$ is the global minimizer of (5.8)
   else {standard Newton model selected}
      $d = \alpha_* \tilde{d}_n + \tilde{g}_s \sqrt{\delta_c^2 - \alpha_*^2}$
      where $\alpha_*$ is the global minimizer of (5.10)
   endif
3. { Check new iterate and update trust region radius.}
   $x_+ = x_c + d$
   if $\dfrac{f_+ - f_c}{pred} \geq 10^{-4}$ then
       the global step $d$ is successful
   else
       decrease trust region
       go to step 1
   endif
   where
   $pred = \left( f_c + g_c \cdot d + \frac{1}{2} H_c \cdot d^2 + \frac{1}{2}(b^T d)(s^T d)^2 + \frac{\gamma}{24}(s^T d)^4 \right) - f_c$, if tensor model selected,
   $pred = \left( f_c + g_c \cdot d + \frac{1}{2} H_c \cdot d^2 \right) - f_c$, if standard Newton model selected.
```

The methods used for adjusting the trust radius during and between steps are given in Algorithm A6.4.5 [9, p. 338]. The initial trust radius can be supplied by the user; if not, it is set to the length of the initial Cauchy step.

6. A high-level algorithm for the tensor method.

In this section, we present the overall algorithm for the tensor method for large, sparse unconstrained optimization. Algorithm 6.1 is a high-level description of an iteration of the tensor method that was described in sections 3–5. A summary of the test results for this implementation is presented in section 7.

Algorithm 6.1. An Iteration of the Tensor Method for Large, Sparse Unconstrained Optimization.

Let x_c be the current iterate,
d_t the tensor step,

and d_n the Newton step.

1. Calculate $\nabla f(x_c)$ and decide whether to stop. If not:
2. Calculate $\nabla^2 f(x_c)$.
3. Calculate the terms T_c and V_c in the tensor model, so that the tensor model interpolates $f(x)$ and $\nabla f(x)$ at the past point.
4. Find a potential minimizer d_t of the tensor model (3.1). If d_t cannot be found, then calculate the modified Newton step d_n.
5. Find an acceptable next iterate x_+ using either a line search or a two-dimensional trust region global strategy.
6. $x_c = x_+$,
   $f(x_c) = f(x_+)$,
   go to step 1.

In step 1, the gradient is either computed analytically or approximated by Algorithm A5.6.3 given in Dennis and Schnabel [9]. In step 2, the Hessian matrix is either calculated analytically or approximated by a graph coloring algorithm described in [8]. Note that it is crucial to supply an analytic gradient if the finite-difference Hessian matrix requires many gradient evaluations. Otherwise, the methods described in this paper may not be practical, and inexact methods may be preferable. The procedures for calculating T_c and V_c in step 3 were discussed in section 2. In step 4, the Hessian matrix is factored using MA27 [10], a sparse Cholesky decomposition package. If the Hessian matrix is nonsingular, then the tensor step d_t is calculated as described in section 3.1. Otherwise, if the Hessian matrix is singular with rank $n-1$, then d_t is computed as outlined in section 3.2. (We comment on the implementation issues related to this case in the next paragraph.) If the rank of the Hessian matrix is less than $n-1$, then the Newton step, d_n, is computed as a by-product of the minimization of the tensor model, and used as the step direction for the current iteration. This Newton step d_n is the modified Newton step $(\nabla^2 f(x_c)+\mu I)^{-1}\nabla f(x_c)$, where $\mu = 0$ if $\nabla^2 f(x_c)$ is safely positive definite, and $\mu > 0$ otherwise. To obtain the perturbation μ, we use a modification of MA27 advocated by Gill et al. in [13]. In this method we first compute the LDL^T of the Hessian matrix using the MA27 package, then change the block diagonal matrix D to $D+E$. The modified matrix is block diagonal positive definite. This guarantees that the decomposition $L(D+E)L^T$ is positive definite as well. Note that the Hessian matrix is not modified if it is already positive definite.

Another implementation issue that deserves some attention is how to solve linear systems of the form $\hat{H}x = b$, where $\hat{H} = H + css^T$, $H \in \Re^{n\times n}$ is sparse and rank deficient, and $s \in \Re^n$ is full (see section 3.2). Such systems can be efficiently solved using the augmented matrix defined in Lemma 3.1. That is, we write $(H+css^T)x = b$ as

$$\begin{bmatrix} H & cs \\ cs^T & -c \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}. \tag{6.1}$$

The $(n+1)\times(n+1)$ matrix in (6.1) is sparse and can be factored efficiently as long as the last row and column are not pivoted until the last few iterations. In fact, we can combine the nonsingular and singular cases by factoring H, but we shift to a factorization of the augmented matrix if H is discovered to be singular with rank $n-1$. However, we use a Schur complement method to obtain the solution of the augmented matrix by updating the solution from the system $Hx = b$. This choice was motivated by the fact that the Schur complement method is simpler and more convenient to use than the factorization of the augmented matrix in (6.1). Note that if the Schur complement method shows that the augmented matrix in (6.1) is rank deficient (a case that is very rare in practice), the modified Newton step described above is used as the step direction for the current iteration.

The Schur complement method requires that H must have full rank. Thus, some modifications are necessary in order for this method to work. We have modified the factorization phase of MA27 to be able to detect the row and column indices of the first pivot whose absolute value is less than or equal to some given tolerance tol. This stability test is clearly not optimal but appears to work in practice. We also modified the solve phase of MA27 such that whenever a pivot fails the stability criterion above, the corresponding solution component is set to zero. This way the solution of $Hx = b$ is the same as the solution of $H_e y = b$ (where H_e is the matrix H minus the row and column at which singularity occurred. Since y has $n - 1$ components, the remaining one, which is also the component corresponding to the pivot failing the stability test, is set to 0). Afterwards, we obtain the solution of an augmented system using a Schur complement method, where the coefficient matrix is the matrix H augmented by two rows and columns; that is, the $(n + 1)$st row and column are the ones at which singularity was detected, and the $(n + 2)$nd row and column are cs^T and cs, respectively. The Schur complement method is implemented by first invoking MA39AD [1] to form the Schur complement $S = D - CH^{-1}B$ of H in the extended matrix, where D is the 2 by 2 lower right submatrix, C is the lower left 2 by n submatrix, and B is the upper right n by 2 submatrix, of the augmented matrix. The Schur complement is then factored into its QR factors. Next, MA39BD [1] solves the extended system (6.1) using the following well-known scheme:

1. Solve $Hu = b$, for u.
2. Solve $Sy = b - Cu$, for y.
3. Solve $Hv = By$, for v.
4. $x = u - v$.

The dominant cost of the above process is the $Hu = b$ and $Hv = By$ solves.

The tensor and Newton algorithms terminate if $\| \nabla f(x_c) \|_2 \leq 10^{-5}$ or $\| d \|_2 < 10^{-9}$.

7. Test results. We tested our tensor and Newton algorithms on a variety of nonsingular and singular test problems. In this section we present and discuss summary statistics of the test results.

All our computations were performed on a Sun Sparc 10 Model 40 machine using double-precision arithmetic.

First, we tested our program on the set of unconstrained optimization problems from the CUTE [3] and the MINPACK-2 [2] collections. Most of these problems have nonsingular Hessians at the solution. We also created singular test problems as proposed in [4, 20] by modifying the nonsingular test problems from the CUTE collection as follows. Let

$$f(x) = \sum_{i=1}^{m} f_i^2(x)$$

be the function to minimize, where $f_i : \Re^n \rightarrow \Re$, m is the number of element functions, and

$$F^T(x) = (f_1(x), \ldots, f_m(x)). \tag{7.1}$$

In many cases, $F(x) = 0$ at the minimizer x_*, and $F'(x_*)$ is nonsingular. Then according to [4, 20], we can create singular systems of nonlinear equations from (7.1) by forming

$$\hat{F}(x) = F(x) - F'(x_*)A(A^TA)^{-1}A^T(x - x_*), \tag{7.2}$$

where $A \in \Re^{n\times k}$ has full column rank with $1 \le k \le n$. Hence, $\hat{F}(x_*) = 0$ and $\hat{F}'(x_*)$ has rank $n-k$. For unconstrained optimization, we simply need to define the singular function

$$\hat{f}(x) = \frac{1}{2}\hat{F}(x)^T\hat{F}(x). \tag{7.3}$$

From (7.3) and $\hat{F}(x_*) = 0$, we obtain $\nabla\hat{f}(x_*) = 0$. From

$$\hat{F}'(x_*) = F'(x_*)[I - A(A^TA)^{-1}A^T]$$

and

$$\nabla^2\hat{f}(x_*) = \hat{F}'(x_*)^T\hat{F}'(x_*) + \sum_{i=1}^{m} f_i(x_*)\nabla^2 f_i(x_*) = \hat{F}'(x_*)^T\hat{F}'(x_*),$$

we know that $\nabla^2\hat{f}(x_*)$ has rank $n-k$.

By using (7.2) and (7.3), we created two sets of singular problems, with $\nabla^2\hat{f}(x_*)$ having rank $n-1$ and $n-2$, respectively, by using

$$A \in \Re^{n\times 1}, \qquad A^T = (1, 0, \dots, 0),$$

and

$$A \in \Re^{n\times 2}, \qquad A^T = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 1 & 0 & 0 & \cdot & \cdot & \cdot & 0 \end{bmatrix},$$

respectively. The reason for choosing unit vectors as columns for the matrix A is mainly to preserve the sparsity of the Hessian during the transformation (7.2).

For all our test problems we used a standard line search backtracking strategy. All the test problems with the exception of rank $n-1$ and rank $n-2$ problems were run with analytic gradients and Hessians provided by the CUTE and MINPACK-2 collections. For rank $n-1$ and $n-2$ test problems, we have modified the analytic gradients provided by the CUTE collection to take into account the modification (7.2). On the other hand, we used the graph coloring algorithm [8] to evaluate the finite difference approximation of the Hessian matrix.

A summary for the test problems whose Hessians at the solution have ranks n, $n-1$, and $n-2$ is presented in Table 7.1. The descriptions of the test problems and the detailed results are given in the Appendix. In Table 7.1 columns "better" and "worse" represent the number of times the tensor method was better and worse, respectively, than Newton's method by more than one gradient evaluation. The "tie" column represents the number of times the tensor and standard methods are required within one gradient evaluation of each other. For each set of problems, we summarize the comparative costs of the tensor and standard methods using average ratios of three measures: gradient evaluations, function evaluations, and execution times. The

Table 7.1
Summary of the CUTE *and* MINPACK-2 *test problems using line search.*

Rank	Tensor/Standard			Pbs Solved		Average Ratio–Tensor/Standard		
$\nabla^2 f(x_*)$	better	tie	worse	t/s	s/t	feval	geval	time
n	53	38	5	4	0	1.09	0.69	0.67
$n-1$	18	2	0	5	0	0.52	0.48	0.41
$n-2$	18	1	1	7	0	0.70	0.63	0.66

average gradient evaluation ratio (geval) is the total number of gradient evaluations required by the tensor method, divided by the total number of gradient evaluations required by the standard method on these problems. The same measure is used for the average function evaluation (feval) and execution time (time) ratios. These average ratios include only problems that were successfully solved by both methods. We have excluded all cases where the tensor and standard methods converged to a different minimizer. However, the statistics for the "better," "worse," and "tie" columns include the cases where only one of the two methods converges, and exclude the cases where both methods do not converge. We also excluded problems requiring a number of gradient evaluations less than or equal to 3 by both methods. Finally, columns "t/s" and "s/t" show the number of problems solved by the tensor method but not by the standard method and the number of problems solved by the standard method but not by the tensor method, respectively.

The improvement by the tensor method over the standard method on problems with rank $n-1$ is dramatic, averaging 48% in function evaluations, 52% in gradient evaluations, and 59% in execution times. This is due in part to the rate of convergence of the tensor method being faster than that of Newton's method, which is known to be only linearly convergent with constant $\frac{2}{3}$. On problems with rank $n-2$, the improvement by the tensor method over the standard method is also substantial, averaging 30% in function evaluations, 37% in gradient evaluations, and 34% in execution times. In the test results obtained for the nonsingular problems, the tensor method is 9% worse than the standard method in function evaluations, but 31% and 33% better in gradient evaluations and in execution times, respectively. The main reason for the tensor method requiring on the average more function evaluations than the standard method is because on some problems, the full tensor step does not provide sufficient decrease in the objective function, and therefore the tensor method has to perform a line search in both the Newton and tensor directions, which causes the number of function evaluations required by the tensor method to be inflated. As a result, we intend to investigate other possible global frameworks for line search methods that could potentially reduce the number of function evaluations for the tensor method.

To obtain an experimental indication of the local convergence behavior of the tensor and Newton methods on problems where $\text{rank}(\nabla^2 f(x_*)) = n-1$, we examined the sequence of ratios

$$\frac{||\ x_k - x_*\ ||}{||\ x_{k-1} - x_*\ ||} \tag{7.4}$$

produced by the Newton and tensor methods on such problems. These ratios for a typical problem are given in Table 7.2. In almost all cases the standard method exhibits local linear convergence with constant near $\frac{2}{3}$, which is consistent with the theoretical analysis. The local convergence rate of the tensor method is faster, with a typical final ratio of around 0.01. Whether this is a superlinear convergence remains to

Table 7.2

Speed of convergence on the BRYBND problem with rank$(\nabla^2 f(x_*)) = n - 1$*, as modified by* (7.2)*,* $n = 5000$*, started from* x_0*. The ratios in the second and third columns are defined by* (7.4)*.*

Iteration (k)	Standard Method	Tensor Method
1	0.659	0.659
2	0.655	0.033
3	0.650	0.459
4	0.641	0.961
5	0.629	0.850
6	0.612	0.667
7	0.590	0.410
8	0.571	0.323
9	0.600	0.126
10	0.760	0.012
11	0.940	
12	0.988	
13	0.970	
14	0.969	
15	0.956	
16	0.926	
17	0.891	
18	0.909	
19	0.848	
20	0.926	
21	0.939	
22	0.896	
23	0.832	
24	0.871	
25	0.742	
26	0.667	
27	0.667	
28	0.666	
29	0.665	
30	0.666	

be determined. We have done similar experiments for problems with rank$(\nabla^2 f(x_*)) = n - 2$, and the tensor method did not show a faster-than-linear convergence rate, because it did not have enough information since $p = 1$.

The tensor method solved a total of four nonsingular problems, five rank $n - 1$ problems, and seven rank $n - 2$ problems that Newton's method failed to solve. The reverse never occurred. These results clearly indicate that the tensor method is most likely to be more robust than Newton's method.

The overall results show that having some extra information about the function and gradient in the past step direction is quite useful in achieving the advantages of tensor methods.

8. Summary and future research.

In this paper we presented new algorithms for solving large, sparse unconstrained optimization problems using tensor methods. Implementations using these tensor methods have been shown to be considerably more efficient, especially on problems where the Hessian matrix has a small rank deficiency at the solution. Typical gains over standard Newton methods range from 40% to 50% in function and gradient evaluations and in computer time. The size and consistency of the efficiency gains indicate that the tensor method may be preferable to Newton's method for solving large, sparse unconstrained optimization problems where analytic gradients and/or Hessians are available. To firmly establish such a conclusion, additional testing is required, including test problems of very large size.

On sparse problems where the function or the gradient is expensive to evaluate, the finite-difference approximation of the Hessian matrix by the graph coloring algorithm [8] may be very costly. Hence, quasi-Newton methods may be preferable to use in this case. These methods involve low-rank corrections to a current approximate Hessian matrix. We are currently attempting to extend our tensor methods to quasi-Newton methods for large, sparse unconstrained minimization problems.

We also considered solving large, sparse, structured unconstrained optimization problems using tensor methods. In this variant, we explored the possibility of using exact third- and fourth-order derivative information. The calculation of these derivatives is simplified using the concept of *partial separability*, a structure that has already proven to be useful when building quadratic models for large-scale nonlinear problems [16]. The calculation of the minimizer of this *exact* tensor model is more problematic, however, because we need to solve a sparse system of nonlinear equations. An obvious approach to solve these equations is to use a Newton-like method. Such a method is characterized by the approximation of the Jacobian used in the Newton process. A simple idea is to use a fixed Jacobian at each step. This has the advantage that the Jacobian will have already been obtained in the current tensor iteration. However, potential slow convergence of such a scheme may make the cost of a tensor iteration prohibitive. We are currently investigating other possible approaches, such as a modified Newton method in which the approximated Jacobian matrix will incorporate more useful information, or an iterative method such as a nonlinear GMRES. This work, in cooperation with Nick Gould [5], will be reported in the near future.

We are almost done with the implementation and testing of the two-dimensional trust region global strategy described in section 5. This work will be reported in a forthcoming paper.

We are also implementing the algorithms discussed in this paper in a software package. This package uses one past point in the formation of the tensor terms, which makes the additional cost and storage of the tensor method over the standard method very small. The package will be available soon.

Appendix A. The columns in Tables A.1–A.6 have the following meanings:

- $func$: name of the problem.
- n: dimension of the problem.
- x_0: starting point. 1, 10, 100 stand for x_0, $10x_0$, and $100x_0$, respectively.
- $initf$: initial value of the objective function.
- fcn: number of function evaluations.
- $grad$: number of gradient evaluations.
- $time$: execution time in seconds.
- $finalf$: final value of the objective function.

IL, NC stand for iteration limit exceeded and convergence to a nonminimizer, respectively. The iteration limit is 300 for the MINPACK-2 collection and 200 for the CUTE collection. All starting points were provided by the MINPACK-2 and CUTE collections.

Remark. For rank $n-1$ and $n-2$ problems $grad$ does not include the number of gradients required by Hessian evaluations. On the other hand, fcn does include the functions evaluations required by Hessian evaluations.

Table A.1
MINPACK-2 *test problems.*

Name	Description
DEPT	Elastic-plastic torsion problem
DGL1	Ginzburg–Landau (1-dimensional) superconductivity problem
DGL2	Ginzburg–Landau (2-dimensional) superconductivity problem
DLJ2	2-dimensional Leonard–Jones clusters (molecular conformation) problem
DLJ3	3-dimensional Leonard–Jones clusters (molecular conformation) problem
DMSA	Minimal surface area problem
DODC	Optimal design with composite materials problem
DPJB	Pressure distribution in a journal bearing problem
DSSC	Steady state combustion problem

Table A.2
CUTE *test problems.*

Name	Description
ARWHEAD	Quartic problem whose Hessian is an arrowhead (downwards) with diagonal central part and borderwidth 1
BDQRTIC	Quartic problem whose Hessian is banded with bandwidth 9
BRYBND	Broyden banded system of nonlinear equations, considered in the least square sense
DIXMAANA	Dixon–Maany test problem (version A)
DIXMAANB	Dixon–Maany test problem (version B)
DIXMAANC	Dixon–Maany test problem (version C)
DIXMAANI	Dixon–Maany test problem (version I)
DIXON3DQ	Dixon's tridiagonal quadratic
EDENSCH	Extended Dennis and Schnabel problem, as defined by Li
ENGVAL1	A sum of $2n-2$ groups, $n-1$ of which contain 2 nonlinear elements
FLETCBV2	Boundary value problem
FREUROTH	Freudenstein and Roth test problem
LIARWHD	A simplified version of the NONDIA problem
MOREBV	Boundary value problem. This is the nonlinear least squares version without fixed variables
NONDIA	Shanno's nondiagonal extension of Rosenbrock function
NONDQUAR	A nondiagonal quartic test problem with an arrowhead-type Hessian having a tridiagonal central part and a borderwidth 1. The Hessian is singular at the solution
PENALTY1	A sum of $n+1$ least squares groups, the first n which have only one linear element
PENALTY2	A nonlinear least squares problem with $m = 2n$ groups, group 1 is linear, groups 2 to n use 2 nonlinear elements, groups $n+1$ to $m-1$ use 1 nonlinear element, and group m uses n nonlinear elements
POWELLSG	Extended Powell singular problem
QUARTC	A simple quartic function
SINQUAD	A function with nontrivial groups and repetitious elements
SROSENBR	Separable extension of Rosenbrock's function
TQUARTIC	A quartic function with nontrivial groups and repetitious elements
TRIDIA	Shanno's TRIDIA quadratic tridiagonal problem
WOODS	Extended Woods problem
WOODS1	Scaled extended Woods problem

Table A.3
Results of the MINPACK-2 *test problems.*

				Standard				Tensor			
func	*n*	x_0	*initf*	*fcn*	*grad*	*finalf*	*time*	*fcn*	*grad*	*finalf*	*time*
DEPT	100	1	-0.36364D+01	2	2	-0.10694D+02	0	2	2	-0.10694D+02	0
DEPT	400	1	-0.36584D+01	2	2	-0.10902D+02	0	2	2	-0.10902D+02	0
DEPT	900	1	-0.36629D+01	2	2	-0.10946D+02	0	2	2	-0.10946D+02	0
DEPT	1600	1	-0.36645D+01	2	2	-0.10961D+02	1	2	2	-0.10961D+02	1
DEPT	2500	1	-0.36653D+01	2	2	-0.10969D+02	2	2	2	-0.10969D+02	2
DEPT	3600	1	-0.36657D+01	2	2	-0.10973D+02	2	2	2	-0.10973D+02	2
DEPT	4900	1	-0.36659D+01	2	2	-0.10976D+02	3	2	2	-0.10976D+02	3
DEPT	6400	1	-0.36661D+01	2	2	-0.10977D+02	5	2	2	-0.10977D+02	5
DEPT	8100	1	-0.36662D+01	2	2	-0.10978D+02	7	2	2	-0.10978D+02	7
DEPT	10000	1	-0.36663D+01	2	2	-0.10979D+02	8	2	2	-0.10979D+02	8
DGL1	100	1	-0.16619D-03	18	18	-0.84562D+04	0	5	5	-0.84562D+04	0
DGL1	400	1	-0.16619D-03	18	18	-0.84562D+04	2	9	6	-0.84562D+04	1
DGL1	900	1	-0.16619D-03	18	18	-0.84562D+04	4	6	6	-0.84562D+04	1
DGL1	1600	1	-0.16619D-03	18	18	-0.84562D+04	7	7	7	-0.84562D+04	3
DGL1	2500	1	-0.16619D-03	18	18	-0.84562D+04	11	8	8	-0.84562D+04	5
DGL1	3600	1	-0.16619D-03	19	19	-0.84562D+04	17	9	9	-0.84562D+04	9
DGL1	4900	1	-0.16619D-03	19	19	-0.84562D+04	23	7	7	-0.84562D+04	9
DGL1	6400	1	-0.16619D-03	17	17	-0.84413D+04	27	7	7	-0.84562D+04	12
DGL1	8100	1	-0.16619D-03	–	NC	–	–	7	7	-0.84562D+04	15
DGL1	10000	1	-0.16619D-03	–	NC	–	–	9	9	-0.84562D+04	24
DGL2	100	1	0.18190D+02	231	84	0.16228D+02	11	150	38	0.16228D+02	5
DGL2	400	1	0.20131D+02	159	67	0.16231D+02	45	210	43	0.16231D+02	31
DGL2	900	1	0.22015D+02	265	96	0.16232D+02	202	418	76	0.16232D+02	169
DGL2	1600	1	0.23884D+02	306	111	0.16232D+02	584	455	81	0.16232D+02	444
DGL2	2500	1	0.25748D+02	354	122	0.16232D+02	1330	607	102	0.16232D+02	1170
DGL2	3600	1	0.27609D+02	503	165	0.16232D+02	3140	751	137	0.16232D+02	2190
DGL2	4900	1	0.29469D+02	686	223	0.16232D+02	12800	849	144	0.16232D+02	6440
DLJ2	100	1	-0.10698D+03	252	107	-0.13375D+03	113	176	51	-0.13396D+03	54
DLJ2	200	1	-0.22945D+03	405	132	-0.28056D+03	1030	475	89	-0.28140D+03	698
DLJ2	300	1	-0.35261D+03	544	145	-0.44216D+03	3720	631	118	-0.44025D+03	3050
DLJ3	120	1	-0.11782D+03	375	112	-0.17954D+03	137	348	65	-0.17073D+03	81
DLJ3	210	1	-0.23253D+03	485	139	-0.34073D+03	838	608	113	-0.34522D+03	687
DLJ3	360	1	-0.42908D+03	1031	281	-0.63744D+03	8260	963	173	-0.63311D+03	4660
DMSA	100	1	0.14608D+01	4	4	0.14185D+01	0	4	4	0.14185D+01	0
DMSA	400	1	0.14891D+01	4	4	0.14206D+01	1	10	4	0.14206D+01	1
DMSA	900	1	0.15035D+01	5	5	0.14210D+01	2	4	4	0.14210D+01	2
DMSA	1600	1	0.15123D+01	5	5	0.14212D+01	4	10	5	0.14212D+01	4
DMSA	2500	1	0.15183D+01	6	6	0.14212D+01	8	14	5	0.14212D+01	8
DMSA	3600	1	0.15227D+01	6	6	0.14213D+01	13	10	6	0.14213D+01	15
DMSA	4900	1	0.15260D+01	6	6	0.14213D+01	19	11	6	0.14213D+01	21
DMSA	6400	1	0.15286D+01	7	7	0.14213D+01	31	9	7	0.14213D+01	34
DMSA	8100	1	0.15307D+01	17	12	0.14213D+01	85	16	8	0.14213D+01	60
DMSA	10000	1	0.15324D+01	21	14	0.14213D+01	117	17	7	0.14213D+01	60
DODC	100	1	0.44626D-01	14	8	-0.10980D-01	0	16	8	-0.10980D-01	0
DODC	400	1	0.47194D-01	13	10	-0.11248D-01	2	19	10	-0.11248D-01	3
DODC	900	1	0.47771D-01	23	13	-0.11329D-01	7	41	14	-0.11329D-01	9
DODC	1600	1	0.47974D-01	55	23	-0.11351D-01	26	56	21	-0.11351D-01	27
DODC	2500	1	0.48082D-01	70	33	-0.11359D-01	62	117	28	-0.11359D-01	62
DODC	3600	1	0.48139D-01	129	49	-0.11368D-01	148	194	42	-0.11368D-01	144
DODC	4900	1	0.48178D-01	565	163	-0.11372D-01	713	406	76	-0.11372D-01	380
DODC	6400	1	0.48202D-01	597	168	-0.11374D-01	999	526	94	-0.11374D-01	640
DODC	8100	1	0.48221D-01	–	IL	–	–	–	IL	–	–
DODC	10000	1	0.48234D-01	–	IL	–	–	–	IL	–	–
DPJB	100	1	0.11274D+02	2	2	-0.27881D+00	0	2	2	-0.27881D+00	0
DPJB	400	1	0.13331D+02	2	2	-0.28144D+00	0	2	2	-0.28144D+00	0
DPJB	900	1	0.14544D+02	2	2	-0.28219D+00	1	2	2	-0.28219D+00	0
DPJB	1600	1	0.15545D+02	2	2	-0.28249D+00	1	2	2	-0.28249D+00	1
DPJB	2500	1	0.16462D+02	2	2	-0.28264D+00	2	2	2	-0.28264D+00	2
DPJB	3600	1	0.17336D+02	2	2	-0.28272D+00	2	2	2	-0.28272D+00	3
DPJB	4900	1	0.18186D+02	2	2	-0.28277D+00	4	2	2	-0.28277D+00	4
DPJB	6400	1	0.19022D+02	2	2	-0.28280D+00	5	2	2	-0.28280D+00	5
DPJB	8100	1	0.19848D+02	2	2	-0.28282D+00	7	2	2	-0.28282D+00	7
DPJB	10000	1	0.20666D+02	2	2	-0.28284D+00	9	2	2	-0.28284D+00	9
DSSC	100	1	-0.52548D+01	3	3	-0.55979D+01	0	3	3	-0.55979D+01	0
DSSC	400	1	-0.50507D+01	3	3	-0.56077D+01	1	3	3	-0.56077D+01	1
DSSC	900	1	-0.49189D+01	3	3	-0.56098D+01	1	3	3	-0.56098D+01	1
DSSC	1600	1	-0.48224D+01	3	3	-0.56105D+01	2	3	3	-0.56105D+01	2
DSSC	2500	1	-0.47466D+01	3	3	-0.56108D+01	4	3	3	-0.56108D+01	4
DSSC	3600	1	-0.46842D+01	3	3	-0.56110D+01	6	3	3	-0.56110D+01	6
DSSC	4900	1	-0.46312D+01	3	3	-0.56112D+01	9	3	3	-0.56112D+01	9
DSSC	6400	1	-0.45852D+01	3	3	-0.56112D+01	12	3	3	-0.56112D+01	12
DSSC	8100	1	-0.45445D+01	3	3	-0.56113D+01	17	3	3	-0.56113D+01	18
DSSC	10000	1	-0.45080D+01	2	2	-0.56113D+01	10	2	2	-0.56113D+01	10

Table A.4
Results of the CUTE *test problems.*

				Standard				Tensor			
func	*n*	x_0	*initf*	*fcn*	*grad*	*finalf*	*time*	*fcn*	*grad*	*finalf*	*time*
ARWHEAD	5000	1	0.14997D+05	7	7	0.00000D+00	50	3	3	0.00000D+00	17
		10	0.19978D+09	12	12	0.00000D+00	91	18	14	0.00000D+00	110
		100	0.19996D+13	18	18	0.00000D+00	140	33	20	0.00000D+00	160
BDQRTIC	1000	1	0.22510D+06	10	10	0.39838D+04	10	24	12	0.39838D+04	13
		10	0.22424D+10	16	16	0.39838D+04	17	38	17	0.39838D+04	19
		100	0.22410D+14	22	22	0.39838D+04	23	51	23	0.39838D+04	25
BRYBND	5000	1	0.12490D+06	24	17	0.13587D-19	33	49	16	0.12928D-16	38
		10	0.10765D+12	37	26	0.14231D-19	51	50	24	0.98532D-17	55
		100	0.12303D+18	–	IL	–	–	810	189	0.35466D-16	473
DIXON3DQ	5000	1	0.80000D+01	2	2	0.11414D-24	1	2	2	0.11414D-24	1
		10	0.24200D+03	2	2	0.34514D-23	1	2	2	0.34514D-23	1
		100	0.20402D+05	2	2	0.29050D-21	1	2	2	0.29050D-21	1
DIXMAANA	3000	1	0.20501D+05	6	6	0.10000D+01	2	8	6	0.10000D+01	2
		10	0.80013D+10	18	12	0.10000D+01	4	19	12	0.10000D+01	5
		100	0.80000D+16	29	21	0.10000D+01	7	19	19	0.10000D+01	7
DIXMAANB	3000	1	0.43242D+05	6	6	0.10000D+01	2	15	6	0.10000D+01	2
		10	0.17227D+11	–	IL	–	–	–	IL	–	–
		100	0.16116D+17	–	IL	–	–	–	IL	–	–
DIXMAANC	3000	1	0.74483D+05	15	15	0.10000D+01	5	15	13	0.10000D+01	5
		10	0.34452D+11	–	IL	–	–	–	IL	–	–
		100	0.32233D+17	–	IL	–	–	–	IL	–	–
DIXMAANI	3000	1	0.12022D+05	100	33	0.10000D+01	12	108	18	0.10000D+01	9
		10	0.80004D+10	184	58	0.10000D+01	22	152	32	0.10000D+01	16
		100	0.80000D+16	263	77	0.10000D+01	29	247	41	0.10000D+01	21
EDENSCH	2000	1	0.73583D+07	13	13	0.12003D+05	4	31	16	0.12003D+05	7
		10	0.15184D+12	19	19	0.12003D+05	7	53	20	0.12003D+05	9
		100	0.16253D+16	24	24	0.12003D+05	8	48	25	0.12003D+05	11
ENGVAL1	5000	1	0.29494D+06	8	8	0.55487D+04	5	7	7	0.55487D+04	5
		10	0.31990D+10	14	14	0.55487D+04	10	27	14	0.55487D+04	12
		100	0.31994D+14	20	20	0.55487D+04	14	49	20	0.55487D+04	19
FLETCBV2	10000	1	-0.50013D+00	1	1	0.00000D+00	0	1	1	0.00000D+00	0
		10	0.39995D+02	2	2	-0.50013D+00	2	2	2	-0.50013D+00	2
		100	0.48995D+04	2	2	-0.50013D+00	2	2	2	-0.50013D+00	2
FREUROTH	5000	1	0.50486D+07	461	83	0.60793D+06	96	424	53	0.60821D+06	79
		10	0.15963D+09	444	77	0.60726D+06	89	200	30	0.35200D+07	41
		100	0.13056D+15	92	45	0.42206D+06	43	155	51	0.53488D+06	61
LIARWHD	10000	1	0.58500D+07	13	13	0.81983D-21	217	13	9	0.49397D-27	148
		10	0.97359D+11	22	21	0.63218D-17	363	24	12	0.11125D-16	205
		100	0.10189D+16	26	26	0.16259D-16	463	48	18	0.31712D-21	319
MOREBV	5000	1	0.15969D-06	2	2	0.58271D-14	1	2	2	0.58271D-14	1
		10	0.15983D-04	2	2	0.22833D-09	1	2	2	0.22833D-09	1
		100	0.17190D-02	2	2	0.32151D-04	1	2	2	0.32151D-04	1
NONDIA	10000	1	0.39996D+07	6	6	0.47632D-24	91	10	5	0.11200D-20	74
		10	0.12099D+11	34	34	0.53482D-25	595	20	16	0.19919D-28	274
		100	0.10200D+15	39	39	0.22382D-20	681	52	21	0.65733D-17	367
NONDQUAR	10000	1	0.10006D+05	20	20	0.41398D-09	965	20	20	0.41413D-09	970
		10	0.99981D+08	25	25	0.12450D-08	1220	25	25	0.12538D-08	1230
		100	0.99980D+12	31	31	0.73954D-09	1520	31	31	0.87210D-09	1530
PENALTY1	100	1	0.11448D+12	47	38	0.90255D-03	5	10	7	0.90249D-03	1
		10	0.11448D+16	51	43	0.90255D-03	6	7	7	0.90249D-03	1
		100	0.11448D+20	55	48	0.90257D-03	6	30	16	0.90252D-03	2
PENALTY2	100	1	0.16885D+07	24	21	0.97096D+05	3	26	20	0.97096D+05	3
		10	0.15939D+11	27	26	0.97096D+05	4	47	27	0.97096D+05	4
		100	0.15939D+15	31	31	0.97096D+05	4	70	31	0.97096D+05	5
POWELLSG	10000	1	0.53750D+06	16	16	0.10947D-04	14	33	15	0.83906D-05	18
		10	0.40385D+10	21	21	0.32920D-04	19	28	22	0.11695D-04	26
		100	0.40251D+14	27	27	0.19556D-04	25	31	27	0.54051D-05	32
QUARTC	1000	1	0.19850D+15	35	35	0.22354D-09	2	35	35	0.22354D-09	3
		10	0.18125D+15	35	35	0.20411D-09	2	35	35	0.20411D-09	3
		100	0.65804D+14	34	34	0.37515D-09	2	35	34	0.37515D-09	3
SINQUAD	10000	1	0.65610D+00	25	20	0.39609D-10	975	66	21	0.35876D-15	1030
		10	0.00000D+00	1	1	0.35876D-15	0	1	1	0.35876D-15	0
		100	0.65610D+04	18	18	0.69625D-08	881	47	19	0.42524D-15	966
SROSENBR	5000	1	0.48500D+05	9	8	0.93253D-11	3	16	7	0.10927D-17	3
		10	0.44893D+10	97	66	0.38588D-18	28	65	33	0.22535D-15	18
		100	0.51123D+14	–	IL	–	–	204	97	0.26051D-08	55
TQUARTIC	1000	1	0.81000D+00	2	2	0.39936D-27	0	2	2	0.39936D-27	0
		10	0.00000D+00	1	1	0.39936D-27	0	1	1	0.39936D-27	0
		100	0.81000D+02	2	2	0.12622D-24	0	2	2	0.12622D-24	0
TRIDIA	10000	1	0.50005D+08	2	2	0.41242D-24	1	2	2	0.41242D-24	1
		10	0.50005D+10	2	2	0.13131D-22	1	2	2	0.13131D-22	1
		100	0.50005D+12	2	2	0.33835D-20	1	2	2	0.33835D-20	1
WOODS	10000	1	0.27296D+08	28	23	0.31973D-14	26	49	21	0.33996D-17	31
		10	0.22566D+12	51	42	0.42521D-12	48	72	34	0.42039D-09	50
		100	0.22122D+16	73	60	0.27578D-10	70	100	49	0.16526D-16	73
WOODS1	10000	1	0.55500D+06	9	9	0.17486D-11	9	12	8	0.25903D-20	10
		10	0.41460D+10	15	15	0.38193D-13	17	22	14	0.26198D-19	20
		100	0.40591D+14	21	21	0.61171D-14	24	33	20	0.17403D-17	29

Table A.5
Results of the rank $n-1$ test problems from the CUTE *collection.*

				Standard				Tensor			
func	n	x_0	*initf*	*fcn*	*grad*	*finalf*	*time*	*fcn*	*grad*	*finalf*	*time*
BRYBND	5000	1	0.12488D+06	488	30	0.17586D-10	376	176	10	0.13179D-10	130
		10	0.10765D+12	–	IL	–	–	1088	60	0.85644D-10	785
		100	0.12303D+18	3396	201	0.97750D-21	2630	1560	84	0.16631D-11	1110
DIXON3DQ	5000	1	0.40000D+01	6	2	0.62536D-17	7	6	2	0.62536D-17	7
		10	0.12100D+03	6	2	0.18917D-15	7	6	2	0.18917D-15	7
		100	0.10201D+05	6	2	0.15948D-13	7	6	2	0.15948D-13	7
NONDQUAR	10000	1	0.10003D+05	–	IL	–	–	182	24	0.57721D-07	635
		10	0.99981D+08	–	IL	–	–	4414	187	0.17004D-07	6080
		100	0.99980D+12	–	IL	–	–	3820	194	0.62846D-07	5600
QUARTC	1000	1	0.45000D+05	57	15	0.61708D-05	6	13	4	0.24654D-07	1
		10	0.45000D+09	81	21	0.36635D-05	9	29	5	0.53107D-07	2
		100	0.45000D+13	101	26	0.11038D-04	11	130	22	0.50906D-06	11
SROSENBR	5000	1	0.48481D+05	30	8	0.11403D-09	48	44	7	0.45822D-12	42
		10	0.44888D+10	286	65	0.23622D-12	440	121	21	0.16587D-10	146
		100	0.51122D+14	–	IL	–	–	242	49	0.35217D-11	344
TQUARTIC	1000	1	0.32368D+04	38	12	0.38436D-15	4	17	4	0.98215D-17	2
		10	0.15962D-23	1	1	0.98215D-17	0	1	1	0.98215D-17	0
		100	0.32368D+06	23	8	0.20695D-15	3	28	9	0.14036D-15	3
TRIDIA	10000	1	0.50005D+08	6	2	0.41155D-14	27	6	2	0.41155D-14	27
		10	0.50005D+10	6	2	0.44999D-12	27	6	2	0.44999D-12	27
		100	0.50005D+12	11	3	0.14577D-13	53	11	3	0.14914D-13	54
WOODS	1000	1	0.27296D+07	248	49	0.52712D-11	24	224	32	0.41898D-10	17
		10	0.22566D+11	342	67	0.63594D-11	32	245	38	0.20790D-11	20
		100	0.22122D+15	446	87	0.44137D-11	42	308	47	0.22064D-10	25
WOODS1	1000	1	0.55491D+05	86	18	0.25201D-09	8	50	10	0.21981D-08	5
		10	0.41460D+09	116	24	0.21634D-09	11	84	16	0.40452D-08	8
		100	0.40591D+13	146	30	0.19591D-09	14	125	22	0.50008D-08	11

Table A.6
Results of the rank $n-2$ test problems from the CUTE *collection.*

				Standard				Tensor			
func	n	x_0	*initf*	*fcn*	*grad*	*finalf*	*time*	*fcn*	*grad*	*finalf*	*time*
BRYBND	5000	1	0.12487D+06	527	29	0.42357D-09	454	268	14	0.30203D-08	219
		10	0.10765D+12	824	46	0.16732D-15	724	670	32	0.34308D-10	519
		100	0.12303D+18	–	IL	–	–	1401	68	0.26897D-12	1100
DIXON3DQ	5000	1	0.80000D+01	7	2	0.62564D-17	9	7	2	0.62564D-17	9
		10	0.24200D+03	7	2	0.18928D-15	9	7	2	0.18928D-15	9
		100	0.20402D+05	7	2	0.15948D-13	9	7	2	0.15948D-13	9
NONDQUAR	10000	1	0.10002D+05	–	IL	–	–	1109	70	0.14468D-06	2710
		10	0.99980D+08	–	IL	–	–	1674	86	0.96220D-07	3320
		100	0.99980D+12	–	IL	–	–	1923	101	0.40263D-07	3820
QUARTC	1000	1	0.45000D+05	57	15	0.61708D-05	6	13	4	0.24654D-07	1
		10	0.45000D+09	81	21	0.36635D-05	9	101	17	0.53107D-07	8
		100	0.45000D+13	101	26	0.11038D-04	12	130	22	0.50906D-06	11
SROSENBR	5000	1	0.48481D+05	72	13	0.82242D-14	108	91	15	0.23908D-16	128
		10	0.44890D+10	429	77	0.69440D-04	683	465	68	0.14337D-16	615
		100	0.51122D+14	–	IL	–	–	1294	201	0.80433D+06	1830
TQUARTIC	1000	1	0.32335D+04	48	12	0.94635D-16	6	30	6	0.65443D-18	3
		10	0.15946D-23	1	1	0.15946D-23	0	1	1	0.15946D-23	0
		100	0.32335D+06	49	12	0.18893D-15	6	54	12	0.56162D-18	6
TRIDIA	10000	1	0.50005D+08	8	2	0.41344D-14	35	8	2	0.41344D-14	35
		10	0.50005D+10	8	2	0.45002D-12	35	8	2	0.45002D-12	35
		100	0.50005D+12	15	3	0.25973D-12	70	15	3	0.25973D-12	71
WOODS	1000	1	0.27277D+07	196	31	0.77284D-13	19	168	26	0.18453D-12	17
		10	0.22564D+11	325	51	0.68702D-06	32	289	41	0.10869D-12	27
		100	0.22121D+15	434	68	0.56038D-05	42	89	11	0.11251D-08	7
WOODS1	1000	1	0.55470D+05	118	18	0.18927D-09	11	91	16	0.10966D-07	10
		10	0.41458D+09	–	NC	–	–	127	22	0.30436D-08	14
		100	0.40590D+13	–	NC	–	–	31	6	0.19654D-08	3

Acknowledgments. The author thanks Professor Bobby Schnabel for his suggestions on how to minimize the tensor model when the Hessian is rank deficient, Nick Gould for discussing a number of implementation issues, Ta-Tung Chow for reviewing the first draft of the paper, and Jacko Koster for his numerous suggestions. He also thanks the referees, and Gail Pieper, from the MCS Division at Argonne National Laboratory, for their suggestions for improvement.

REFERENCES

[1] Anon, *Harwell Subroutine Library* (*Release* 11), Theoretical Studies Department, AEA Industrial Technology, Oxforshire, UK, 1993.

[2] B. M. Averick, R. G. Carter, J. J. Moré, and G. L. Xue, *The MINPACK-2 Test Problem Collection*, Tech. Rep. ANL/MCS-P153-0692, Argonne National Laboratory, Argone, IL, 1992.

[3] I. Bongartz, A. R. Conn, N. I. M. Gould, and Ph. L. Toint, CUTE*: Constrained and unconstrained testing environment*, ACM Trans. Math. Software, 21 (1995), pp. 123–160.

[4] A. Bouaricha, *Solving Large Sparse Systems of Nonlinear Equations and Nonlinear Least Squares Problems Using Tensor Methods on Sequential and Parallel Computers*, Ph.D. thesis, Computer Science Department, University of Colorado at Boulder, 1992.

[5] A. Bouaricha and N. I. M. Gould, Personal communication, Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique (CERFACS), Toulouse, France, 1994.

[6] A. Bouaricha and R. B. Schnabel, TENSOLVE*: A Software Package for Solving Systems of Nonlinear Equations and Nonlinear Least Squares Problems Using Tensor Methods*, Preprint MCS-P463-0894, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1994.

[7] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, *Approximation solution of the trust region problem by minimization over two-dimensional subspaces*, Math. Programming, 40 (1988), pp. 247–263.

[8] T. F. Coleman, B. S. Garbow, and J. J. Moré, *Estimating sparse Hessian matrices*, ACM Trans. Math. Software, 11 (1985), pp. 363–377.

[9] J. E. Dennis and R. B. Schnabel *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall, Englewood Cliffs, NJ, 1983.

[10] I. S. Duff and J. K. Reid, *MA27 : A Set of Fortran Subroutines for Solving Sparse Symmetric Sets of Linear Equations*, Tech. Rep. R-10533, AERE Harwell Laboratory, Harwell, UK, 1983.

[11] D. Feng, P. Frank, and R. B. Schnabel, *Local convergence analysis of tensor methods for nonlinear equations*, Math. Programming, 62 (1993), pp. 427–459.

[12] R. Fletcher, *Practical Method of Optimization. Volume* 1: *Unconstrained Optimization*, John Wiley and Sons, New York, 1980.

[13] P. E. Gill, W. Murray, D. B. Ponceleon, and M. A. Saunders, *Preconditioners for Indefinite Systems Arising in Optimization and Nonlinear Least Squares Problems*, Tech. Rep. SOL 90-8, Department of Operations Research, Stanford University, California, 1990.

[14] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London, 1981.

[15] A. Griewank and M. R. Osborne, *Analysis of Newton's method at irregular singularities*, SIAM J. Numer. Anal., 18 (1981), pp. 145–150.

[16] A. Griewank and Ph. L. Toint, *On the unconstrained optimization of partially separable functions*, in Nonlinear Optimization 1981, M. J. D. Powell, ed., Academic Press, New York, 1982, pp. 301–312.

[17] J. J. Moré, *The Levenberg-Marquardt algorithm: Implementation and theory*, in Proceedings Dundee 1977, G. A. Watson, ed., Lecture Notes in Mathematics 630, Springer-Verlag, Berlin, 1978, pp. 105–116.

[18] M. J. D. Powell, *A new algorithm for unconstrained optimization*, in Nonlinear Programming, J. B. Rosen, O. L. Mangasarian, and K. Ritter, eds., Academic Press, New York, 1970, pp. 33–65.

[19] R. B. Schnabel and T. Chow, *Tensor methods for unconstrained optimization using second derivatives*, SIAM J. Optim., 1 (1991), pp. 293–315.

[20] R. B. Schnabel and P. D. Frank, *Tensor methods for nonlinear equations*, SIAM J. Numer. Anal., 21 (1984), pp. 815–843.

SIAM J. OPTIM.
Vol. 7, No. 3, pp. 757–779, August 1997

010

TENSOR-GMRES METHOD FOR LARGE SYSTEMS OF NONLINEAR EQUATIONS*

DAN FENG† AND THOMAS H. PULLIAM‡

Abstract. This paper introduces a tensor-Krylov method, the tensor-GMRES method, for large systems of nonlinear equations. Krylov subspace projection techniques for asymmetric systems of linear equations are coupled with a tensor model formation and solution technique for nonlinear equations. Similar to traditional tensor methods, the new tensor method is shown to have significant computational advantages over the analogous Newton counterpart on a set of nonsingular and singular problems. For example, an application to the Euler equations for the flow through a nozzle with a given area ratio shows that the tensor-GMRES method can be much more efficient than the analogous Newton-GMRES method. The new tensor method is also consistent with preconditioning and matrix-free implementation.

Key words. nonlinear systems, Krylov subspaces, inexact Newton methods, tensor methods, generalized minimal residual methods, singularity

AMS subject classification. 65H10

PII. S105262349527646X

1. Introduction.

This paper introduces a tensor-Krylov method for solving the nonlinear equations problem

$$\text{given } F : \Re^N \to \Re^N, \text{ find } x_* \in \Re^N \text{ such that } F(x_*) = 0. \tag{1.1}$$

Standard methods (such as Newton's method) widely used in practice for solving (1.1) are iterative methods which base each iteration upon a linear model $M(x)$ of $F(x)$ around the current point x_c:

$$M(x_c + d) = F(x_c) + J_c d, \tag{1.2}$$

where $d \in \Re^N$ and $J_c \in \Re^{N \times N}$ is either the current Jacobian matrix or an approximation. When J_c is very large, (1.1) is often (inexactly) solved via a Krylov method, such as the GMRES method, which does not require the factorization of J_c. The distinct advantage of Krylov methods is their minimum storage requirement. Newton–Krylov schemes are considered by many authors, including Brown and Saad [6, 7], Chan and Jackson [8], and Brown and Hindmarsh [5]. Their computational results show that these methods can be quite effective for many classes of problems in the context of systems of partial differential equations or ordinary differential equations.

The distinguishing feature of the Newton-equation–based algorithm is that if $F'(x_c)$ is Lipschitz continuous in a neighborhood containing the root x_*, $F'(x_*)$ is nonsingular and (1.2) is solved exactly (or to certain accuracy; e.g., see [7] for more details), then the sequence of iterates produced converges locally and q-quadratically to x_*. This means eventual fast convergence in practice. If $F'(x_*)$ is singular, however,

* Received by the editors August 28, 1995; accepted for publication (in revised form) March 1, 1996.

http://www.siam.org/journals/siopt/7-3/27646.html

† Cadence Design Systems, Inc., 555 River Oaks Parkway, Building 3, MS 3B1, San Jose, CA 95134 (feng@cadence.com). The research of this author was supported by NASA under contract NAS 2-13721 between NASA and the Universities Space Research Association (USRA).

‡ Fluid Dynamics Division, Mail Stop T27B-1, NASA Ames Research Center, Moffett Field, CA 94035-1000 (pulliam@nas.nasa.gov).

then the Newton-equation–based methods usually do not have rapid local convergence. This situation is analyzed and acceleration techniques are suggested by many authors, including Reddien [28], Decker and Kelley [10], [11], [12], Decker, Keller, and Kelley [9], Kelley and Suresh [21], Griewank and Osborne [19], and Griewank [18]. Recent work by Kelley and Xue [22] discusses inexact Newton methods for singular problems. In summary, their papers show that when the Jacobian matrix at the solution has a nontrivial null space, the Newton-equation–based methods have at best a linear rate of convergence. Acceleration techniques presented in those papers depend upon a priori knowledge that the problem is singular.

Tensor methods for nonlinear equations introduced by Schnabel and Frank [31] are intended to be efficient both for nonsingular problems and for problems with low rank deficiency. However, these methods rely on the factorization of the Jacobian matrix, which makes them unsuitable for many large systems of nonlinear equations. The goal of this paper is to develop a tensor method (referred to as the tensor-GMRES) for large systems of nonlinear equations using Krylov subspace projection techniques. This method is independent of matrix factorization and can have efficient matrix-free implementations. In addition, it is intended to inherit the advantage of traditional tensor methods over the standard Newton's method for both singular and nonsingular problems.

Tensor-Krylov methods were first considered by Bouaricha in his Ph.D. thesis [2]. The basic idea is to solve the tensor model by calling a linear Krylov method twice in each tensor iteration. Although the second call of the Krylov method could be less expensive (due to a possible good initial guess) close to the solution, the computational cost of one iteration of tensor methods based on this idea is likely to be significantly more expensive than one iteration of the analogous Newton–Krylov (e.g., see [3]). This difficulty could make these tensor methods noncompetitive with their Newton counterpart in many situations.

The new tensor-GMRES method introduced in this paper requires no more function and derivative evaluations (and hardly more storage or arithmetic per iteration) than the analogous Newton-GMRES method. This is achieved by formulating the tensor term in a more restricted form than that of a traditional tensor model. The restriction is observed to have minimal impact on the performance of the tensor method. Comparative test results show that the tensor-GMRES method is more efficient than an analogous Newton-GMRES method, particularly on problems where the Jacobian matrix provides insufficient information.

We would like to introduce some notation that will be used later on in this paper. The solution to the system is represented by x_*, and a current iterate is represented by x_c or x_k. Consistent with tradition, we denote $F'(x)$ by $J(x)$ and usually abbreviate $J(x_c)$, $J(x_*)$ as J_c , J_*, respectively. Similarly, $F(x_c)$, $F(x_*)$, $F''(x_c)$, and $F''(x_*)$ are often abbreviated as F_c , F_* , F''_c , and F''_*. The notation $\|\cdot\|$ denotes the Euclidean vector norm. We use N to denote the length of x, which is also the number of variables (equations) in the system.

This paper is organized as follows. Section 2 reviews the GMRES algorithm and a line search Newton-GMRES algorithm. Traditional tensor methods for nonlinear equations are briefly reviewed in section 3. The core of this paper is section 4, which introduces the formation and solution of the new tensor-GMRES model. Implementation of the tensor-GMRES algorithm is given in section 5. Test results comparing the tensor-GMRES method versus the analogous Newton-GMRES method are also reported in this section. Finally, in section 6, we summarize our research and make some brief comments on areas for future related research.

2. Newton-GMRES method for nonlinear equations.

The GMRES (Generalized Minimal RESidual) method was introduced by Saad and Schultz [30] for solving large asymmetric systems of linear equations. This method is very effective when coupled with preconditioning techniques. It is also very competitive compared to other iterative methods in many applications.

Given a matrix $A \in \Re^{N \times N}$, a vector $v_1 \in \Re^N$, and an integer $m \geq 1$, the Krylov subspace associated with A, v_1, and m is defined as

$$K_m(A, v_1) = \text{span}\{v_1, Av_1, A^2 v_1, \ldots, A^{m-1} v_1\}.$$

Consider a system of linear equations $Ax = b$. Given an initial guess x_0, the initial residual is defined as $r_0 = Ax_0 - b$. The GMRES method attempts to find $z_m \in K_m(A, r_0)$ such that the residual vector $A(x_0 + z_m) - b$ has minimal norm. The Gram–Schmidt method is used to compute an l_2-orthonormal basis $\{v_1, v_2, \ldots, v_m\}$ of $K_m(A, v)$. (In practice, a modified Gram–Schmidt is often used instead.)

Algorithm G. GMRES.

(G-1) Start. Choose x_0 and compute $r_0 = b - Ax_0$ and $v_1 = r_0/\|r_0\|$.

(G-2) Iterate. For $j = 1, 2, \ldots, m, \ldots$ until satisfied do:

$$\begin{aligned} h_{i,j} &= (Av_j, v_i), i = 1, 2, \ldots, j, \\ \hat{v}_{j+1} &= Av_j - \sum_{i=1}^{j} h_{i,j} v_i, \\ h_{j+1,j} &= \|\hat{v}_{j+1}\|, \\ v_{j+1} &= \hat{v}_{j+1}/h_{j+1,j}. \end{aligned}$$

(G-3) Form the approximate solution:
$x_m = x_0 + V_m y_m$ where y_m minimizes $\|\beta e_1 - \bar{H}_m y_m\|$, $y \in \Re^m$.

As consequences of m iterations of (G-2) (assume it does not break down; i.e., $\|\hat{v}_{j+1}\|$ does not vanish throughout), we have $m+1$ orthonormal vectors $v_1, \ldots, v_{m+1}$, and an $(m+1) \times m$ Hessenberg matrix $\bar{H}_m$ whose nonzero entries are given by the $h_{i,j}$ produced by the algorithm. Let $V_m = [v_1, \ldots, v_m]$. An important relation $AV_m = V_{m+1}\bar{H}_m$ holds after each iteration of (G-2).

The GMRES scheme is based on solving the following least squares problem:

$$\min_{z_m \in K_m} \|b - A(x_0 + z_m)\| = \min_{z_m \in K_m} \|r_0 - Az_m\|, \tag{2.1}$$

where $r_0 = b - Ax_0$. If we set $z_m = V_m y$, $v_1 = r_0/\|r_0\|$, and $\beta = \|r_0\|$, this is equivalent to solving

$$\begin{aligned} \min_{y \in \Re^m} \|\beta v_1 - AV_m y\| &= \min_{y \in \Re^m} \|V_{m+1}(\beta e_1 - \bar{H}_m y)\| \\ &= \min_{y \in \Re^m} \|\beta e_1 - \bar{H}_m y\|. \end{aligned} \tag{2.2}$$

The least squares problem (2.2) is solved via a QR factorization of $\bar{H}_m$, which is fairly inexpensive because of the Hessenberg form of $\bar{H}_m$. When m is small, the cost of solving (2.2) is minimal.

Due to memory limitations, it is necessary to restrict the number of iterations taken by the Arnoldi process in (G-2). This leads to restarted versions of GMRES.

The idea is to use the GMRES iteratively by restarting the algorithm every m steps, where m is some fixed integer parameter. Using the GMRES as a linear solver, one can obtain a Newton-GMRES algorithm for nonlinear equations. At each iteration of the nonlinear algorithm, a solution (or an approximate solution) is sought to the linear system

$$J_c d = -F_c, \tag{2.3}$$

with J_c being the current Jacobian matrix and F_c being the current function value. The Newton-GMRES method is an inexact Newton method in the sense that at each iteration, a Newton-like step is obtained by solving the Newton equation approximately instead of exactly for a step d such that $\|F_c + J_c d\| < \|F_c\|$. (Inexact Newton methods originated from the work of Dembo, Eisenstat, and Steihaug [13].) A step obtained in this way is a descent direction for $\frac{1}{2}\|F(x)\|^2$ (see [7] and [15] for details). A global convergence strategy such as backtracking line search is employed to determine the step length along this descent direction, which will force progress towards the solution.

Algorithm NG. An iteration of the Newton-GMRES.
Given x_k, $J_k \in \Re^{N\times N}$ and $F_k \in \Re^N$.
(NG-1) Choose $\epsilon_k \in [0, 1)$, the tolerance for the Newton equation step.
(NG-2) Do GMRES (restart if necessary) to find $d^n = d_0 + V_m y^n$ such that

$$F_k + J_k d^n = r_k, \text{ with } \|r_k\|/\|F_k\| \le \epsilon_k,$$

where d_0 is the initial guess to the solution of the Newton equation, and the columns of V_m form an orthonormal basis for the Krylov space generated by the Arnoldi process.
(NG-3) Find $\lambda > 0$ using a backtracking line search global strategy and form the next iterate $x_{k+1} = x_k + \lambda d^n$.

The residual vector r_k is the amount by which d^n fails to satisfy the Newton equation $J_k d + F_k = 0$. The forcing sequence ϵ_k is used to control the level of accuracy. The seminal local convergence analysis for inexact Newton methods was given by Dembo, Eisenstat, and Steihaug [13]. Their theory implies that if the sequence $\epsilon_k \to 0$, then under certain conditions (such as the Jacobian matrix being nonsingular at the solution) the iterates generated by Algorithm NG converge to the solution superlinearly; the convergence is quadratic if $\epsilon_k = O(\|F_k\|)$. This means eventual fast convergence in practice for nonsingular problems.

An attractive feature of Newton–Krylov algorithms is that the explicit computation of the Jacobian matrix is not necessary. This is owing to the fact that the only computation involving the Jacobian matrix is the product of the Jacobian matrix and a vector, which can be approximated by finite difference

$$J(x)v \approx \frac{F(x + \sigma v) - F(x)}{\sigma}, \tag{2.4}$$

with an appropriately chosen value of σ (see, e.g., [14] for details).

3. Traditional tensor methods for nonlinear equations.

The tensor model for nonlinear equations introduced by Schnabel and Frank [31] is a quadratic model $M(x)$ of $F(x)$ formed by adding a second order term to the linear Taylor series model, giving

$$M_T(x_k + d) = F(x_k) + F'(x_k)d + \frac{1}{2}T_k dd, \tag{3.1}$$

where $T_k \in \Re^{n\times n\times n}$ is intended to supply second order information about $F(x)$ around x_k, without appreciably increasing the cost of forming, storing, or solving the model. Schnabel and Frank [31] form T_k by requiring the tensor model to interpolate the function values at a very small number, p, of past iterates. This requires no additional function or derivative evaluations. By choosing the smallest T_k in the Frobenuis norm that meets these conditions, T_k has rank p, the term $T_k dd$ has a simple form, and the cost of forming and storing T_k is very small.

Since (3.1) may or may not have a root, the goal is to solve

$$\min_{d\in\Re^n} \|M_T(x_k+d)\|_2.$$

Schnabel and Frank [31] show that when the factorization of the Jacobian matrix is available, this can be done nearly as efficiently as solving the standard linear model, usually by solving p quadratic equations in p unknowns and $n-p$ linear equations in $n-p$ unknowns. If $F'(x_k)$ is large and sparse, the tensor model solution could still cost very little more than the standard sparse Newton iteration; see [2].

Computational results in [31] and [2] show that the tensor method is more efficient than an analogous standard method based upon Newton's method on both nonsingular and singular problems, with a particularly large advantage on singular problems. In tests in [31] on a standard set of nonsingular test problems, the tensor method is almost always more efficient than the standard method and is never significantly less efficient, with an average improvement over 20% for all problems and over 35% for harder problems. The average improvement in iterations and function evaluations for singular problems is larger, generally in the range of 30% to 65%. More recent computational experiments in [2], including experiments on much larger problems, show similar advantages for tensor methods.

Furthermore, tensor methods have theoretical advantages over standard methods. It is shown in [16] that under mild conditions, tensor methods have local superlinear convergence for a large class of singular problems. In the same situation, standard methods only have linear convergence. The analysis in [16] also confirms that tensor methods converge at least quadratically on problems where the Jacobian matrix at the root is nonsingular.

4. Tensor-GMRES method for nonlinear equations.

4.1. Introduction. The tensor method introduced here is primarily intended to improve upon the Newton–Krylov methods in cases where the Jacobian matrix is singular or ill conditioned at the solution. The basic idea is to first compute the Newton-GMRES step. Then the Krylov subspace which resulted from the Newton-GMRES step computation is utilized to form a tensor model, which is subsequently solved to give a tensor step.

The tensor model considered here only uses information from one past iterate. There are three reasons for this. First, tensor methods that use one past point are easier to implement and have satisfactory computational performance in practice. Second, tensor methods based on a single past point are theoretically better understood. Third and most importantly, using more past points may require significantly more storage.

The major difference between the new tensor model and the traditional tensor model is that the new model has a more restricted second order term. The analysis of tensor methods for nonlinear equations by Feng, Frank, and Schnabel [16] indicates that tensor methods will not lose fast local convergence on singular problems if the tensor term is projected into proper subspaces. As a matter of fact, we can show that for problems where the Jacobian matrix has rank deficiency one at the solution, if the second order term in the tensor model is projected into the subspace spanned by the left singular vector corresponding the smallest singular value of the Jacobian matrix, the theoretical results given in [16] remain intact. This is the theoretical foundation of our tensor-GMRES method. The idea of projected tensor methods was first implemented in [17] for constrained optimization, where the projection took place in the variable space. The difference here is that the projection occurs in the function space.

Some notation is also useful to us. Let $F'(x_c) = U_c D_c V_c^T$ be the singular value decomposition of $F'(x)$ at x_c, where $U_c = [u_1^c, u_2^c, \ldots, u_N^c]$, $V_c = [v_1^c, v_2^c, \ldots, v_N^c]$, and $D_c = \mathrm{diag}[\sigma_1^c, \sigma_2^c, \ldots, \sigma_N^c]$, with $\sigma_1^c \geq \sigma_2^c \geq \cdots \geq \sigma_N^c \geq 0$ being the singular values of $F'(x_c)$ and $\{u_i^c\}$, $\{v_i^c\}$ being the corresponding left and right singular vectors. Similarly, let $F'(x_*) = UDV^T$. Let v and u be the right and left singular vector of $F'(x_*)$ corresponding to the zero singular value, when $F'(x_*)$ has rank deficiency one.

4.2. Analysis of an ideal tensor method. The analysis in this section is an extension of the analysis of tensor methods for nonlinear equations by Feng, Frank, and Schnabel [16]. We will refer to some of their lemmas and theorems, and sometimes parts of their proofs. For the sake of brevity, we do not reiterate their results here. See [16] for details.

The sequence of iterates produced by the algorithm analyzed is invariant to translations in the variable space. Thus, no generality is lost by making the assumption that the solution occurs at $x_* = 0$ (which is consistent with [16]), and this assumption is made throughout this section. We also assume that v_N^c and u_N^c are chosen so that $\|v_N^c - v\| = O(\|x_c\|)$ and $\|u_N^c - u\| = O(\|x_c\|)$, whenever x_c is sufficiently close to x_*. This assumption is valid from the theorems about continuity of eigenvectors in Ortega [25] and Stewart [32], as long as $F'(x)$ is continuous near x_* and has rank deficiency no greater than one at x_*.

Before going into the details of the analysis, we give Assumption 4.0. It basically states that near x_*, the second order term supplies useful information in the null space direction of $F'(x_*)$, where $F'(x_*)$ lacks information.

Assumption 4.0. Let $F : \Re^N \to \Re^N$ have two Lipschitz continuous derivatives. Let $F(x_*) = 0, F'(x_*)$ be singular with only one zero singular value, and let u and v be the left and right singular vectors of $F'(x_*)$ corresponding to the zero singular value. Then we assume that

$$u^T F''(x_*) vv \neq 0, \tag{4.1}$$

where $F''(x_*) \in \Re^{N \times N \times N}$.

Assumption 4.0 is satisfied by most problems with $\mathrm{rank}(F'(x_*)) = N - 1$ and has been assumed in most papers that analyze the behavior of Newton's method on singular problems. When $N = 1$, Assumption 4.0 is equivalent to $F''(x_*) \neq 0$.

Suppose we know the right and left singular vectors v_N^c and u_N^c corresponding to the least singular value of $F'(x_c)$, where x_c is the current iterate and $\|v_N^c\| = \|u_N^c\| = 1$. Let $W \in \Re^{N \times m}$ with $m \leq N$ orthonormal columns. Consider an ideal tensor model

$$M_{T_W}(x_c + d) = F(x_c) + F'(x_c)d + \frac{1}{2}(WW^T)a_c(v_N^{cT} d)^2, \tag{4.2}$$

where u_N^c is in the span of the column vectors of W, i.e., $u_N^c = Wy$ for some $y \in \Re^m \neq 0$, and $a_c = F''(x_c)v_N^c v_N^c$. The model is an excellent model of $F(x)$ at x_c because it contains the correct second order information where the Jacobian contains the least information and, correspondingly, where the second order term has the greatest influence. Based on (4.2), a simple tensor algorithm, Algorithm PT, is designed.

Algorithm PT. Projected Tensor Algorithm.

```
IF (4.2) has real roots THEN
        $d \leftarrow d_R$ where $d_R$ solves $M_{T_W}(x_c + d) = 0$
ELSE $d \leftarrow d_M$ where $d_M$ minimizes $\|M_{T_W}(x_c + d)\|$
```

Since (4.2) is the basis for the new tensor-GMRES model, an analysis of Algorithm PT is given. The tensor model specified in (4.2) is closely related to the ideal tensor model analyzed by Feng, Frank, and Schnabel [16]. The only difference is that their ideal model does not have a projection matrix WW^T in front of a_c. It can be shown that this difference does not change their results.

Corollary 4.1. *Let Assumption* 4.0 *hold and let* $\{x_k\}$ *be the sequence of iterates produced by Algorithm* PT. *There exist constants* K_1, K_2 *such that if* $\|x_0\| \leq K_1$, *then the sequence* $\{x_k\}$ *converges to* x_* *and* $\|x_{k+2}\| \leq K_2 \|x_k\|^{\frac{3}{2}}$ *for* $k = 0, 1, 2, \ldots.$

Proof. Since $u_N^c = Wy$, from the orthogonality of columns of W, we have

$$u_N^c {u_N^c}^T W W^T = u_N^c y^T W^T W W^T = u_N^c y^T W^T = u_N^c {u_N^c}^T. \tag{4.3}$$

Using (4.3),

$$\begin{aligned} & M_{T_W}(x_c + d) \\ &= \left(\sum_{i=1}^{N} u_i^c {u_i^c}^T \right) \left(F_c + \sum_{i=1}^{N} \sigma_i^c u_i^c {v_i^c}^T d + \tfrac{1}{2} (W W^T) a_c ({v_N^c}^T d)^2 \right) \\ &= \left[\sum_{i=1}^{N-1} (\sigma_i^c {v_i^c}^T d + {u_i^c}^T F_c + \tfrac{1}{2} {u_i^c}^T (W W^T) a_c ({v_N^c}^T d)^2) u_i^c \right] \\ & \quad + (\sigma_N^c {v_N^c}^T d + {u_N^c}^T F_c + \tfrac{1}{2} {u_N^c}^T a_c ({v_N^c}^T d)^2) u_N^c. \end{aligned} \tag{4.4}$$

Note that the difference between (4.4) and (4.2) of [16] is only a second order term in the coefficient of each u_i^c for $i = 1, \ldots, N-1$, which does not affect either of the proofs of Lemmas 4.1 and 4.2 of [16]. The rest of the proof can be completed by following exactly the proof of Theorem 4.4 of [16]. □

4.3. Formation of the tensor model. The first stage of the tensor-GMRES algorithm is to compute a Newton-GMRES step. The resulting Krylov subspace information from the Newton-GMRES step calculation is then utilized to form and solve a tensor-GMRES model. At the current iterate x_c, an ideal situation is that

$$F_c + J_c d^n = 0 \tag{4.5}$$

is solved exactly by the GMRES method with $d^n = V_m y_m + d_0$ (starting from d_0). An interesting fact is that the resulting Newton step d^n is in the span of $\{V_m, d_0\}$. The analysis of Feng, Frank, and Schnabel [16] indicates that when the Jacobian matrix has a null space of dimension one at the solution, the Newton iterates fall into a funnel around the null space close to the solution. In this situation, their theory also implies that the angle between d^n and v_N^c, the right singular vector corresponding to the smallest singular value of the current Jacobian matrix J_c, will be arbitrarily close to zero, close to the solution. As a consequence of $d^n \in \{V_m, d_0\}$, v_N^c will be arbitrarily close to being in the span of $\{V_m, d_0\}$. Hence u_N^c, in the same direction as $J_c v_N^c$, will be arbitrarily close to being in the span of $J_c[V_m\ d_0]$. Therefore, a good approximate to the projection matrix WW^T in (4.2) would be the projection matrix

$$P = Y(Y^TY)^{-1}Y^T, \quad \text{where} \quad Y = J_c[V_m\ d_0]. \tag{4.6}$$

In practice, a Newton-GMRES step is usually required to give sufficient reduction in the residual norm of (4.5), rather than solve the equation exactly. Nevertheless, for a sufficient small residual norm, P would still be a reasonable and useful approximation to WW^T. This is confirmed by test results given in the next section.

The singular vectors and the exact second order derivative used in (4.2) are normally too expensive to obtain. We approximate them in the following manner. As in the situation of the traditional tensor model, let $s_c = x_p - x_c$, the difference between the past iterate x_p and the current iterate x_c. There are two choices for approximating v_N^c in (4.2); one choice uses $d^n/\|d^n\|$, since the Newton step d^n is likely to be along the null space close to the solution for singular problems. Another choice uses $h = s_c/\|s_c\|$, since the difference between the two consecutive iterates is also likely to be along the null space when consecutive iterates are in the funnel around the null space near the solution. We choose to use h because, as we will see later, this will cause our tensor model to interpolate a past point in a projected space. The ability to interpolate past points is vital for the success of traditional tensor methods. Close to the solution, the term $a_c = F''(x_c)v_N^c v_N^c$ in (4.2) is likely to be approached by

$$\bar{a}_c = \frac{2(F(x_p) - F(x_c) - J(x_c)s_c)}{s_c^T s_c} = F''(x_c)hh + E, \tag{4.7}$$

where $\|E\| = O(\|s_c\|)$ (see [16]). Equation (4.7) is standard in a tensor model formulation, which requires no extra function or derivative evaluations.

Putting all the pieces together, we arrive at the following tensor model:

$$M_{T_P}(x_c + d) = F_c + J_c d + \tfrac{1}{2}P\bar{a}_c(h^T d)^2, \tag{4.8}$$

where P is given by (4.6). It is easy to verify that the unprojected tensor model

$$M_T(x_c + d) = F_c + J_c d + \tfrac{1}{2}\bar{a}_c(h^T d)^2 \tag{4.9}$$

interpolates the function value at the past point x_p. Hence,

$$\begin{aligned} PM_{T_P}(x_c + d) &= PF_c + PJ_c d + \tfrac{1}{2}PP\bar{a}_c(h^T d)^2 \\ &= PF_c + PJ_c d + \tfrac{1}{2}P\bar{a}_c(h^T d)^2 \\ &= P(F_c + J_c d + \tfrac{1}{2}\bar{a}_c(h^T d)^2) \end{aligned}$$

implies that the interpolation property holds in a projected space. A second property of (4.8) is that when $m = N$ and J_c is nonsingular, the projector matrix P is equal to identity, which recovers the full tensor model (4.9).

4.4. Solution of the tensor model. Solving the tensor model (4.8) in the full variable space is not preferable since it could be as expensive as solving the full tensor model. An alternative is to solve (4.8) along a subspace that spans the Newton step direction. Despite tending to undershoot (or overshoot) when the first order information is lacking, the Newton step usually gives good directional information. Therefore, we would like to solve the least squares problem

$$\min_{d \in \{d_0\} \cup K_m} \|F_c + J_c d + \tfrac{1}{2}P\bar{a}_c(h^T d)^2\|. \tag{4.10}$$

Recall that $J_cV_m = V_{m+1}\bar{H}_m$. Let $\bar{H}_m = \bar{Q}_m\bar{R}_m$ be the QR factorization of $\bar{H}_m$. (Note that $\bar{Q}_m$ is the product of m Givens rotations; for details see [30].) Let $d = V_my + d_0\tau$. Using $r_0 = -F_c - J_cd_0$, (4.10) is equivalent to solving

$$\min_{y\in\Re^m,\tau\in\Re}\left\| F_c + J_c[V_m, d_0]\begin{pmatrix} y \\ \tau \end{pmatrix} + \tfrac{1}{2}P\bar{a}_c\left\{h^T[V_m, d_0]\begin{pmatrix} y \\ \tau \end{pmatrix}\right\}^2\right\|$$

$$(4.11)\quad = \min_{y\in\Re^m,\tau\in\Re}\left\| F_c + [V_{m+1}\bar{H}_m, g]\begin{pmatrix} y \\ \tau \end{pmatrix} + \tfrac{1}{2}P\bar{a}_c\left\{h^T[V_m, d_0]\begin{pmatrix} y \\ \tau \end{pmatrix}\right\}^2\right\|,$$

where $g = -F_c - r_0$. An important feature of (4.11) is that it does not involve the Jacobian matrix J_c. We simplify (4.11) to

$$(4.12)\qquad \min_{\hat{y}\in\Re^{m+1}} \|F_c + \hat{J}\hat{y} + \tfrac{1}{2}\hat{a}_c(\hat{s}^T\hat{y})^2\|,$$

where $\hat{J} = [V_{m+1}\bar{H}_m, g]$, $\hat{y} = [y^T\ \tau]^T$, $\hat{a}_c = P\bar{a}_c$, and $\hat{s} = [V_m, d_0]^Th$. The solution of this type of nonlinear least squares problem is studied by Bouaricha and Schnabel in [4]. Their theory shows that when $\hat{J}$ has full rank, the solution of (4.12) is given by

$$(4.13)\qquad \hat{y}_* = (\hat{J}^T\hat{J})^{-1}\hat{s}\ q(\beta_*)/\omega - (\hat{J}^T\hat{J})^{-1}\hat{J}^T(F_c + \tfrac{1}{2}\hat{a}_c\beta_*^2),$$

where $q(\beta)$ and ω are defined by

$$(4.14)\qquad q(\beta) = \hat{s}^T(\hat{J}^T\hat{J})^{-1}\hat{J}^TF_c + \beta + \tfrac{1}{2}\hat{s}^T(\hat{J}^T\hat{J})^{-1}\hat{J}^T\hat{a}_c\beta^2,$$

$$(4.15)\qquad \omega = \hat{s}(\hat{J}^T\hat{J})^{-1}\hat{s},$$

and the value of β_* is determined from

$$(4.16)\qquad \min_{\beta\in\Re} \|q(\beta)/\sqrt{\omega}\|^2 + \|n(\beta)\|^2,$$

where $\|n(\beta)\|^2 = \|(F_c - PF_c) + \frac{1}{2}(\hat{a}_c - P\hat{a}_c)\beta^2\|^2$.

In our situation, since P is a projector matrix and

$$\hat{a}_c - P\hat{a}_c = P\bar{a}_c - PP\bar{a}_c = P\bar{a}_c - P\bar{a}_c = 0,$$

$n(\beta)$ is a constant function. Hence, the minimization problem (4.16) is equivalent to

$$(4.17)\qquad \min_{\beta\in\Re} \|q(\beta)\|.$$

To obtain $\hat{y}_*$, the critical computational work comes from the factorization of $\hat{J}^T\hat{J}$, since all the computations involving $(\hat{J}^T\hat{J})^{-1}$ can be achieved through backsolves when this factorization is available. For this reason, we discuss the factorization of $\hat{J}^T\hat{J}$. Recall that $\hat{J} = [V_{m+1}\bar{H}_m, g]$ and $\bar{H}_m = \bar{Q}_m\bar{R}_m$. Hence, we have

$$\begin{aligned}
\hat{J}^T\hat{J} &= [V_{m+1}\bar{H}_m, g]^T[V_{m+1}\bar{H}_m, g] \\
&= \begin{pmatrix} \bar{H}_m^T\bar{H}_m & \bar{H}_m^TV_{m+1}^Tg \\ g^TV_{m+1}\bar{H}_m & g^Tg \end{pmatrix} \\
&= \begin{pmatrix} \bar{R}_m^T\bar{R}_m & \bar{H}_m^TV_{m+1}^Tg \\ g^TV_{m+1}\bar{H}_m & g^Tg \end{pmatrix} \\
&= \begin{pmatrix} \bar{R_m^1}^T & 0 \\ w^T & \gamma \end{pmatrix}\begin{pmatrix} \bar{R}_m^1 & w \\ 0 & \gamma \end{pmatrix},
\end{aligned}$$

where $\bar{R}_m^1$ is the first m rows of $\bar{R}_m$, $w = \bar{Q}_m^T V_{m+1}^T g$, and $\gamma = \sqrt{g^T g - w^T w}$. The factorization is possible since $\hat{J}^T \hat{J}$ is always at least positive semidefinite. After $\hat{y}_*$ is obtained, (4.10) is solved by $d^t = [V_m, d_0]\hat{y}_*$.

However, the calculation of expressions involving $(\hat{J}^T\hat{J})^{-1}$ is impossible if $\gamma = 0$; i.e., $\hat{J}$ is rank deficient. We discuss how to overcome this difficulty. Since $\hat{J} = [V_{m+1}\bar{H}_m, g]$ and $V_{m+1}\bar{H}_m$ has full rank, $\hat{J}$ being singular implies that g has to be in the span of $\{V_{m+1}\bar{H}_m\}$, which implies that $J_c d_0 = g$ is in the span of $J_c V_m = V_{m+1}\bar{H}_m$. When J_c has full rank, this implies that d_0 is in the span of $\{V_m\}$, which in turn implies that $d^n = d_0 + V_m y^n$ is in the span of V_m. In this situation, based on previous discussions, we actually would like to solve the tensor model (4.8) in the Krylov subspace V_m only, i.e.,

$$\min_{z \in K_m} \|F_c + J_c(d_0 + z) + \tfrac{1}{2}\bar{P}\bar{a}_c(h^T(d_0 + z))^2\|, \tag{4.18}$$

where $\bar{P} = \bar{Y}(\bar{Y}^T\bar{Y})^{-1}\bar{Y}^T$ with $\bar{Y} = J_c V_m$, which is equivalent to solving

$$\min_{y \in \Re^m} \|F_c + J_c d_0 + J_c V_m y + \tfrac{1}{2}\bar{P}\bar{a}_c(h^T(d_0 + V_m y))^2\|. \tag{4.19}$$

Note that

$$\begin{aligned} \bar{P} &= V_{m+1}\bar{Q}_m\bar{R}_m[(V_{m+1}\bar{Q}_m\bar{R}_m)^T(V_{m+1}\bar{Q}_m\bar{R}_m)]^{-1}(V_{m+1}\bar{Q}_m\bar{R}_m)^T \\ &= V_{m+1}\bar{Q}_m\bar{R}_m(\bar{R}_m^T\bar{R}_m)^{-1}\bar{R}_m^T\bar{Q}_m^T V_{m+1}^T \\ &= V_{m+1}\bar{Q}_m\begin{pmatrix} I_m & 0 \\ 0 & 0 \end{pmatrix}\bar{Q}_m^T V_{m+1}^T. \end{aligned} \tag{4.20}$$

Using $J_c V_m = V_{m+1}\bar{H}_m$, $\bar{H}_m = \bar{Q}_m\bar{R}_m$, $r_0 = -F_c - J_c d_0$, (4.20), and by letting b be the first m components of $\bar{Q}_m^T V_{m+1}^T \bar{a}_c$, (4.19) is equivalent to

$$\begin{aligned} &\min_{y \in \Re^m} \left\| -r_0 + V_{m+1}\bar{H}_m y + \tfrac{1}{2}V_{m+1}\bar{Q}_m\begin{pmatrix} b \\ 0 \end{pmatrix}(h^T(d_0 + V_m y))^2 \right\| \\ &= \min_{y \in \Re^m} \left\| V_{m+1}\left(\|r_0\|e_1 - \bar{Q}_m\bar{R}_m y - \tfrac{1}{2}\bar{Q}_m\begin{pmatrix} b \\ 0 \end{pmatrix}(h^T(d_0 + V_m y))^2\right) \right\| \\ &= \min_{y \in \Re^m} \left\| \bar{Q}_m\left(\bar{Q}_m^T\|r_0\|e_1 - \bar{R}_m y - \tfrac{1}{2}\begin{pmatrix} b \\ 0 \end{pmatrix}(h^T(d_0 + V_m y))^2\right) \right\| \\ &= \min_{y \in \Re^m} \|w - \bar{R}_m^1 y - b(h^T(d_0 + V_m y))^2\| + |\tau|, \end{aligned} \tag{4.21}$$

where $[m^T\ \tau]^T = \bar{Q}_m^T\|r_0\|e_1$ and $\bar{R}_m^1$ is the first m rows of $\bar{R}_m$. Again, using the techniques for solving the tensor model of nonlinear least squares developed in [4], we form the β function

$$\begin{aligned} q(\beta) &= h^T V_m(\bar{R}_m^1)^{-1}w - h^T V_m y - \tfrac{1}{2}h^T V_m(\bar{R}_m^1)^{-1}b(h^T(d_0 + V_m y))^2 \\ &= \hat{h}^T(\bar{R}_m^1)^{-1}w + h^T d_0 - \beta - \tfrac{1}{2}\hat{h}^T(\bar{R}_m^1)^{-1}b\beta^2, \end{aligned} \tag{4.22}$$

where $\beta = h^T(d_0 + V_m y)$ and $\hat{h} = V_m^T h$, and solve the minimization problem

$$\min_{\beta \in \Re} \|q(\beta)\|. \tag{4.23}$$

Let β_* be a solution to (4.23). By the theory established in [4], (4.21) is solved by

$$y_*^t = [(\bar{R}_m^1)^T(\bar{R}_m^1)]^{-1}\hat{h}q(\beta_*)/\omega + (\bar{R}_m^1)^{-1}w - \tfrac{1}{2}(\bar{R}_m^1)^{-1}b\beta_*^2.$$

Then the tensor step for the system of nonlinear equations is given by

$$d^t = d_0 + V_m y_*^t. \tag{4.24}$$

4.5. Preconditioning and matrix-free implementation. The success of the GMRES method on a system of linear equations usually depends on a good preconditioner. The formation and solution of the tensor model is consistent with preconditioning. When a preconditioner M is used in solving the Newton equation by the GMRES algorithm, $\bar{a}_c$ is replaced by $M^{-1}\bar{a}_c$, and F_c is replaced by $M^{-1}F_c$ for left preconditioning (or s by $M^{-1}s$ for right preconditioning). The rest of the solution procedure is unchanged.

Compared to the Newton-GMRES method, the only extra computation involving the Jacobian matrix in the tensor-GMRES method is the computation of Js in the formation of the tensor term. In a Jacobian-free implementation, this matrix-vector product can be approximated by the finite difference formula specified by (2.4). Hence, the tensor-GMRES scheme is consistent with matrix-free implementation.

4.6. Work comparison. If m steps of GMRES are required to solve the Newton equation, in addition to m Jacobian-vector products, the Newton-GMRES iteration costs $m(m+2)N$ multiplications and requires storage of $m+1$ N-vectors. The extra storage required by the tensor-GMRES method is two N-vectors.

The extra computational cost of forming the tensor model is N multiplications. Compared to the solution of the Newton equation by the GMRES algorithm in a similar situation, the solution of the tensor model requires a minimal amount of extra work. The major extra work comes from forming d^t, $\hat{s}$, $\hat{J}^T\hat{a}_c$, and $\hat{J}^T F_c$, each requiring $(m+1)N$ multiplications (note that $\hat{J}^T\hat{a}_c = \hat{J}^T P\bar{a}_c = \hat{J}^T\bar{a}_c$ from the definition of P). Since $V_{m+1}^T g = V_{m+1}^T(-F_c - r_0) = -V_{m+1}^T F_c - \|r_0\|e_1$, given $V_{m+1}^T F_c$, the cost of $V_{m+1}^T g$ is only a single subtraction. Hence, the extra cost of factorization of $\hat{J}^T\hat{J}$, which involves the calculation of $\bar{Q}_m^T V_{m+1}^T g$, $g^T g$ and $w^T w$, is $N+5m$ multiplications. The operation count is accumulated from an application of m Givens rotations costing $4m$ multiplications, a dot-product of two N-vectors costing N multiplications, and a dot-product of two m-vectors costing m multiplications. Given $\hat{s}$, $\hat{J}^T\hat{a}_c$, and $\hat{J}^T F_c$, the major cost of forming $q(\beta)$ and ω defined in (4.14) and (4.15), respectively, comes from the calculation of $\hat{s}^T(\hat{J}^T\hat{J})^{-1}$. Using the available factorization of $\hat{J}^T\hat{J}$, this can be done by two backsolves of $(m+1)\times(m+1)$ triangular systems, which costs $(m+1)^2$ multiplications. After $\hat{s}^T(\hat{J}^T\hat{J})^{-1}$ is obtained, the cost of forming $q(\beta)$ and ω is three dot-products of two m-vectors costing $3m$ multiplications. The cost for obtaining $\hat{y}_*$ using (4.13) needs two extra backsolves of $(m+1)\times(m+1)$ triangular systems, which costs $(m+1)^2$ multiplications, given $\hat{s}^T(\hat{J}^T\hat{J})^{-1}$, $\hat{J}^T\hat{a}_c$, and $\hat{J}^T F_c$. In summary, the total extra work required by solving the tensor model in the worse situation is at most $(4(m+1)+1)N + 2(m+1)^2 + 8m$ multiplications compared to the GMRES algorithm.

Because of the memory limitation, restarts might be required for the GMRES to solve the Newton equation. However, we should point out that the tensor model is not formed until the Newton equation is approximately solved by the GMRES algorithm or, in other words, until the Krylov space that contains the solution to the Newton equation is found. We form the tensor model only using the Krylov subspace generated in the last restarted GMRES algorithm. The tensor model has nothing to do with the intermediate Krylov spaces generated by the GMRES algorithm which resulted from restarts before the final restart. In addition, no extra preconditioning calculation is needed for obtaining the tensor step. Therefore, compared to the total cost of the Newton-GMRES with restarts, the extra cost of formation and solution of the tensor model is likely to be minimal for a large portion of nonlinear problems, particularly hard problems that need many restarts of the GMRES algorithm.

5. Implementation and testing. In the previous section, we presented the main new features of our tensor-GMRES method for nonlinear equations, namely, how to form the quadratic model of the nonlinear function and how to solve this model efficiently. In this section the complete algorithm is implemented to test these ideas. Various aspects of this algorithm are discussed; then, test results on several problems are presented.

Although the tensor model is derived from an ideal situation, i.e., assuming that the Newton equation is solved exactly by the GMRES method, test results indicate that the tensor method based on this model works fairly well in inexact situations.

This section is organized as follows. Section 5.1 gives a complete tensor-GMRES algorithm for systems of nonlinear equations and discusses the implementation of each step in detail. In sections 5.2–5.4 we will show comparative test results for the tensor-GMRES algorithm given in section 5.1 (Algorithm TG) versus the Newton-GMRES algorithm given in section 2 (Algorithm NG) with the same implementation. Three distinct test problems, i.e., the Broyden tridiagonal problem, a Bratu problem, and the one-dimensional Euler equations problem, and several of their variants were used in the testing. The tests on the Bratu problem, the Broyden tridiagonal problem, and their variants were performed on a Sun Super Workstation II+/50 using MATLAB. The test on the one-dimensional Euler equations problem was performed on a Cray Y-MP using FORTRAN 90.

5.1. A complete algorithm. The full algorithm of the tensor-GMRES method is defined in the following algorithm.

ALGORITHM TG. AN ITERATION OF THE TENSOR-GMRES METHOD.
Given x_k, $x_{k-1} \in \Re^N$, $J_k \in \Re^{N \times N}$, $F_k \in \Re^N$ and $F_{k-1} \in \Re^N$.

(TG-1) Decide whether to stop. If not:

(TG-2) Set $s = x_{k-1} - x_k$, $a = 2(F_{k-1} - F_k - J_k s)/(s^T s)$ and $h = s/\|s\|$. Choose a tolerance $\epsilon_k \in [0, 1)$.

(TG-3) Do GMRES (restart if necessary) to find $d^n = d_0 + V_m y^n$ such that

$$F_k + J_k d^n = r_k \quad \text{with} \quad \|r_k\|/\|F_k\| < \epsilon_k,$$

where d_0 is the starting point of the last restarted GMRES procedure, and the columns of V_m form an orthonormal basis for the Krylov space generated by the corresponding Arnoldi process. In addition, let $\bar{H}_m$ be the Hessenberg matrix generated from the Arnoldi process, and $\bar{H}_m = \bar{Q}_m \bar{R}_m$ be its QR-factorization. Let $\bar{R}_m^1$ be the first m rows of $\bar{R}_m$.

(TG-4) If $d_0 = 0$ or $d_0 \in \{V_m\}$ then

Solve

$$(5.1)\quad \min_{y \in \Re^m} \|F_k + J_k d_0 + J_k V_m y + \tfrac{1}{2}\bar{a}\{h^T(d_0 + V_m y)\}^2\|,$$

where $\bar{a} = (J_k V_m)\{(J_k V_m)^T (J_k V_m)\}^{-1}(J_k V_m)^T a$, by first solving

$$\min_{\beta \in \Re} \|q_1(\beta) \equiv \hat{h}_1^T (\bar{R}_m^1)^{-1} w + h^T d_0 - \beta - \tfrac{1}{2}\hat{h}_1^T (\bar{R}_m^1)^{-1} b \beta^2\|,$$

to obtain a solution β_*, where w is the first m components of $Q_m^T \|F_k + J_k d_0\| e_1$, b is the first m components of $Q_m^T V_{m+1}^T a$, and $\hat{h}_1 = V_m^T h$. Then the solution to (5.1) is given by

$$y_*^t = [(\bar{R}_m^1)^T (\bar{R}_m^1)]^{-1} \hat{h}_1 q_1(\beta_*)/\omega + (\bar{R}_m^1)^{-1} w - \tfrac{1}{2} (\bar{R}_m^1)^{-1} b \beta_*^2,$$

where $\omega = \hat{h}_1^T [(\bar{R}_m^1)^T (\bar{R}_m^1)]^{-1} \hat{h}_1$.

Form the tensor step $d^t = d_0 + V_m y_*^t$.

Otherwise ($d_0 \neq 0$ and $d_0 \notin \{V_m\}$),

Solve

$$\min_{\hat{y} \in \Re^{m+1}} \|F_c + \hat{J}\hat{y} + \tfrac{1}{2} \hat{a}_c (\hat{h}_2^T \hat{y})^2\|, \tag{5.2}$$

where $\hat{J} = [V_{m+1} \bar{H}_m, -F_k - r_0]$, $\hat{a}_c = \hat{J}(\hat{J}^T \hat{J})^{-1} \hat{J}^T a$ and $\hat{h}_2 = [V_m, d_0]^T h$. This is done by first solving

$$\min_{\beta \in \Re} \|q_2(\beta) \equiv \hat{h}_2^T (\hat{J}^T \hat{J})^{-1} \hat{J}^T F_c + \beta + \tfrac{1}{2} \hat{h}_2^T (\hat{J}^T \hat{J})^{-1} \hat{J}^T \hat{a}_c \beta^2\|$$

with solution β_*. Then the solution to (5.2) is given by

$$\hat{y}_* = (\hat{J}^T \hat{J})^{-1} \hat{h}_2 q_2(\beta_*)/\omega - (\hat{J}^T \hat{J})^{-1} \hat{J}^T (F_c + \tfrac{1}{2} \hat{a}_c \beta_*^2),$$

where $\omega = \hat{h}_2 (\hat{J}^T \hat{J})^{-1} \hat{h}_2$.

Form the tensor step $d^t = [V_m, d_0] \hat{y}_*$.

(TG-5) Choose a new step d between d^n and d^t.

(TG-6) Find $\lambda > 0$ using a backtracking line search global strategy and form the next iterate $x_{k+1} = x_k + \lambda d$.

Several tests are performed to determine whether to stop the algorithm in step (TG-1). These stopping criteria are described by Dennis and Schnabel in Chapter 7 of [14]. For the sake of simplicity, we only use simplified versions of their criteria. The first test determines whether x_k solves (1.1). This is accomplished by using $\|F(x_k)\| \leq$ FTOL, where a typical value of FTOL is around 10^{-5}. A much more stringent test used here is to set FTOL to 10^{-12}. The second test determines whether the algorithm has converged or stalled at x_k. It is done by measuring the relative change in the iterates from one step to the next. We use $\|x_k - x_{k-1}\|/\|x_{k-1}\| \leq$ STPTOL, where a typical STPTOL is around 10^{-8} in our implementation. Finally, we test if a maximum number of iterations is exceeded. Currently this value is 150.

In step (TG-2), we need to choose the tolerance ϵ_k, which is passed to the GMRES algorithm when it is called to solve the Newton equation at the kth iteration. In [6], Brown and Saad suggested a sequence $\epsilon_k = (\frac{1}{2})^k$ for $k = 1, 2, \ldots$. Since a good sequence is normally problem related, again for the sake of simplicity, we use a fixed ϵ_k at every iteration for our test problems.

Step (TG-3) calls the GMRES. The number of Arnoldi iterations allowed between restarts is usually set to 20, and the number of maximum restarts of GMRES allowed is usually set to 150. However, these two values can be provided by the users based on their experience. When the required tolerance is not reached after a maximum number of restarts, we simply go ahead and use the last computed data. When it returns, the GMRES algorithm readily provides $\bar{R}_m^1$ and $\bar{Q}_m$, which is a product of the m Givens rotations. One byproduct of step (TG-3) is the Newton-GMRES step.

Step (TG-4) calculates the tensor-GMRES step. It is basically a concise reiterate of the solution of the tensor model described in the previous section. The minimization of a quadratic function in one variable is done using standard root formula. When a quadratic function has two distinct roots, the root that is smaller in absolute value is chosen.

Step (TG-5) usually consists of choosing the tensor step direction d^t obtained in step (TG-4). However, the Newton step direction is chosen instead when the tensor step direction is not a descent direction for $\frac{1}{2}\|F(x)\|^2$, which rarely occurs in practice but is not precluded in theory. Since the gradient of $\frac{1}{2}\|F(x)\|^2$ is $J(x)^T F(x)$, d^t is a descent direction if $(d^t)^T J(x)^T F(x) < 0$. We discuss how to compute this expression efficiently. At current iterate x_c, on the one hand, when $d_0 \in \{V_m\}$, using $d^t = V_m y^t$, $J_c V_m = V_{m+1}\bar{H}_m$ and $F_c = -\|F_c\| v_1$ yields

$$\begin{aligned}(d^t)^T J_c^T F_c &= (V_m y^t)^T J_c^T F_c = (y^t)^T (J_c V_m)^T F_c \\ &= (y^t)^T (V_{m+1}\bar{H}_m)^T F_c = (\bar{H}_m y^t)^T (V_{m+1}^T F_c) = -(\bar{H}_m y^t)^T \|F_c\| e_1. \end{aligned} \tag{5.3}$$

The cost of calculating (5.3) is minimal. On the other hand, when $d_0 \notin \{V_m\}$, using $d^t = [V_m, d_0]\hat{y}^t$, $J_c V_m = V_{m+1}\bar{H}_m$ and $r_0 = -F_c - J_c d_0$ yields

$$\begin{aligned}(d^t)^T J_c^T F_c &= ([V_m, d_0]\hat{y}^t)^T J_c^T F_c = (\hat{y}^t)^T [J_c V_m, J_c d_0]^T F_c \\ &= (\hat{y}^t)^T [V_{m+1}\bar{H}_m, -r_0 - F_c]^T F_c = (\hat{y}^t)^T \begin{pmatrix} \bar{H}_m^T V_{m+1}^T F_c \\ -r_0^T F_c - \|F_c\|^2 \end{pmatrix}. \end{aligned} \tag{5.4}$$

The major work in (5.4) is the calculation of $V_{m+1}^T F_c$. However, since this calculation is already done in the solution of the tensor step, no extra cost is necessary.

Finally, in step (TG-6), we use a standard quadratic backtracking line search algorithm (see [14]). The merit function $\frac{1}{2}\|F(x)\|^2$ is used for measuring the progress towards the solution.

When d^n is chosen in step (TG-5), the directional derivative is given by $(d^n)^T J_c^T F_c$, which can be calculated in a fashion similar to when d^t is chosen. When $d_0 \in \{V_m\}$, we can simply replace y^t in (5.3) by y^n and calculate $-(\bar{H}_m y^n)\|F_c\| e_1$. When $d_0 \notin \{V_m\}$,

$$\begin{aligned}(d^n)^T J_c^T F_c &= (d_0 + V_m y^n)^T J_c^T F_c = (J_c d_0 + J_c V_m y^n)^T F_c \\ &= (-F_c - r_0 + V_{m+1}\bar{H}_m y^n)^T F_c = -\|F_c\|^2 - r_0^T F_c + (y^n)^T \bar{H}_m^T (V_{m+1}^T F_c), \end{aligned}$$

which is easy to calculate given $V_{m+1}^T F_c$.

5.2. Test results for the Broyden tridiagonal problem and its variants.

As a first test, the Broyden tridiagonal problem is chosen from a standard test set of Moré, Garbow, and Hillstrom [24]. The function is defined as

$$f_i(x) = (3 - 2x_i)x_i - x_{i-1} - 2x_{i+1} + 1 \quad \text{for} \quad i = 1, \dots, n, \tag{5.5}$$

where $x_0 = x_{n+1} = 0$ and n can be any positive integer. A root of $f = 0$ is sought. For our test, we set $n = 1000$, which results in a system of 1000 nonlinear equations in 1000 unknowns. The Jacobian matrix has full rank at the solution. The standard starting point is $x_0 = [-1, -1, \dots, -1]$.

Since starting from the standard starting point x_0 is too easy for both algorithms, we tried to start farther away from x_0, i.e., from $100{*}x_0$. Figure 1 shows the test results of using $\epsilon_k = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$. At the nonlinear level, both methods

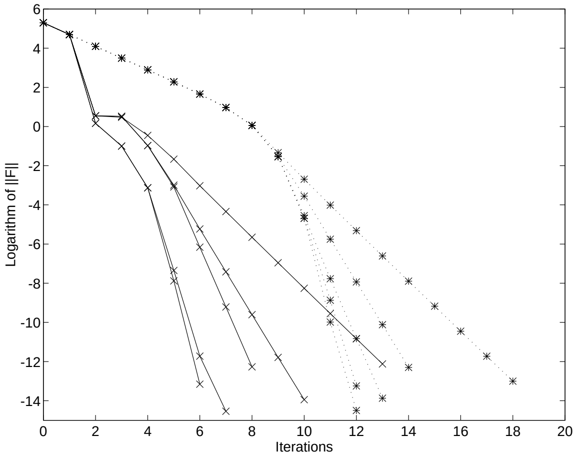


FIG. 1. *Results for the Broyden tridiagonal problem.* $x_0 = 100 * [-1, -1, \ldots, -1]^T$. *Diagonal preconditioning: solid line, tensor-GMRES; dotted line, Newton-GMRES.* $\epsilon_k = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$.

performed better as the tolerance goes down. In all five cases, the tensor-GMRES method required fewer nonlinear iterations. The margin of improvement ranges from 28% to 50%.

Next we give the test results for a rank one deficient modification of the Broyden tridiagonal problem. The problem was constructed by squaring the last function defined by (5.5). This construction does not alter the solutions to the original system and results in a system whose Jacobian matrix has rank deficiency one at the solution. Test results are given in Figure 2.

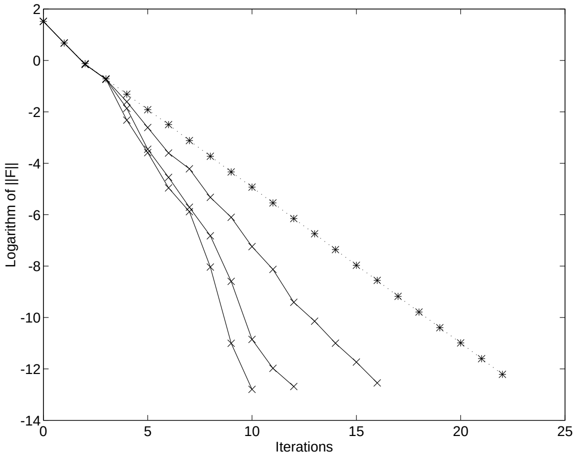


FIG. 2. *Results for a rank one deficient modification of the Broyden tridiagonal problem.* $x_0 = [-1, -1, \ldots, -1]^T$. *Diagonal preconditioning: solid line, tensor-GMRES; dotted line, Newton-GMRES.* $\epsilon = 10^{-1}, 10^{-2}, 10^{-9}$.

The tolerance ϵ_k was taken as $10^{-1}, 10^{-2}, 10^{-9}$, respectively. For all the three cases, the Newton-GMRES method performed about the same (not distinguishable in the figure). It took 22 iterations, while the tensor-GMRES method required fewer

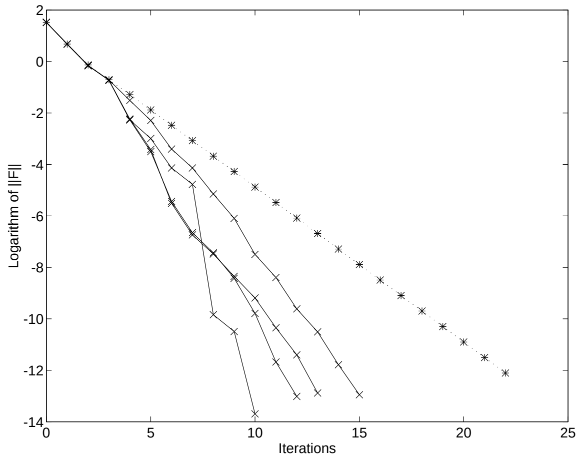


FIG. 3. *Results for a rank two deficient modification of the Broyden tridiagonal problem.* $x_0 = [-1, -1, \dots, -1]^T$. *Diagonal preconditioning: solid line, tensor-GMRES; dotted line, Newton-GMRES.* $\epsilon = 10^{-1}, 10^{-2}, 10^{-4}, 10^{-8}$.

iterations as the tolerance went down. The margin of improvement ranges from 27% to 55%. The tensor-GMRES method also shows superlinear-like convergence in two situations.

Finally, we give the test results for a rank two deficient modification of the Broyden tridiagonal problem. The problem was constructed by squaring the last two functions defined by (5.5). This construction does not alter the solutions to the original system and results in a system whose Jacobian matrix has rank deficiency two at the solution. The test results are given by Figure 3.

The tolerance ϵ_k was taken as $10^{-1}, 10^{-2}, 10^{-4}, 10^{-8}$, respectively. For all four cases, the Newton-GMRES method again performed about the same (taking 22 iterations), while the tensor-GMRES method required fewer iterations as the tolerance went down. The margin of improvement ranged from 32% to 55%.

5.3. Test results for a Bratu problem. As a second test, we choose to solve the nonlinear partial differential equation

$$-\Delta u = \lambda e^u \quad \text{in} \quad \Omega, \qquad u = 0 \quad \text{on} \quad \Gamma, \tag{5.6}$$

where $\Delta = \nabla^2 = \sum_{i=1}^{2} \partial^2/\partial x_i^2$ is the Laplace operator, $\Omega = (0,1) \times (0,1)$, and Γ is the boundary of Ω. This version of the Bratu problem is chosen from a set of nonlinear model problems collected by Moré [23].

We define $h = 1/(n+1)$, where n is a positive integer, and then a mesh is given by

$$M_{ij} = \{ih, jh\}, \qquad 0 \le i, j \le n.$$

To approximate problem (5.6) we use the following finite-difference scheme:

$$\begin{aligned} -\frac{u_{i+1j} + u_{i-1j} + u_{ij+1} + u_{ij-1} - 4u_{ij}}{h^2} &= \lambda e^{u_{ij}}, \qquad 1 \le i, j \le n \\ u_{kl} &= 0, \qquad\quad \text{if} \quad M_{kl} \in \Gamma. \end{aligned} \tag{5.7}$$

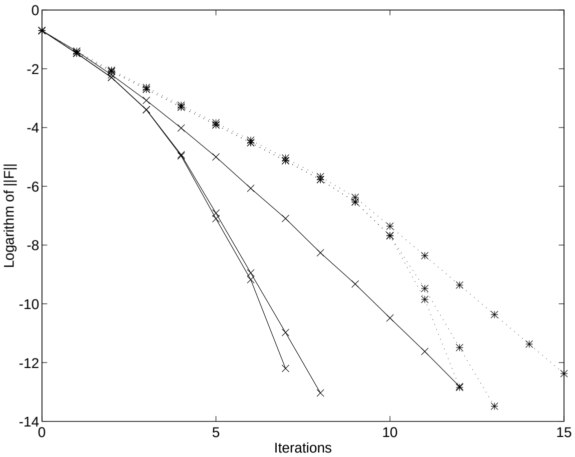


FIG. 4. *Results for the Bratu problem,* $\lambda = 6.80673$. *Diagonal preconditioning: solid line, tensor-GMRES*; *dotted line, Newton-GMRES.* $\epsilon_k = 10^{-1}, 10^{-2}, 10^{-3}$.

In (5.7), u_{ij} is an approximation to $u(M_{ij})$. For $\lambda \leq 0$, (5.6) has a unique solution. For $\lambda > 0$, (5.6) may have one, several, or no solutions. In this test, we took $n = 32$ and $\lambda = 6.80673$, which yields a system of $N = 1024$ equations in N unknowns. Since the branch of solutions has a limit point at $\lambda = 6.80812$, this gives a good test problem that has a very ill-conditioned Jacobian matrix at the solution. We did not use the exact limit point since our preconditioner is not good enough to converge the GMRES algorithm in this situation. The size of the Krylov subspace is set to 50. The initial guess was chosen as $u_0 = 0$. Test results are shown in Figure 4 for $\epsilon_k = 10^{-1}, 10^{-2}, 10^{-3}$. The tensor-GMRES method outperformed the Newton-GMRES method in all three cases with the margin of improvement ranging from 20% to 42%.

Before going to the next test problem, we would like to make a few more comments regarding Figures 1–4. For the Broyden tridiagonal problem with a nonsingular Jacobian, Figure 1 clearly shows linear convergence of both the tensor-GMRES and Newton-GMRES methods for each choice of ϵ, with smaller ϵ resulting in faster convergence and with the two methods exhibiting about the same ultimate speed of linear convergence for each choice of ϵ. However, it is very interesting that the tensor-GMRES method breaks into the regime in which the choice of ϵ controls the speed of convergence significantly sooner than Newton-GMRES. Figures 2 and 3 show advantages of the tensor-GMRES method when the Jacobian is singular. In these figures, the Newton-GMRES iterates show rather slow linear convergence, independent of ϵ, until termination, while the tensor-GMRES iterates show increasingly fast, roughly linear convergence as ϵ becomes smaller; thus, the convergence of Newton-GMRES is sharply limited by the singularity of the problem, while the convergence of tensor-GMRES is not. Figure 4 for the Bratu problem, in which the Jacobian is nearly singular, clearly shows the superiority of tensor-GMRES in overcoming the near singularity of the Jacobian at an early stage. Specifically, the Newton-GMRES method is "fooled" by the near singularity into exhibiting linear convergence until about the ninth iteration, after which the convergence accelerates, while the tensor-GMRES method exhibits accelerated convergence after the second iteration.

5.4. Test results for one-dimensional Euler equations. One of the target applications for the tensor-Krylov methods is the nonlinear differential systems arising in physical problems, e.g., aerodynamics. One good model problem is the quasi-one-dimensional Euler equations for flow through a nozzle with a given area ratio. In particular, transonic conditions which generate a shock within the nozzle present a difficult test case, where methods typical of practical aerodynamic applications are required. Such methods include finite difference, finite element, and unstructured grid finite volume techniques employing various forms of highly nonlinear algorithm constructions. For our purposes here, we have chosen one popular form of central finite differences with nonlinear artificial dissipation; see [26] for general details.

The steady quasi-one-dimensional Euler equations are

$$\mathbf{F}(\mathbf{Q}) = \partial_x \mathbf{E}(\mathbf{Q}) - \mathbf{H}(\mathbf{Q}) = 0, \quad 0.0 \le x \le 1.0, \tag{5.8}$$

where

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ e \end{bmatrix}, \quad \mathbf{E} = a(x) \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(e+p) \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} 0 \\ -p\partial_x a(x) \\ 0 \end{bmatrix}, \tag{5.9}$$

with ρ (density), u (velocity), e (energy), $p = (\gamma - 1)(e - 0.5\rho u^2)$ (pressure), $\gamma = 1.4$ (ratio of specific heats), and $a(x) = (1. - 4.(1 - a_t)x(1 - x))$ (the nozzle area ratio), with $a_t = 0.8$. For a given area ratio and shock location (here $x = 0.7$), an exact solution can be obtained from the method of characteristics.

We elect to use second order central differences

$$\partial_x u \approx \delta_x u_j = \frac{u_{j+1} - u_{j-1}}{2\Delta x}, \quad j = 0, \ldots, J_N, \quad \Delta x = 1.0/J_N, \quad u_j = u(j\Delta x). \tag{5.10}$$

It is common practice and well known that artificial dissipation must be added to the discrete central difference approximations in the absence of any other dissipative mechanism, especially for transonic flows. Nonlinear dissipation as defined in [27] is used where 2nd order ($D^2(\mathbf{Q})$) and 4th order ($D^4(\mathbf{Q})$) difference formulas are employed.

$$D_j^2(\mathbf{Q}) = -\nabla_x \left(\sigma_{j+1} + \sigma_j\right) \left(\epsilon_j^{(2)} \Delta_x \mathbf{Q}_j\right), \tag{5.11a}$$

$$D_j^4(\mathbf{Q}) = \nabla_x \left(\sigma_{j+1} + \sigma_j\right) \left(\epsilon_j^{(4)} \Delta_x \nabla_x \Delta_x \mathbf{Q}_j\right), \tag{5.11b}$$

with

$$\begin{aligned} \nabla_x q_j &= q_j - q_{j-1}, \quad \Delta_x q_j = q_{j+1} - q_j, \\ \epsilon_j^{(2)} &= \kappa_2 \max(\Upsilon_{j+1}, \Upsilon_j, \Upsilon_{j-1}), \\ \Upsilon_j &= \frac{|p_{j+1} - 2p_j + p_{j-1}|}{|p_{j+1} + 2p_j + p_{j-1}|}, \end{aligned} \tag{5.11c}$$

$$\epsilon_j^{(4)} = \max\left(0, \kappa_4 - \epsilon_j^{(2)}\right), \tag{5.11d}$$

where typical values of the constants are $\kappa_2 = 1/4$ and $\kappa_4 = 1/100$. The term $\sigma_j = |u| + c$ (where $c = \sqrt{\gamma p/\rho}$ is the speed of sound) is a spectral radius scaling.

Boundary operators at $j = 0$ and $j = J_N$ are defined in terms of physical conditions (taken from exact solution values) and the use of Riemann invariants. For this problem, both inflow and outflow boundaries are subsonic and locally one-dimensional Riemann invariants are used. The locally one-dimensional Riemann invariants are given in terms of the velocity component as

$$R_1 = u - 2c/(\gamma - 1) \quad \text{and} \quad R_2 = u + 2c/(\gamma - 1). \tag{5.12}$$

The Riemann invariants R_1, R_2 are associated with the two characteristic velocities $\lambda_1 = u - c$ and $\lambda_2 = u + c$, respectively. One other equation is needed so that the three flow variables can be calculated. We choose $S = \ln(p/\rho^\gamma)$, where S is entropy. For subsonic inflow ($u < c$), characteristic velocity $\lambda_2 > 0$ carries information into the domain, and therefore the characteristic variable R_2 can be specified along with one other condition. The Riemann invariant R_2 and S are set to exact values. The other characteristic velocity $\lambda_1 < 0$ carries information outside the domain, and therefore R_1 is extrapolated from the interior flow variables. On subsonic outflow $u < c$ and $\lambda_2 > 0$ carry information outside the domain, while $\lambda_1 < 0$ propagates into the domain, so only R_1 is fixed to exact values, and R_2 and S are extrapolated. Once these three variables are available at the boundary the three flow variables $\mathbf{Q}$ can be obtained. If we consider the boundary procedure as an operator on the interior data, we can cast the boundary scheme as

$$B(\mathbf{Q})_i = \mathbf{Q}_i - \mathbf{B}(\mathbf{Q}_{i+1}) = 0, \quad i = 0$$

and

$$B(\mathbf{Q})_i = \mathbf{Q}_i - \mathbf{B}(\mathbf{Q}_{i-1}) = 0, \quad i = J_N,$$

which are nonlinear equations at the boundaries, where $\mathbf{B}(\mathbf{Q})$ represents the action of the boundary condition operator (subroutine) on interior data.

The total system we shall solve is

$$\mathcal{F}(\mathbf{Q}) = \begin{cases} \delta_x \mathbf{E}(\mathbf{Q})_j - \mathbf{H}(\mathbf{Q})_j + D_j^2(\mathbf{Q}) + D_j^4(\mathbf{Q}), & j = 1, \ldots, J_N - 1, \\ B(\mathbf{Q})_i = 0, & i = 0, J_N. \end{cases} \tag{5.13}$$

An analytic Jacobian can be formed directly from (5.13) by differentiating each term with respect to $\mathbf{Q}_j$, thereby producing a block banded matrix. The order of the resulting system is $N = (J_N + 1) \times 3$.

Forming the Jacobian from derivatives using (5.13) is only difficult when dealing with the highly nonlinear, nonanalytic coefficients of the artificial dissipation (5.11d). In the case of the artificial dissipation Jacobians (e.g., $\partial \mathbf{Q}_j(D_j^4)$), two approximations are made. In the first part of the approximation, the spectral radius is not linearized since it contains an absolute value function, a nondifferentiable form. In the second part of the approximation, the nonlinear switch functions (5.11d), containing absolute values and max operators, are also not linearized. In both cases, frozen values at the local states are used to evaluate these terms as variable coefficients. It is common practice in fluid dynamic algorithms to avoid linearization of nonlinear switches and higher order terms by either restricting bandwidth or avoiding dealing with complicated functional forms, e.g., [1, 29].

The calculation of these nondifferentiable terms could be avoided by approximating Jacobian-vector products with finite differences (see (2.4)). If an exact linearization is used (and since the system is nonsingular), it is then expected (and confirmed numerically) that q-quadratic-like convergence would be achieved by both Newton-GMRES and tensor-GMRES when the tight tolerance level $\epsilon_k = 10^{-8}$ is applied. In

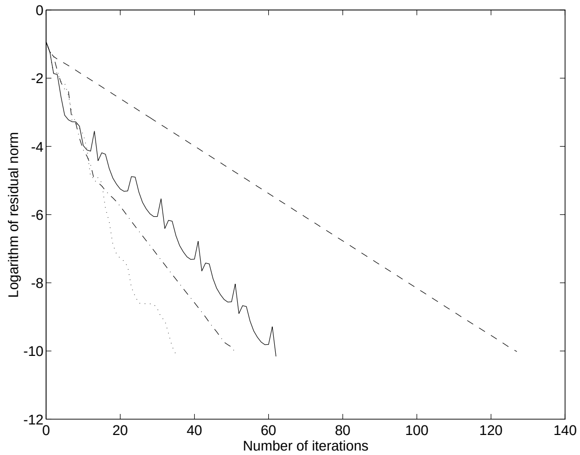


FIG. 5. *Results for one-dimensional Euler using full nonlinear dissipation. Solid line, dashed line, Newton-GMRES with* $\epsilon_k = 10^{-3}, 10^{-5}$; *dotted line, dash-dotted line, tensor-GMRES with* $\epsilon_k = 10^{-3}, 10^{-5}$.

order to provide a more realistic setting (as noted above it is commonplace to employ approximate Jacobians in practical applications), most of the results presented employ the approximate Jacobian.

A key element to the success of the solution using the Krylov subspace methods is the choice of preconditioning. This issue for systems which are not diagonally dominant, such as (5.13), is not straightforward and is still the subject of active research. One choice of a preconditioner is the inverse of the Jacobian matrix (which is not too difficult to construct for this one-dimensional problem but would be very difficult in multidimensions). Inexact preconditioners are employed in the test, which allows us to study the effect of different tolerance levels for the linear steps. In a one-dimensional setting, a large number of effective approximate preconditioners are possible. For our study here, we only require one which gives a finite number of required subspace iterations (the linear GMRES phase) meeting some preset tolerance level ϵ_k. We shall not go into the details of the preconditioner here and only state that the same preconditioner is used for both Algorithms NG and TG so that consistent comparisons can be made.

Figure 5 shows Algorithms NG and TG applied to (5.13) for $J_N = 200$; $N = 603$. Two sets of results are shown for two values of the tolerance parameter $\epsilon_k = 10^{-3}$ and $\epsilon_k = 10^{-5}$. In the case of Algorithm NG, the convergence appears linear (it takes 127 steps to converge for a tight tolerance $\epsilon_k = 10^{-5}$), while Algorithm TG for $\epsilon_k = 10^{-5}$ shows about a 60% (76 steps) decrease in the number of nonlinear iterations. When the tolerance is increased, i.e., $\epsilon_k = 10^{-3}$, Algorithm NG actually achieves improved performance (it takes 62 steps to converge), while Algorithm TG for $\epsilon_k = 10^{-3}$ shows about a 42% (26 steps) decrease in the number of nonlinear iterations. In addition, the convergence for the tensor algorithm is much smoother. The "scalloping" behavior of Algorithm NG is due to the use of the minimal step length triggered by line search failures.

To date, our analysis indicates that the system derived from (5.13) is nonsingular, so we do not consider this an example similar to the singular ones presented above. The inaccurateness of the Jacobian approximation is the source of the linear behavior

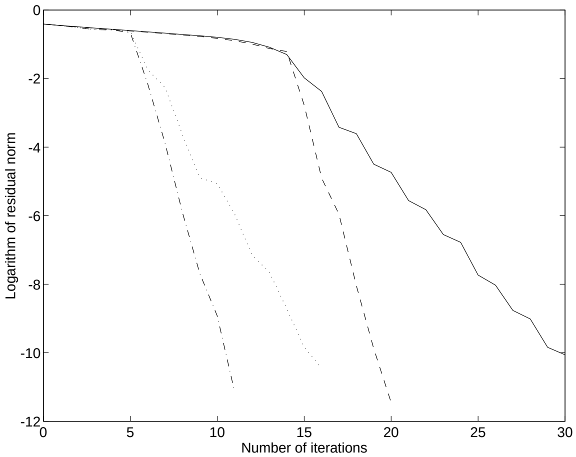


FIG. 6. *Results for one-dimensional Euler using nonlimited dissipation. Solid line, dotted line, Newton-GMRES with* $\epsilon_k = 10^{-3}, 10^{-5}$; *dashed line, dash-dotted line, tensor-GMRES with* $\epsilon_k = 10^{-3}, 10^{-5}$.

observed. In particular, the inaccurate linearization of the second order dissipation term, $D^2(\mathbf{Q})$, produces the largest error. Figure 6 shows the convergence results with $\kappa_2 = 0$, resulting in a very fast linear convergence ($\epsilon_k = 10^{-5}$) from both Algorithms NG and TG. For both cases of $\epsilon_k = 10^{-3}$ and $\epsilon_k = 10^{-5}$, the tensor algorithm shows about a 50% improvement in the number of nonlinear iterations over the Newton counterpart. Nonlinear switching, such as is defined in (5.11a–5.11d), is typical of current numerical algorithms for the Euler and Navier–Stokes equations. They may take a similar form to (5.11a–5.11d), see [27], or be in the form of limiters for upwind techniques, e.g., [33], [20]. The nonlinear switching (limiting) is necessary to eliminate overshoots at shocks, where higher order schemes are limited to lower order, which more correctly differences the equations across discontinuities.

6. Summary and topics for future research. This paper has introduced the tensor-GMRES method for systems of nonlinear equations. This method has requirements similar to the Newton-GMRES method in terms of storage and arithmetic per iteration. The method is also consistent with preconditioning and matrix-free implementation. An implementation of the full nonlinear algorithm using the tensor-GMRES method has shown to be more efficient on both nonsingular and singular problems than analogous implementation of the Newton-GMRES method. The efficiency advantage of the tensor-GMRES method is significantly larger on problems where the Newton-GMRES method exhibits linear convergence (due to lack of sufficient first order information).

One interesting question is whether the tensor-GMRES method would be more robust than the Newton-GMRES method since it depends on the solution of the Newton equation by the GMRES algorithm. The answer is twofold. At the linear level, the answer is no, since the tensor-GMRES fails when the GMRES fails. However, at the nonlinear level, from our experience with traditional tensor methods where the Newton equation is solved exactly, tensor model-based methods solve more problems than linear-model–based methods. Hence, the tensor-GMRES is expected to be more robust than the Newton-GMRES method at the nonlinear level.

Based on these results, it would appear worthwhile to continue research on tensor-

Krylov methods for nonlinear equations. The two main topics for future research would appear to be practical implementation and further testing of the tensor-GMRES methods for nonlinear equations and new tensor-Krylov methods for nonlinear equations. We discuss each of these briefly.

As seen in section 5, our implementation is still in an early stage. Several directions can be pursued immediately to improve the current implementation: (1) scaling in both the variable space and the function space; (2) matrix-free implementation of the tensor-GMRES method, which can be achieved in a fashion similar to analogous implementation of the Newton-GMRES method; (3) more sophisticated stopping criteria in the nonlinear algorithm; (4) more global convergence strategies such as model trust region techniques. We would like to continue our testing of the tensor-GMRES method on more practical problems. One interesting task is to test the tensor-GMRES method on the ARC2D code [26] which is the two dimensional version of the ARC1D code that we tested in section 5.

Secondly, new tensor-Krylov methods can be developed. A nonstraightforward direction that can be pursued in the future is to combine tensor methods with Krylov methods that use two mutually orthogonal sequences such as BiCG and QMR. We are currently investigating this possibility.

Acknowledgments. We thank Prof. Robert Schnabel for many helpful discussions and suggestions related to this paper. We also thank the anonymous referees for their careful reading of the manuscript and for their constructive criticism.

REFERENCES

[1] T. J. Barth, *Analysis of Implicit Local Linearization Techniques for Upwind and TVD Algorithms*, Technical report AIAA-87-0595, AIAA 25th Aerospace Sciences Meeting, Reno, Neveda, 1987.

[2] A. Bouaricha, *Solving Large Sparse Systems of Nonlinear Equations and Nonlinear Least Squares Problems using Tensor Methods on Sequential and Parallel Computers*, Ph.D. thesis, Department of Computer Science, University of Colorado, Boulder, CO, 1992.

[3] A. Bouaricha, *Tensor-Krylov Methods for Large Nonlinear Equations*, Technical report MCS-P482-1194, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1994.

[4] A. Bouaricha and R. B. Schnabel, *Tensor Methods for Large, Sparse Nonlinear Least Squares Problems*, Preprint MCS-P552-1295, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1994.

[5] P. N. Brown and A. C. Hindmarsh, *Reduced storage methods in stiff ODE systems*, J. Appl. Math. Comput., 31 (1989), pp. 40–91.

[6] P. N. Brown and Y. Saad, *Hybrid Krylov methods for nonlinear systems of equations*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 450–481.

[7] P. N. Brown and Y. Saad, *Convergent theory of nonlinear Newton–Krylov algorithms*, SIAM J. Optim., 4 (1994), pp. 297–330.

[8] T. F. Chan and K. R. Jackson, *The use of iterative linear equation solvers in codes for large systems of stiff IVPs for ODEs*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 378–417.

[9] D. W. Decker, H. B. Keller, and C. T. Kelley, *Convergence rate for Newton's method at singular points*, SIAM J. Numer. Anal., 20 (1983), pp. 296–314.

[10] D. W. Decker and C. T. Kelley, *Newton's method at singular points* I, SIAM J. Numer. Anal., 17 (1980), pp. 66–70.

[11] D. W. Decker and C. T. Kelley, *Newton's method at singular points* II, SIAM J. Numer. Anal., 17 (1980), pp. 465–471.

[12] D. W. Decker and C. T. Kelley, *Convergence acceleration for Newton's method at singular points*, SIAM J. Numer. Anal., 19 (1981), pp. 219–229.

[13] R. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[14] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Nonlinear Equations and Unconstrained Optimization*, Prentice–Hall, Englewood Cliffs, NJ, 1983.

[15] S. C. Eisenstat and H. F. Walker, *Globally convergent inexact Newton methods*, SIAM J. Optim., 4 (1994), pp. 393–422.

[16] D. Feng, P. D. Frank, and R. B. Schnabel, *Local convergence analysis of tensor methods for nonlinear equations*, Math. Programming, 62 (1993), pp. 427–459.

[17] D. Feng and R. B. Schnabel, *Tensor methods for equality constrained optimization*, SIAM J. Optim., 6 (1996), pp. 653–673.

[18] A. Griewank, *On solving nonlinear equations with simple singularities or nearly singular solutions*, SIAM Rev., 27 (1985), pp. 537–563.

[19] A. Griewank and M. R. Osborne, *Analysis of Newton's method at irregular singularities*, SIAM J. Numer. Anal., 20 (1983), pp. 747–773.

[20] A. Harten, *High resolution schemes for hyperbolic conservation laws*, J. Comput. Phys., 49 (1983), pp. 357–393.

[21] C. T. Kelley and R. Suresh, *A new acceleration method for Newton's method at singular points*, SIAM J. Numer. Anal., 20 (1983), pp. 1001–1009.

[22] C. T. Kelley and Z. Q. Xue, *Inexact Newton methods for singular problems*, Optim. Methods Software, 2 (1993), pp. 249–267.

[23] J. J. Moré, *A collection of nonlinear model problems*, Lectures Appl. Math., 26 (1990), pp. 723–762.

[24] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, *Testing unconstrained optimization software*, ACM Trans. Math. Software, 7 (1981), pp. 17–41.

[25] J. M. Ortega, *Numerical Analysis*, Academic Press, New York, 1972.

[26] T. H. Pulliam, *Efficient solution methods for the Navier–Stokes equations*, in Lecture Notes for the von Kármán Institute For Fluid Dynamics Lecture Series: Numerical Techniques for Viscous Flow Computation In Turbomachinery Bladings, von Kármán Institute, Rhode-St-Genese, Belgium, 1985.

[27] T. H. Pulliam, *Artificial dissipation models for the Euler equations*, AIAA J., 24 (1986), pp. 1931–1940.

[28] G. W. Reddien, *On Newton's method for singular problems*, SIAM J. Numer. Anal., 15 (1978), pp. 993–996.

[29] S. E. Rogers, *A Comparison of Implicit Schemes for the Incompressible Navier–Stokes Equations with Artificial Compressiblity*, Technical report AIAA-95-0567, AIAA 33rd Aerospace Sciences Meeting, Reno, NV, 1995.

[30] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[31] R. B. Schnabel and P. D. Frank, *Tensor methods for nonlinear equations*, SIAM J. Numer. Anal., 21 (1984), pp. 815–843.

[32] G. W. Stewart, *Error and perturbation bounds for subspaces associated with certain eigenvalue problems*, SIAM Rev., 15 (1973), pp. 727–764.

[33] P. K. Sweby, *High resolution schemes using flux limiters for hyperbolic conservation laws*, SIAM J. Numer. Anal., 21 (1984), pp. 995–1011.

SIAM J. Optim.
Vol. 7, No. 3, pp. 780–813, August 1997

011

ON THE REALIZATION OF THE WOLFE CONDITIONS IN REDUCED QUASI-NEWTON METHODS FOR EQUALITY CONSTRAINED OPTIMIZATION [*]

JEAN CHARLES GILBERT[†]

Abstract. This paper describes a reduced quasi-Newton method for solving equality constrained optimization problems. A major difficulty encountered by this type of algorithm is the design of a consistent technique for maintaining the positive definiteness of the matrices approximating the reduced Hessian of the Lagrangian. A new approach is proposed in this paper. The idea is to search for the next iterate along a piecewise linear path. The path is designed so that some generalized Wolfe conditions can be satisfied. These conditions allow the algorithm to sustain the positive definiteness of the matrices from iteration to iteration by a mechanism that has turned out to be efficient in unconstrained optimization.

Key words. constrained optimization, exact penalty function, global convergence, piecewise line-search, reduced quasi-Newton, successive quadratic programming, Wolfe's conditions

AMS subject classifications. Primary, 49M37; Secondary, 65K05, 90C30

PII. S1052623493259604

1. Introduction.

In unconstrained optimization, when a function $x \in \mathbb{R}^n \mapsto \xi(x) \in \mathbb{R}$ is minimized using descent direction methods, there is a nice combination of a line-search technique attributed to Wolfe [43, 44] and some quasi-Newton methods. On the one hand, if d_k is a descent direction of ξ at the current iterate x_k (i.e., $\nabla\xi(x_k)^\top d_k < 0$), the Wolfe line-search consists in determining a step-size $\alpha_k > 0$ along d_k such that the next iterate $x_{k+1} = x_k + \alpha_k d_k$ satisfies

$$\xi(x_{k+1}) \le \xi(x_k) + \omega_1\, \alpha_k\, \nabla\xi(x_k)^\top d_k, \tag{1.1}$$

$$\nabla\xi(x_{k+1})^\top d_k \ge \omega_2\, \nabla\xi(x_k)^\top d_k, \tag{1.2}$$

where $0 < \omega_1 < \omega_2 < 1$ are constants (independent of k). These conditions contribute to the convergence of descent direction methods. On the other hand, in quasi-Newton methods the descent direction has the form $d_k = -B_k^{-1}\nabla\xi(x_k)$, where B_k is an updated symmetric matrix approximating the Hessian of ξ. It is interesting to maintain this matrix positive definite, in particular because d_k is then a descent direction. With most update formulas, the new matrix B_{k+1} satisfies the so-called quasi-Newton equation

$$\gamma_k = B_{k+1}\delta_k,$$

where $\gamma_k = \nabla\xi(x_{k+1}) - \nabla\xi(x_k)$ is the change in the gradient of ξ and $\delta_k = \alpha_k d_k$ is the step. Of course, if B_{k+1} is positive definite, the quasi-Newton equation implies that

$$\gamma_k^\top \delta_k > 0. \tag{1.3}$$

[*] Received by the editors December 9, 1993; accepted for publication (in revised form) February 12, 1996.
http://www.siam.org/journals/siopt/7-3/25960.html

[†] INRIA, Rocquencourt, BP 105, 78153 Le Chesnay Cedex, France (Jean-Charles.Gilbert@inria.fr).

Therefore, this curvature condition (1.3) has to be satisfied if one expects B_{k+1} to be positive definite. For some quasi-Newton formulas (for instance the BFGS formula, see below), which update B_k using γ_k and δ_k, this inequality is also sufficient to have B_{k+1} positive definite (provided B_k is already positive definite). The remarkable fact is that the second Wolfe condition above guarantees this inequality. Hence, using the Wolfe line-search and the BFGS formula, e.g., ensures that all the search directions have the descent property.

For various reasons (see, for example, Powell [33]), it is not straightforward to extend the above scheme to a minimization problem with constraints on the variables. Such an extension is desirable, however, because numerical experience has shown that the approach is very successful in unconstrained minimization, even when the number of variables is large (see Liu and Nocedal [26] and Gilbert and Lemaréchal [20]).

In this paper, we study in more detail the matter for the equality constrained minimization problem

$$\begin{cases} \min f(x) \\ c(x) = 0, \quad x \in \Omega, \end{cases} \tag{1.4}$$

where $\Omega \subset \mathbb{R}^n$ is an *open* set and $f : \Omega \to \mathbb{R}$ and $c : \Omega \to \mathbb{R}^m$ $(m < n)$ are smooth functions.

Since the set Ω is supposed to be open, it cannot be used to define general constraints. It is the set where f and c have nice properties. For example, we always suppose that the $m \times n$ Jacobian matrix of the constraints

$$A(x) = \nabla c(x)^\top$$

is surjective (i.e., has full row rank) for any $x \in \Omega$. We also suppose that this matrix has a right inverse $A^-(x)$ depending smoothly on x:

$$A(x)A^-(x) = I \quad \forall x \in \Omega.$$

Besides, we assume that for all $x \in \Omega$, there is a basis $Z^-(x)$ of the null space $N(A(x))$ of $A(x)$, which means that $Z^-(x)$ is an injective (or full column rank) $n \times (n-m)$ matrix satisfying

$$A(x)Z^-(x) = 0 \quad \forall x \in \Omega.$$

We also suppose that the map $x \mapsto Z^-(x)$ is smooth. These assumptions on Z^- are not restrictive if Ω may differ from $\mathbb{R}^n$ but can rarely be satisfied when $\Omega = \mathbb{R}^n$ (for example, the assumptions on Z^- cannot be satisfied on even-dimensional spheres). Observe that for $A^-(x)$ and $Z^-(x)$ defined as above, there exists a unique $(n-m) \times n$ matrix $Z(x)$ such that

$$Z(x)Z^-(x) = I \quad \text{and} \quad Z(x)A^-(x) = 0 \tag{1.5}$$

in $\mathbb{R}^{(n-m)\times(n-m)}$ and $\mathbb{R}^{(n-m)\times m}$, respectively (see Gabay [14], for example).

The *Lagrangian function* of problem (1.4) is the function $\ell : (x, \lambda) \in \Omega \times \mathbb{R}^m \to \mathbb{R}$, defined by

$$\ell(x, \lambda) = f(x) + \lambda^\top c(x).$$

Its Hessian with respect to x is denoted by $L(x, \lambda) = \nabla^2_{xx}\ell(x, \lambda)$. The *reduced Hessian of the Lagrangian* is the order $n - m$ matrix $Z^-(x)^\top L(x, \lambda)Z^-(x)$. We denote by x_*

a solution of (1.4) and by λ_* its associated multiplier and denote $L_* = L(x_*, \lambda_*)$ and $B_* = Z^-(x_*)^\top L_* Z^-(x_*)$.

Our study is done in the framework of those reduced quasi-Newton methods that, near a solution x_*, generate the sequence of iterates $\{x_k\} \subset \Omega$ approximating x_* by

$$x_{k+1} = x_k + d_k, \tag{1.6}$$

where d_k is the solution of the quadratic program

$$\begin{cases} \min \nabla f(x_k)^\top d + \frac{1}{2} d^\top Z_k^\top B_k Z_k d \\ c_k + A_k d = 0. \end{cases} \tag{1.7}$$

In (1.7), $c_k = c(x_k)$, $A_k = A(x_k)$, $Z_k = Z(x_k)$, and the order $n-m$ matrix B_k is an approximation of the reduced Hessian of the Lagrangian (see Murray and Wright [30] and Gabay [15]). Since $Z_k^\top B_k Z_k$ approximates only a part of the Hessian of the Lagrangian, the method differs from the well-known sequential quadratic programming (SQP) algorithm (see Wilson [42], Han [22], and Pshenichnyi and Danilin [10]) in which an approximation of the full Hessian of the Lagrangian is updated. These reduced quasi-Newton algorithms have a lower speed of convergence than SQP methods, but they may be used for larger problems because they need to update smaller matrices.

Any direction d satisfying the linear constraints in (1.7) has the form $d = Z_k^- h - A_k^- c_k$, where $Z_k^- = Z^-(x_k)$, $A_k^- = A^-(x_k)$, and h is some vector in $\mathbb{R}^{n-m}$. Substituting this in the objective function of (1.7), assuming that B_k is positive definite, and minimizing in h, we obtain as a solution of (1.7)

$$d_k = t_k + r_k = -Z_k^- B_k^{-1} g_k - A_k^- c_k,$$

where $g_k = g(x_k) = Z_k^{-\top} \nabla f(x_k) \in \mathbb{R}^{n-m}$ is called the *reduced gradient* of f at x_k, $t_k = -Z_k^- B_k^{-1} g_k$ is called the *tangential* or *longitudinal* component of the step, and $r_k = -A_k^- c_k$ is called the *restoration* or *transversal* component of the step.

One of the main concerns of this paper is to develop a technique that maintains the positive definiteness of the matrices B_k. This property is interesting because it makes the direction t_k a descent direction of most merit functions used to globalize the local method (1.6)–(1.7). It is also natural since this matrix approximates the reduced Hessian of the Lagrangian, which is positive semidefinite at the solution. To obtain this property, our approach mimics what is done in unconstrained optimization, as was recalled in the beginning of this introduction. First, we use an update formula allowing the positive definiteness to be transmitted from one matrix to the next one. A typical example is the BFGS formula (see [13, 21, 11])

$$B_{k+1} = B_k - \frac{B_k \delta_k \delta_k^\top B_k}{\delta_k^\top B_k \delta_k} + \frac{\gamma_k \gamma_k^\top}{\gamma_k^\top \delta_k}. \tag{1.8}$$

This formula requires the use of two vectors γ_k and δ_k in $\mathbb{R}^{n-m}$, which will be specified in a moment. The important point is that the positive definiteness is sustained from B_k to B_{k+1} if the vectors γ_k and δ_k satisfy the following condition:

$$\gamma_k^\top \delta_k > 0. \tag{1.9}$$

Next, we propose a "piecewise line-search" (PLS) technique that finds a point satisfying *generalized* Wolfe conditions, which reduce to conditions (1.1)–(1.2) when there are no constraints. These conditions imply inequality (1.9) for appropriate vectors γ_k and δ_k and, therefore, also the positive definiteness of the matrices updated by using these vectors.

The local analysis of algorithm (1.6)–(1.7) shows that it is convenient to take for γ_k the change in the reduced gradient and for δ_k the reduced displacement

$$\gamma_k = g_{k+1} - g_k \quad \text{and} \quad \delta_k = \alpha_k Z_k t_k. \tag{1.10}$$

Other choices are sometimes proposed: see, for instance, Coleman and Conn [7] and Nocedal and Overton [31]. All of them are asymptotically equivalent to the above choice, which is preferred for its geometrical interpretation (see section 3) and its simplicity. In these formulas appears a step-size $\alpha_k > 0$ (see section 3) because the matrices B_k are also updated far from the solution where the algorithm differs from (1.6)–(1.7). Note, however, that x_{k+1} is obtained in a more sophisticated way than a simple move along the tangent direction t_k. This is necessary because such a move does not usually yield (1.9) (see [17]).

Condition (1.9) holds if the search algorithm determines x_{k+1} such that

$$g_{k+1}^\top Z_k t_k \geq \omega_2\, g_k^\top Z_k t_k, \tag{1.11}$$

where $0 < \omega_2 < 1$. This is actually what the search algorithm realizes. Now, this algorithm has another role to play, which is to contribute to the global convergence of the method. This is achieved by sufficiently decreasing some merit function, which we choose to be

$$\Theta_\sigma(x) = f(x) + \sigma \|c(x)\|, \tag{1.12}$$

where σ is positive number and $\|\cdot\|$ denotes a norm in $\mathbb{R}^m$. This penalty function is exact when σ is sufficiently large (see, for example, Han and Mangasarian [23]). The decrease in Θ_σ is typically forced by requiring that

$$\Theta_\sigma(x_{k+1}) \leq \Theta_\sigma(x_k) + \omega_1\, \nu_k(\alpha_k), \tag{1.13}$$

where $0 < \omega_1 < 1$ and $\nu_k(\alpha)$ is negative for positive α. Note that we do not need $\omega_1 < \omega_2$ in the PLS algorithm.

The difficulty of realizing both (1.11) and (1.13) simultaneously comes from the fact that, unlike what happens for unconstrained problems, the left-hand side in (1.11) is not the directional derivative of Θ_σ at x_{k+1} along commonly used search directions such as t_k or d_k. We shall see that it is the directional derivative $\Theta'_\sigma(x_{k+1}; Z^-_{k+1} Z_k t_k)$. This suggests making a reorientation of the search direction when (1.11) does not hold by using the new basis Z^-_{k+1}, while keeping the same reduced tangent direction $Z_k t_k$. This is the idea underlying the search algorithm proposed in [17], where the search path has only longitudinal components, i.e., components in the range space of the matrices $Z^-(x_k^i)$, where x_k^i $(i = 0, \ldots, i_k - 1)$ are intermediate points. Here we show how to implement this idea for paths also having transversal components, i.e., components in the range space of $A^-(x_k^i)$. This improves the algorithm, since asymptotically the constraints need no longer be linearized twice per iteration of the overall algorithm.

The analysis results in a quite simple search algorithm, which can be described as follows. At each inner iteration i of the PLS algorithm, condition (1.13) is first realized and, next, condition (1.11) is tested. If the latter holds, the PLS algorithm terminates with a suitable point. Otherwise, a new inner search direction is defined, using the *same* matrix B_k and the *same* reduced gradient g_k as for the previous inner direction. A new inner iteration is then started.

Other authors have proposed techniques for maintaining the positive definiteness of the generated matrices for constrained minimization problems, but none uses the search algorithm to achieve this goal. These papers also deal with the SQP method, in which approximations of the full Hessian of the Lagrangian are generated. In this case γ_k is usually γ_k^ℓ, the change in the gradient of the Lagrangian, and δ_k is the step. The first proposal, due to Powell [33], was to take for γ_k a convex combination of γ_k^ℓ and $B_k\delta_k$ such that (1.9) holds. According to Powell [35], the method may lead to ill conditioning when the problem is difficult to solve. We have also observed the failure of this technique on some academic problems (see Armand and Gilbert [1]). Due to its great simplicity, however, it is the most widely implemented technique. Another promising idea, proposed by Han [22] and Tapia [40] and subsequently explored by Tapia [41] and Byrd, Tapia, and Zhang [5], is to generate approximations of the Hessian of the augmented Lagrangian, which is positive definite at the solution when the penalty parameter is sufficiently large. The difficulty in choosing the penalty parameter has always been the stumbling block of this approach, and we believe that more research is needed to improve the method satisfactorily. Finally, Fenyes [12] and Coleman and Fenyes [8] separately update approximations of $Z^-(x_*)^\top L_* Z^-(x_*)$ and $A^-(x_*)^\top L_* Z^-(x_*)$, maintaining positive definite the approximations of the former matrix.

We conclude this introduction with a few remarks. First, our PLS algorithm also can be used for the reduced quasi-Newton method of Coleman and Conn [6] with minor modifications (see [18]), while its use for the SQP method has been investigated by Armand and Gilbert [1]. An important point to mention is that when the reduced Hessian of the Lagrangian is computed exactly and used in place of B_k in (1.7), there is no need to use the PLS algorithm. In this case, a simple Armijo [2] backtracking along d_k is preferable, since it is less expensive and easier to implement than the PLS algorithm.

The paper is organized as follows. In section 2, we make the hypotheses and notation more precise. In section 3, the search path is introduced and its meaning is discussed. Also, conditions for obtaining finite termination of the search algorithm are given. Section 4 contains a global convergence result and, finally, some numerical experiments are reported in section 5.

2. Hypotheses and notation. We suppose that the function c defining the constraints in (1.4) is a submersion on Ω, which means that its Jacobian matrix $A(x)$ is surjective for all x in Ω. Then, for any $x \in \Omega$, the set

$$\mathcal{M}_x = \{y \in \Omega : c(y) = c(x)\}$$

forms a smooth submanifold of $\mathbb{R}^n$, having dimension $n-m$ (for the geometrical tools, we refer the reader to Spivak [38], Boothby [3], or Conlon [9], for example).

We quote the fact that the columns of the basis $Z^-(x)$ introduced in section 1 span the space tangent to $\mathcal{M}_x$ at x and that the columns of the right inverse $A^-(x)$ span a space complementary to this tangent space. The matrix $Z(x)$ defined by (1.5) is also characterized by the useful identity

$$A^-(x)A(x) + Z^-(x)Z(x) = I,$$

which allows us to decompose a direction d of $\mathbb{R}^n$ in its *transversal* component $A^-(x)$ $A(x)d$ and its *longitudinal* component $Z^-(x)Z(x)d$. Also,

$$Z(x) = \begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} Z^-(x) & A^-(x) \end{pmatrix}^{-1},$$

so that the map $x \mapsto Z(x)$ inherits the smoothness of Z^- and A^-.

We assume that there is a pair $(x_*, \lambda_*) \in \Omega \times \mathbb{R}^m$ satisfying the sufficient second order conditions of optimality, i.e.,

$$\begin{cases} c(x_*) = 0, \\ \nabla f(x_*) + A(x_*)^\top \lambda_* = 0, \end{cases}$$

and $h^\top L_* h > 0$ for all nonzero $h \in N(A(x_*))$. By these assumptions, the reduced Hessian of the Lagrangian at the solution $B_* = Z^-(x_*)^\top L_* Z^-(x_*)$ is positive definite. We also introduce

$$\lambda(x) = -A^-(x)^\top \nabla f(x), \tag{2.1}$$

which estimates the Lagrange multiplier at the solution: $\lambda(x_*) = \lambda_*$.

We recall that we use the penalty function Θ_{σ_k} defined in (1.12) to globalize the local method (1.6)–(1.7). The *penalty parameter* σ_k depends on the iteration index k and is updated to satisfy at each iteration

$$\sigma_k \geq \|\lambda_k\|_D + \overline{\sigma}, \tag{2.2}$$

where $\lambda_k = \lambda(x_k)$ and $\overline{\sigma}$ is a fixed positive number. We have denoted by $\|\cdot\|_D$ the dual norm of the norm $\|\cdot\|$ used in (1.12). It is defined by

$$\|v\|_D = \sup_{\|u\|=1} u^\top v.$$

The manifolds on which the reduced gradient g is constant are denoted by

$$\mathcal{N}_x = \{y \in \Omega : g(y) = g(x)\}.$$

These sets are indeed manifolds if Ω is sufficiently "small," because g is a submersion in a neighborhood of x_*. To see this, observe that $g'(x_*) = Z_*^{-\top} L_*$ (see Stoer [39] or Nocedal and Overton [31]) and that $Z_*^{-\top} L_*$ is surjective.

We denote by $\xi'(u; v)$ the directional derivative of a function ξ at u along the direction v. In particular, if ξ is a function of a real variable α, $\xi'(\alpha; 1)$ denotes its right derivative. We quote the fact that if $\mathcal{C}$ is a convex continuous function and if ξ has directional derivatives, then $\mathcal{C} \circ \xi$ also has directional derivatives ("$\circ$" denotes composition). This can be seen by using the local Lipschitz continuity of $\mathcal{C}$, implied by its continuity (see Theorem 10.4 in [36] or Theorem IV.3.1.2 in [24]). As a result, when the constraint function c is smooth, Θ_σ defined in (1.12) has directional derivatives.

The following identity will be used several times. If f and c are smooth and $h \in \mathbb{R}^{n-m}$, we have for Θ_σ defined by (1.12)

$$\Theta'_\sigma\Big(x; Z^-(x)h - A^-(x)c(x)\Big) = g(x)^\top h + \lambda(x)^\top c(x) - \sigma\|c(x)\|. \tag{2.3}$$

Indeed, function f in Θ_σ gives the first two terms in the right-hand side of (2.3) (use the definition of $g(x)$ and (2.1)). Next, taking the notation $\eta(x) = \|x\|$ and knowing that $(\eta \circ c)'(x; v) = \eta'(c(x); A(x)v)$, the directional derivative of the second term in Θ_σ is given by

$$\begin{aligned}\sigma(\eta \circ c)'\Big(x; Z^-(x)h - A^-(x)c(x)\Big) &= \sigma\eta'(c(x); -c(x))\\ &= \sigma \lim_{t\to 0+} \frac{1}{t}\Big(\|c(x) - tc(x)\| - \|c(x)\|\Big)\\ &= -\sigma\|c(x)\|.\end{aligned}$$

3. The search algorithm. In unconstrained optimization, the path $p_k : \alpha \in \mathbb{R}_+ \mapsto p_k(\alpha)$ starting at the current iterate $p_k(0) = x_k \in \Omega$ and along which a step-size is taken is most commonly a straight line, which can be determined before the search begins. When constraints are present, a search along a line is no longer possible if one aims at satisfying the *reduced Wolfe conditions*

$$\Theta_{\sigma_k}(p_k(\alpha)) \le \Theta_{\sigma_k}(x_k) + \omega_1\, \nu_k(\alpha), \tag{3.1}$$

$$g(p_k(\alpha))^\top Z_k t_k \ge \omega_2\, g_k^\top Z_k t_k \tag{3.2}$$

for some $\alpha > 0$. In (3.1) and (3.2), the constants ω_1 and ω_2 are chosen in $(0, 1)$, and $\alpha \mapsto \nu_k(\alpha)$ is a function forcing the decrease of Θ_{σ_k} by the properties

$$\begin{cases} \nu_k(0) = 0 \\ \Theta'_{\sigma_k}(x_k; p'_k(0; 1)) \le \nu'_k(0; 1) < 0. \end{cases} \tag{3.3}$$

These properties and $\omega_1 < 1$ make it possible to realize (3.1) for small positive α. We have assumed that p_k is a descent path for Θ_{σ_k}, i.e., $\Theta'_{\sigma_k}(x_k; p'_k(0; 1)) < 0$.

In our proposal, the description of the search path is not as easy as in unconstrained optimization, because it depends on some intermediate step-sizes. From the point of view taken here, a reorientation of the search path is indeed necessary at some intermediate step-sizes α_k^i, $i = 1, \dots, i_k - 1$. Furthermore, condition (3.1) also depends on the step-sizes α_k^i through the function ν_k, which cannot be given before the search is completed. For these reasons, we have to specify simultaneously the function ν_k and the way the search path is designed.

The algorithm we discuss has some similarities with the one given in [17], but here the path has at once a longitudinal and a transversal component. More basically, one can see it as an extension of the method proposed by Fletcher [13] and Lemaréchal [25] for finding a Wolfe point in unconstrained optimization. With the option $\rho_k^i = 1$ below, the algorithm is related to the search technique of Moré and Sorensen [28] for realizing the strong Wolfe conditions for unconstrained problems.

3.1. Guiding paths. Before giving a precise description of the search algorithm, we would like to show by some observations why trying to realize conditions (3.1) and (3.2) simultaneously can succeed. On the way, we exhibit conditions under which our search technique should be numerically efficient.

First, let us introduce a path $\alpha \mapsto \bar{p}_k(\alpha)$ as a solution of the following differential equation:

$$\begin{cases} \bar{p}'_k(\alpha) = Z^-(\bar{p}_k(\alpha))Z_k t_k, \\ \bar{p}_k(0) = x_k. \end{cases} \tag{3.4}$$

This trajectory belongs to the manifold $\mathcal{M}_k = \mathcal{M}_{x_k}$ because multiplying the first equation in (3.4) by $A(\bar{p}_k(\alpha))$ gives $(c \circ \bar{p}_k)'(\alpha) = 0$, which means that c remains

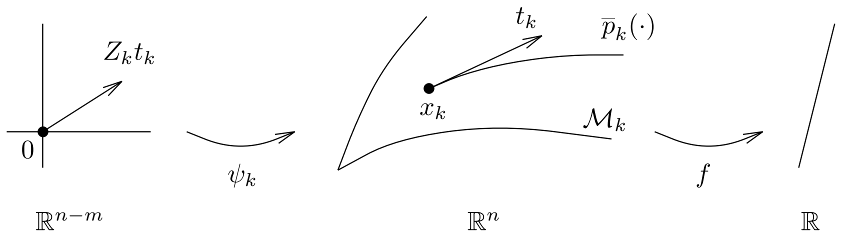


FIG. 3.1. *An interpretation of the longitudinal guiding path.*

constant along the path. As quoted in [17], if this path is defined for sufficiently large α and if f is bounded from below on $\mathcal{M}_k$, there exists a step-size $\overline{\alpha}_k$ such that (here $\omega_1 < \omega_2$ is necessary)

$$\Theta_{\sigma_k}(\overline{p}_k(\overline{\alpha}_k)) \le \Theta_{\sigma_k}(x_k) + \omega_1\, \overline{\alpha}_k\, g_k^\top Z_k t_k, \tag{3.5}$$

$$g(\overline{p}_k(\overline{\alpha}_k))^\top Z_k t_k \ge \omega_2\, g_k^\top Z_k t_k. \tag{3.6}$$

This can be seen by considering the standard Wolfe [43, 44] conditions (recalled in the introduction) on the function

$$\alpha \mapsto (\Theta_{\sigma_k} \circ \overline{p}_k)(\alpha) = (f \circ \overline{p}_k)(\alpha) + \sigma_k \|c_k\|.$$

Indeed, using (2.3), the derivative of this map at $\overline{\alpha}_k$ is $g(\overline{p}_k(\overline{\alpha}_k))^\top Z_k t_k$, the left-hand side of (3.6). Note that condition (3.5) has the form (3.1) with a linear function ν_k.

Locally, the search along $\overline{p}_k$ also has the following geometrical interpretation, illustrated in Figure 3.1. Suppose that there exists a parametrization $\psi_k : U \subset \mathbb{R}^{n-m} \to \mathcal{M}_k \subset \mathbb{R}^n$ of $\mathcal{M}_k$ around x_k such that $0 \in U$, $\psi_k(0) = x_k$, and

$$\psi_k'(u) = Z^-(\psi_k(u)) \quad \forall\ u \in U. \tag{3.7}$$

The existence of such parametrization depends on the choice of the tangent basis map Z^- (see [19]). Then, it is easy to see that

$$\overline{p}_k(\alpha) = \psi_k(\alpha Z_k t_k).$$

Indeed, denoting $q_k(\alpha) = \psi_k(\alpha Z_k t_k)$, we have $q_k'(\alpha) = Z^-(q_k(\alpha)) Z_k t_k$, by (3.7), and $q_k(0) = x_k$; hence, q_k satisfies the differential equation (3.4), which implies that $q_k = \overline{p}_k$. As a result, $(f \circ \psi_k)(\alpha Z_k t_k) = (\Theta_{\sigma_k} \circ \overline{p}_k)(\alpha) - \sigma_k \|c_k\|$ and $\nabla(f \circ \psi_k)(\alpha Z_k t_k) = g(\overline{p}_k(\alpha))$, so that the search to realize (3.5)–(3.6) can now be seen as a standard Wolfe search on the function $f \circ \psi_k$ starting at $0 \in \mathbb{R}^{n-m}$ along the reduced direction $Z_k t_k$. From this interpretation, we define the *reduced* (*longitudinal*) *displacement* from x_k to $\overline{p}_k(\overline{\alpha}_k)$ as the vector $\overline{\delta}_k = \overline{\alpha}_k Z_k t_k$.

The path $\alpha \mapsto \overline{p}_k(\alpha)$ shows that there is at least one way of generalizing the Wolfe conditions to equality constrained problems. We call this path the *longitudinal guiding path*; we say longitudinal because its image

$$\overline{\mathcal{P}}_k = \{\overline{p}_k(\alpha) : \alpha \ge 0 \text{ and } \overline{p}_k(\alpha) \text{ exists}\}$$

lies in $\mathcal{M}_k$. This trajectory can be used as a guide for designing a search path having points satisfying (3.5)–(3.6) but easier to compute than $\overline{p}_k(\cdot)$; see [17].

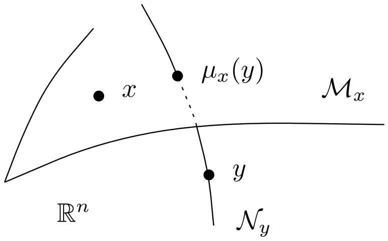


Fig. 3.2. *The map μ_x.*

In this paper, we follow the same strategy and introduce a smooth guiding path having longitudinal and transversal components, i.e., neither c nor g is constant along the path. Later, a discretization will be introduced. We proceed step by step and begin by some definitions.

Definition 3.1. *Let F be the function*

$$F : \Omega \to \mathbb{R}^n : x \mapsto \begin{pmatrix} c(x) \\ g(x) \end{pmatrix}.$$

Definition 3.2. *Let us also introduce an open subset Ω_0 of Ω such that:* (i) *$x_* \in \Omega_0$;* (ii) *$F'(x)$ is nonsingular when $x \in \Omega_0$;* (iii) *$F(\Omega_0)$ has the form $U_0 \times V_0$, where U_0 and V_0 are open sets in $\mathbb{R}^m$ and $\mathbb{R}^{n-m}$, respectively;* (iv) *$F : \Omega_0 \to U_0 \times V_0$ is a diffeomorphism.*

Note that such open subset Ω_0 always exists when $B_* = Z^-(x_*)^\top L_* Z^-(x_*)$ is nonsingular, which we assume. Indeed, in this case, recalling that $g'(x_*) = Z_*^{-\top} L_*$, we see that $F'(x_*)$ is nonsingular, so conditions (i)–(iv) are satisfied for some (possibly large) neighborhood Ω_0 of x_*.

Definition 3.3. *For a point x fixed in Ω_0, we introduce the map*

$$\mu_x : \Omega_0 \to \mathcal{M}_x \cap \Omega_0,$$

defined in the following way. For $y \in \Omega_0$, $\mu_x(y) \in \mathcal{M}_x \cap \Omega_0$ is defined as the unique *point in $\mathcal{M}_x \cap \mathcal{N}_y \cap \Omega_0$ ($\mathcal{N}_y$ is the reduced gradient manifold containing y); see Figure* 3.2.

To see that the set $\mathcal{M}_x \cap \mathcal{N}_y \cap \Omega_0$ is formed of just one point, note that $x \in \Omega_0$ and $y \in \Omega_0$ imply that $(c(x), g(y)) \in U_0 \times V_0 = F(\Omega_0)$. As F is a diffeomorphism on Ω_0, $\mathcal{M}_x \cap \mathcal{N}_y \cap \Omega_0 = F^{-1}((c(x), g(y))) \cap \Omega_0$ is a singleton. As we see, μ_x maps a point $y \in \Omega_0$ to a point in $\mathcal{M}_x \cap \Omega_0$ by following the manifold of constant reduced gradient $\mathcal{N}_y$. The following result will be useful.

Proposition 3.4. *Suppose that c and g are of class C^l ($l \geq 1$) on Ω_0 and let $x \in \Omega_0$. Then, $\mu_x : \Omega_0 \to \mathcal{M}_x \cap \Omega_0$ is of class C^l and, as a function with values in $\mathbb{R}^n$, its Jacobian matrix at $y \in \Omega_0$ is given by*

$$\mu_x'(y) = \widetilde{Z}^-(z)\Big(g'(z)\widetilde{Z}^-(z)\Big)^{-1} g'(y), \tag{3.8}$$

where $z = \mu_x(y)$ and $\widetilde{Z}^-(z)$ is an arbitrary basis of the space tangent to $\mathcal{M}_x$ at z (one can take $\widetilde{Z}^-(z) = Z^-(z)$).

Proof. To show that μ_x is of class C^l, we "read" this map with appropriate C^l-compatible coordinate charts. Let us take $(U, \varphi) = (\Omega_0, F)$ as a chart of Ω_0 at y and

$(V, \psi) = (\mathcal{M}_x \cap \Omega_0, g|_{\mathcal{M}_x \cap \Omega_0})$ as a chart of $\mathcal{M}_x \cap \Omega_0$ at $z = \mu_x(y)$. These coordinate charts are C^l because c and g are C^l. Then, μ_x is read with φ and ψ as

$$\psi \circ \mu_x \circ \varphi^{-1},$$

which is the C^∞ map $\mathbb{R}^n \to \mathbb{R}^{n-m} : (u_1, \dots, u_n) \mapsto (u_{m+1}, \dots, u_n)$. This shows that μ_x is of class C^l.

Since c is of class C^l, the canonical injection $j : \mathcal{M}_x \cap \Omega_0 \to \mathbb{R}^n$ is of class C^l, and μ_x with values in $\mathbb{R}^n$ (more precisely, $j \circ \mu_x$) is also of class C^l. Then, we can differentiate the identities

$$c(\mu_x(y)) = c(x) \quad \text{and} \quad g(\mu_x(y)) = g(y)$$

with respect to y. This gives, with $z = \mu_x(y)$,

$$A(z)\mu_x'(y) = 0 \quad \text{and} \quad g'(z)\mu_x'(y) = g'(y).$$

To solve this system in $\mu_x'(y)$, we introduce an arbitrary basis $\widetilde{Z}^-(z)$ of the null space of $A(z)$. From the first identity, we see that $\mu_x'(y) = \widetilde{Z}^-(z)M$ for some $(n-m) \times n$ matrix M. Then, by the nonsingularity of $F'(z)$ when $z \in \Omega_0$ (see Definition 3.2), $g'(z)\widetilde{Z}^-(z)$ is nonsingular and the second identity above leads to (3.8). □

Let us now go back to our problem of designing a suitable path $\alpha \mapsto \tilde{p}_k(\alpha)$, with longitudinal and transversal components. Suppose we ensure that its image by μ_{x_k} lies in $\overline{\mathcal{P}}_k$; i.e.,

$$\mu_{x_k}(\tilde{p}_k(\alpha)) \in \overline{\mathcal{P}}_k \quad \text{for } \alpha \geq 0.$$

We recall that $\bar{\alpha}_k$ is some step-size such that (3.5)–(3.6) hold. Then, if $\tilde{p}_k(\alpha)$ exists for sufficiently large α, it is reasonable to expect to find some positive $\tilde{\alpha}_k$ such that $g(\tilde{p}_k(\tilde{\alpha}_k)) = g(\overline{p}_k(\bar{\alpha}_k))$—this assumes that the path $\tilde{p}_k$ does not blow up for a finite longitudinal displacement. Using (3.6), we obtain

$$g(\tilde{p}_k(\tilde{\alpha}_k))^\top Z_k t_k \geq \omega_2\, g_k^\top Z_k t_k.$$

This shows that condition (3.2) can be satisfied along a path not belonging to $\mathcal{M}_k$.

For two reasons, this is not enough, however, to have a satisfactory search. First, the two conditions (3.1) and (3.2) have to be satisfied simultaneously. Second, if we want to minimize approximation errors by updating the matrix with $\tilde{\gamma}_k = g(\tilde{p}_k(\tilde{\alpha}_k)) - g_k = g(\overline{p}_k(\bar{\alpha}_k)) - g_k$ and $\tilde{\delta}_k = \tilde{\alpha}_k Z_k t_k$, we also need to have $\tilde{\alpha}_k = \bar{\alpha}_k$ so that the changes in the reduced gradient along $\overline{p}_k$ and $\tilde{p}_k$ will correspond to the same reduced displacement $\bar{\delta}_k = \bar{\alpha}_k Z_k t_k$.

This latter condition will be satisfied if we build a path $\alpha \mapsto \tilde{p}_k(\alpha)$ such that

$$g(\tilde{p}_k(\alpha)) = g(\overline{p}_k(\alpha)) \tag{3.9}$$

for all α for which $\tilde{p}_k(\alpha)$ and $\overline{p}_k(\alpha)$ exist. In the next proposition, we show that this can be achieved when $\tilde{p}_k(\cdot)$ is defined as a solution of the following differential equation:

$$\begin{cases} \tilde{p}_k'(\alpha) = Z^-(\tilde{p}_k(\alpha))Z_k t_k - A^-(\tilde{p}_k(\alpha))c(\tilde{p}_k(\alpha)), \\ \tilde{p}_k(0) = x_k, \end{cases} \tag{3.10}$$

and the maps $Z^-(\cdot)$ and $A^-(\cdot)$ are chosen such that

$$\left\{ \begin{array}{l} g'A^- = 0, \\ g'Z^- \text{ is constant on the reduced gradient manifolds.} \end{array} \right. \tag{3.11}$$

The first condition in (3.11) requires that the transversal displacements (in the range space of A^-) be in the space tangent to the reduced gradient manifold. The matrix $g'Z^-$ appearing in (3.11) is the matrix from which information is collected by the pair (γ_k, δ_k) in (1.10). At the solution, it is also the reduced Hessian of the Lagrangian. The second condition of (3.11) requires that this matrix be constant along the reduced gradient manifolds.

PROPOSITION 3.5. *Suppose that c and g are continuously differentiable on the set Ω_0 introduced in Definition 3.2, that $x_k \in \Omega_0$, and that the maps Z^- and A^- are such that (3.11) holds on Ω_0. Consider the paths $\bar{p}_k$ and $\tilde{p}_k$ defined by (3.4) and (3.10), respectively. Then, (3.9) holds as long as both $\bar{p}_k(\alpha)$ and $\tilde{p}_k(\alpha)$ exist in Ω_0.*

Proof. Let us define $q_k = \mu_{x_k} \circ \tilde{p}_k$, a path in $\mathcal{M}_k$. This path is well defined as long as $\tilde{p}_k$ exists in Ω_0. By the definition of μ_{x_k}, $g(q_k(\alpha)) = g(\tilde{p}_k(\alpha))$. Hence, we just have to prove that $q_k = \bar{p}_k$.

Note that since $\tilde{p}_k(\alpha)$ and $q_k(\alpha)$ belong to the same reduced gradient manifold, the second condition in (3.11) gives

$$g'(\tilde{p}_k(\alpha))Z^-(\tilde{p}_k(\alpha)) = g'(q_k(\alpha))Z^-(q_k(\alpha)). \tag{3.12}$$

Now, by Proposition 3.4, we see that q_k is differentiable; by using $\widetilde{Z}^- = Z^-$ in (3.8), we have

$$\begin{aligned} q_k'(\alpha) &= \mu_{x_k}'(\tilde{p}_k(\alpha))\tilde{p}_k'(\alpha) \\ &= Z^-(q_k(\alpha))\Big(g'(q_k(\alpha))Z^-(q_k(\alpha))\Big)^{-1} g'(\tilde{p}_k(\alpha))Z^-(\tilde{p}_k(\alpha))Z_k t_k \\ &= Z^-(q_k(\alpha))Z_k t_k, \end{aligned}$$

where we also used (3.10), the first condition in (3.11), and (3.12). Therefore, q_k satisfies the same differential equation as $\bar{p}_k$, with the same initial condition x_k at $\alpha = 0$ (see (3.4)). Hence, $q_k = \bar{p}_k$ and the proposition is proved. □

We call the path defined by $\tilde{p}_k$, the solution of (3.10), the *bicomponent guiding path*. The actual PLS path introduced in section 3.2 will be a discretization of this one. From Proposition 3.5 and the discussion that precedes it, one can say that *the PLS should be numerically efficient when Z^- and A^- are chosen such that* (3.7) *holds for some parametrization ψ_k and* (3.11) *holds.*

If (3.7) can always be realized by choosing suitable tangent bases (see [19]), it is unrealistic to ask the user to realize (3.11), because the computation of g' requires the evaluation of second derivatives, which are not available in the quasi-Newton framework. Also, we shall not assume that (3.11) holds and, therefore, (3.9) may not hold either, even at the first order. Figure 3.3 represents a still rather favorable situation without (3.9); it is favorable because the dashed curve in $\mathcal{M}_k$ and in the reduced space is still rather close to the solid curve. On the other hand, we shall keep the path defined by (3.10). As we shall see below (Proposition 3.6), no matter the realization of (3.9), one can satisfy the reduced Wolfe conditions (3.1)–(3.2) along the path $\tilde{p}_k$ for an appropriate function ν_k. Of course, an update of the matrix without (3.9) may not be safe. We believe, however, that it could be the role of the update

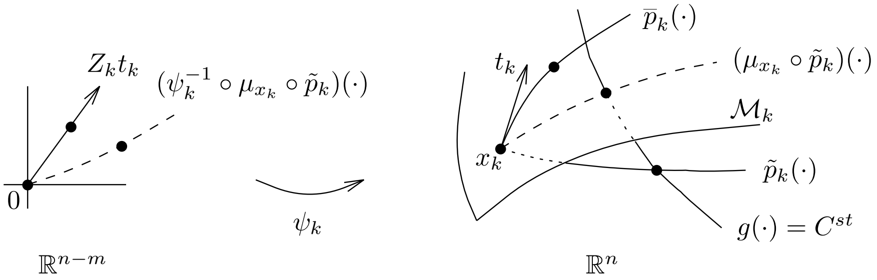


FIG. 3.3. *The bicomponent guiding path.*

criterion to detect situations where (3.11) is not violated by much. Nevertheless, the update criteria introduced by Nocedal and Overton [31] and Gilbert [16], as well as the one used in the numerical experiments below, are based on a condition different from (3.11).

Let us now show how to realize (3.1)–(3.2) along the path $p_k = \tilde{p}_k$ without condition (3.9). For this, we take for ν_k in (3.1) the function defined by

$$\nu_k(\alpha) = \Theta_{\sigma_k}(\tilde{p}_k(\alpha)) - \Theta_{\sigma_k}(x_k), \tag{3.13}$$

with $\tilde{p}_k$ given by (3.10). Conditions (3.3) are satisfied for this choice, provided that σ_k is sufficiently large. Then (3.1) is equivalent to requiring that

$$\Theta_{\sigma_k}(\tilde{p}_k(\alpha)) \leq \Theta_{\sigma_k}(x_k). \tag{3.14}$$

At this point, function ν_k does not look very useful, since it no longer appears in the descent condition (3.14). But this is only true in the present smooth case. In the discretized version of the search algorithm, it is (3.1) with (3.13) that will be discretized, not (3.14), so that terms coming from the discretization will force the merit function to decrease.

Requiring (3.14) is not very demanding, but it gives the time for (3.2) to be realized before violating (3.1); this is shown in Proposition 3.6 below. Note that the result of this proposition can be obtained without the inequality $\omega_1 < \omega_2$ (ω_1 is not used in the statement of the proposition).

PROPOSITION 3.6. *Suppose that the path $\alpha \mapsto \tilde{p}_k(\alpha)$ defined by* (3.10) *exists for sufficiently large step-size $\alpha \geq 0$, that Θ_{σ_k} is bounded from below along this path, that $\sigma_k \geq \|\lambda(\tilde{p}_k(\alpha))\|_D$ whenever $\tilde{p}_k(\alpha)$ exists, and that $\omega_2 \in (0,1)$. Then, the inequalities*

$$\Theta_{\sigma_k}(\tilde{p}_k(\alpha)) \leq \Theta_{\sigma_k}(x_k), \tag{3.15}$$

$$g(\tilde{p}_k(\alpha))^\top Z_k t_k \geq \omega_2\, g_k^\top Z_k t_k \tag{3.16}$$

are satisfied for some $\alpha > 0$.

Proof. We recall that if ξ_1 and ξ_2 are continuous functions on an interval $[a, b]$ having right derivatives on (a, b) with $\xi_1'(\alpha; 1) \leq \xi_2'(\alpha; 1)$ for all $\alpha \in (a, b)$, then $\xi_1(b) - \xi_1(a) \leq \xi_2(b) - \xi_2(a)$ (see, for instance, Schwartz [37, Chapter III, section 5, Remark 3]). Also, as in the proof of (2.3), denoting the norm $\|\cdot\|$ by η, we have from (3.10)

$$\begin{aligned}(\eta \circ c \circ \tilde{p}_k)'(\alpha; 1) &= \eta'(c(\tilde{p}_k(\alpha)); (c \circ \tilde{p}_k)'(\alpha))\\ &= \eta'(c(\tilde{p}_k(\alpha)); -c(\tilde{p}_k(\alpha)))\\ &= -\|c(\tilde{p}_k(\alpha))\|.\end{aligned}$$

Hence, using (2.3) and $\sigma_k \geq \|\lambda(\tilde{p}_k(\alpha))\|_D$, we obtain

$$\begin{aligned}&(\Theta_{\sigma_k} \circ \tilde{p}_k)'(\alpha; 1)\\ &\quad = g(\tilde{p}_k(\alpha))^\top Z_k t_k + \lambda(\tilde{p}_k(\alpha))^\top c(\tilde{p}_k(\alpha)) - \sigma_k \|c(\tilde{p}_k(\alpha))\|\\ &\quad \leq g(\tilde{p}_k(\alpha))^\top Z_k t_k. \end{aligned}\tag{3.17}$$

Then, the result of the proposition is clear when $g_k = 0$, because (3.16) readily holds ($t_k = 0$) and (3.17) implies that $(\Theta_{\sigma_k} \circ \tilde{p}_k)'(\alpha; 1) \leq 0$ for small $\alpha \geq 0$ (those for which $\tilde{p}_k(\alpha)$ exists). Therefore, (3.15) is satisfied for small $\alpha \geq 0$.

Suppose now that $g_k \neq 0$. Since $g_k^\top Z_k t_k < 0$ and $\omega_2 < 1$, (3.16) is not verified for small positive α, so there is a nonempty interval of the form $(0, \overline{\alpha}]$ on which (3.16) is false. Now, when (3.16) is not verified, one has from (3.17) that

$$(\Theta_{\sigma_k} \circ \tilde{p}_k)'(\alpha; 1) \leq \omega_2\, g_k^\top Z_k t_k.$$

Therefore, we obtain

$$\Theta_{\sigma_k}(\tilde{p}_k(\alpha)) - \Theta_{\sigma_k}(x_k) \leq \omega_2\, \alpha\, g_k^\top Z_k t_k \quad \text{for } \alpha \in (0, \overline{\alpha}].$$

Hence, (3.15) is trivially satisfied on $(0, \overline{\alpha}]$. On the other hand, because of this last inequality and the fact that $\alpha \mapsto \Theta_{\sigma_k}(\tilde{p}_k(\alpha))$ is bounded below, the interval $(0, \overline{\alpha}]$ cannot be arbitrarily large. Therefore, (3.16) must eventually be satisfied. At the first step-size $\alpha > 0$ for which (3.16) holds, (3.15) is still verified by continuity. The proposition is proved. □

We are now ready to describe the actual search path, which may be seen as an explicit Euler approximation of the solution of (3.10) with well-chosen discretization points. Similarly, the actual function ν_k is not given by (3.13) (with which global convergence could not be obtained) but is a piecewise linear approximation of this function with the same discretization points. A successful idea is to introduce a discretization point α_k^i only when the discretized form of (3.1) holds for $\alpha = \alpha_k^i$.

3.2. The search algorithm. We assume that the current iterate x_k is in Ω and that it is not stationary: $\|g_k\| + \|c_k\| \neq 0$. Let constants ω_1 and ω_2 be given in $(0, 1)$.

The search algorithm is iterative and generates, for $i = 0, \ldots, i_k - 1$, intermediate step-size candidates α_k^i, points x_k^i, descent directions d_k^i of Θ_{σ_k} at x_k^i, piecewise linear search paths p_k^i, and piecewise linear forcing functions ν_k^i. These functions ν_k^i, playing the role of ν_k in (3.1), may be taken as discontinuous. The following conditions will hold for $i = 1, \ldots, i_k - 1$:

$$x_k^i \in \Omega, \tag{3.18a}$$

$$\Theta_{\sigma_k}(x_k^i) \leq \Theta_{\sigma_k}(x_k) + \omega_1\, \nu_k^{i-1}(\alpha_k^i), \tag{3.18b}$$

$$g(x_k^i)^\top Z_k t_k < \omega_2\, g_k^\top Z_k t_k, \tag{3.18c}$$

$$\sigma_k \geq \|\lambda(x_k^i)\|_D + \overline{\sigma}. \tag{3.18d}$$

Inequality (3.18b) means that descent is forced at each iteration, while (3.18c) means that the curvature condition does not hold. Note that (3.18c) implies that x_k^i is not stationary.

At the beginning, the iteration index i is set to 0, $\alpha_k^0 = 0$, and $x_k^0 = x_k$. To initialize the recurrence, we define $\nu_k^{-1}(0) = 0$. It is also assumed that B_k is positive definite and that $\sigma_k \geq \|\lambda_k\|_D + \bar{\sigma}$. Then, (3.18a, b, d) clearly hold for $i = 0$. Stage i ($i \geq 0$) of the search comprises the following steps.

Stage i of the PLS algorithm.

1. Choose a tangent scaling factor $\tau_k^i > 0$ and compute the direction d_k^i defined by

$$d_k^i = \tau_k^i Z^-(x_k^i) Z_k t_k - A^-(x_k^i) c(x_k^i). \tag{3.19}$$

Update the search path p_k^i:

$$p_k^i(\alpha) = \begin{cases} p_k^{i-1}(\alpha) & \text{for } 0 \leq \alpha < \alpha_k^i \\ x_k^i + (\alpha - \alpha_k^i) d_k^i & \text{for } \alpha \geq \alpha_k^i. \end{cases}$$

Update the function ν_k^i (see below).

2. Determine a step-size $\alpha_k^{i+1} > \alpha_k^i$ from x_k^i along d_k^i such that

$$x_k^{i+1} = x_k^i + (\alpha_k^{i+1} - \alpha_k^i) d_k^i \tag{3.20}$$

is in Ω and the descent condition

$$\Theta_{\sigma_k}(p_k^i(\alpha)) \leq \Theta_{\sigma_k}(x_k) + \omega_1 \, \nu_k^i(\alpha) \tag{3.21}$$

holds for $\alpha = \alpha_k^{i+1}$.

3. If $i = 0$ and some (unspecified) update criterion does not hold, set $i_k = 1$, $\alpha_k = \alpha_k^1$, $x_{k+1} = x_k^1$, $p_k = p_k^0$, $\nu_k = \nu_k^0$ and quit the PLS algorithm.

4. Linearize the constraints at x_k^{i+1} and test the curvature condition

$$g(x_k^{i+1})^\top Z_k t_k \geq \omega_2 \, g_k^\top Z_k t_k. \tag{3.22}$$

If the latter holds, set $i_k = i + 1$, $\alpha_k = \alpha_k^{i+1}$, $x_{k+1} = x_k^{i+1}$, $p_k = p_k^i$, $\nu_k = \nu_k^i$, and quit the PLS algorithm.

5. If the penalty parameter σ_k is not sufficiently large to have

$$\sigma_k \geq \|\lambda(x_k^{i+1})\|_D + \bar{\sigma}, \tag{3.23}$$

set $i_k = i + 1$, $\alpha_k = \alpha_k^{i+1}$, $x_{k+1} = x_k^{i+1}$, $p_k = p_k^i$, $\nu_k = \nu_k^i$, and quit the PLS algorithm.

Let us give more details on the steps of the algorithm.

Step 1. The factor $\tau_k^i > 0$ scales the tangential component of the direction d_k^i. One reason for introducing this factor is that it may be convenient to use different step-sizes for the transversal and longitudinal part of the displacement. Indeed, second order information is used transversally, while a quasi-Newton model is used longitudinally. Another reason for using different transversal and longitudinal step-sizes will come from the discussion in section 3.6.

When $i = 0$ (initially) and $\tau_k^0 = 1$, d_k^0 has the form of the reduced SQP direction d_k and is tangent to $\tilde{p}_k$ at 0. For $i \geq 1$, the direction comes from the discretization of (3.10): its longitudinal component $\tau_k^i Z^-(x_k^i) Z_k t_k$ is tangent to $\mathcal{M}_{x_k^i}$ at x_k^i, and the unscaled reduced direction $Z_k t_k$ is kept unchanged from one stage to the other.

The update formula of ν_k^{i-1} includes two natural possibilities. They correspond to the setting of the parameter ρ_k^i to 0 or 1 below. So, let ρ_k^i be any number in $[0,1]$ and let us introduce the notion of *total decrease* of Θ_{σ_k} at x_k^i as the positive quantity

$$T_k^i = \Theta_{\sigma_k}(x_k) - \Theta_{\sigma_k}(x_k^i). \tag{3.24}$$

Then, ν_k^i is defined by

$$\nu_k^i(\alpha) = \begin{cases} \nu_k^{i-1}(\alpha) \quad \text{for } 0 \leq \alpha < \alpha_k^i, \\ (1-\rho_k^i)\nu_k^{i-1}(\alpha_k^i) \\ \qquad +\rho_k^i(-T_k^i/\omega_1) + (\alpha - \alpha_k^i)\Theta'_{\sigma_k}(x_k^i; d_k^i) \quad \text{for } \alpha \geq \alpha_k^i. \end{cases} \tag{3.25}$$

When $\rho_k^i = 0$, ν_k^i is continuous and the search can be viewed as a discretization of the smooth search described in section 3.1. This corresponds to a loose search. When $\rho_k^i = 1$, the search is closer to the "skipping rule" strategy discussed in section 3.3 below. It is also more demanding, since ν_k^i is more negative (use (3.18b)).

Step 2. Observe that d_k^i is a descent direction of Θ_{σ_k} at x_k^i, since by (2.3)

$$\Theta'_{\sigma_k}(x_k^i; d_k^i) = \tau_k^i g(x_k^i)^\top Z_k t_k + \lambda(x_k^i)^\top c(x_k^i) - \sigma_k \|c(x_k^i)\|, \tag{3.26}$$

which is negative when (3.18c) and (3.18d) hold. Then, it is standard to verify that, with conditions (3.18a), (3.18b), and $\omega_1 \in (0,1)$, one can find a step-size $\alpha = \alpha_k^{i+1} > \alpha_k^i$ such that x_k^{i+1} is in Ω and the descent condition (3.21) holds.

Step 3. If $i = 0$, it is the right place to ask whether the pursuit of the search is useful. Indeed, unlike in unconstrained optimization, the curvature condition (3.2) is not strong enough to force global convergence (see section 4). It is only useful for guaranteeing the positive definiteness of the generated matrices. On the other hand, the role of the update criterion is to judge whether an update is appropriate by appreciating the quality of the information contained in the pair (γ_k, δ_k). We believe that this appreciation has to be done when $i = 0$ so that a PLS is not launched without necessity. We shall not be more specific on this update criterion, because the results below do not need it. For these results, it can be any rule such as "never update" or "always update." A better rule is used, however, in the numerical experiments of section 5. For more information on this subject, see Nocedal and Overton [31] or Gilbert [16].

Step 4. By linearization of the constraints at a point x, we mean the computation of the Jacobian matrix $A(x)$, the basis $Z^-(x)$, and the right inverse $A^-(x)$. At step 4, the curvature condition (3.22) is tested. If it holds, the search terminates. From step 2 and (3.22), the point x_{k+1} is in Ω and satisfies the reduced Wolfe conditions (3.1)–(3.2) with $\alpha = \alpha_k$.

Step 5. If (3.22) is not satisfied, one has to check whether the penalty parameter is sufficiently large to continue the search from x_k^{i+1}, i.e., whether (3.23) holds. If such is the case, all the conditions in (3.18) hold and a new iteration can start after having increased i by one. Otherwise, the search is interrupted (another possibility would have been to increase σ_k and to pursue the search).

3.3. Additional comments.

To summarize, there are three facts that can interrupt the search algorithm: either (i) the update criterion does not hold in step 3 after α_k^1 is determined in step 2, or (ii) the conditions (3.1)–(3.2) are satisfied in step 4, or (iii) the penalty parameter σ_k is not large enough to guarantee that the next search direction is a descent direction of Θ_{σ_k} (step 5). We shall show in section 4 that under natural assumptions the algorithm does not cycle and terminates on one of these situations.

As announced above, when $\tau_k^i = 1$ for all i, the path p_k is a piecewise linear approximation of the bicomponent guiding path $\tilde{p}_k$, which was obtained by an explicit Euler discretization of the differential equation (3.10) at the step-sizes α_k^i. Furthermore, if $\rho_k^i = 0$ for all i, ν_k can also be viewed as a discretization of the function ν_k defined by (3.13) with p_k instead of $\tilde{p}_k$: $\nu_k'(\alpha_k^i;1) = \Theta_{\sigma_k}'(x_k^i;d_k^i) = (\Theta_{\sigma_k} \circ p_k)'(\alpha_k^i;1)$.

Remark that the search direction d_k^1 is close to

$$\check{d}_k^1 = -Z^-(x_k^1)B_k^{-1}g(x_k^1) - A^-(x_k^1)c(x_k^1),$$

which is the direction that would be taken in an algorithm skipping the update of B_k at x_k^1 when $\gamma_k^\top \delta_k$ is nonpositive or when the curvature condition does not hold (*skipping rule*). When $\rho_k^1 = 1$ in the definition of ν_k^1 above, inequality (3.21) becomes

$$\Theta_{\sigma_k}(x_k^1 + (\alpha - \alpha_k^1)d_k^1) \leq \Theta_{\sigma_k}(x_k^1) + \omega_1\,(\alpha - \alpha_k^1)\,\Theta_{\sigma_k}'(x_k^1;d_k^1),$$

which is also the condition to realize in an algorithm with skipping rule.

The only difference between $\check{d}_k^1$ and d_k^1 is that in the latter the reduced gradient is also kept unchanged. The main motivation for this choice is explained in section 3.1: if the matrices $A^-(x_k^i)$ and $Z^-(x_k^i)$ are good in the sense of (3.11), the search consists of minimizing $(f \circ \psi_k)$ along the reduced direction $Z_k t_k$ (the meaning of ψ_k is given in Figure 3.1). With this in mind, it makes sense to update the matrix B_k using the vectors

$$\gamma_k = g_{k+1} - g_k \quad \text{and} \quad \delta_k = \left(\sum_{i=0}^{i_k - 1} \tau_k^i (\alpha_k^{i+1} - \alpha_k^i) \right) Z_k t_k. \tag{3.27}$$

Note that when $\tau_k^i = 1$ for all i, $\delta_k = \alpha_k Z_k t_k$, simply.

When $\rho_k^i = 1$ for all i, the PLS algorithm applied to unconstrained problems ($c(x_k^i) = 0$ for all i) is related to the method of Moré and Sorensen [28] (see also Moré and Thuente [29, Section 2]). The differences are that Moré and Sorensen look for a point satisfying the *strong* Wolfe conditions (for this reason our method terminates more quickly), and the slope of the pieces of the forcing function ν_k^i is kept unchanged in their method (while we adapt it to the current point x_k^i).

For $i \geq 0$, we introduce the notion of *forced decreased* of Θ_{σ_k} at x_k^{i+1} as the positive quantity

$$F_k^{i+1} = -\,\omega_1 \sum_{l=0}^{i} (\alpha_k^{l+1} - \alpha_k^l)\Theta_{\sigma_k}'(x_k^l;d_k^l). \tag{3.28}$$

Using (3.18b) and the definition (3.25) of ν_k^i, we get, for $i \geq 1$,

$$F_k^i \leq -\,\omega_1\,\nu_k^{i-1}(\alpha_k^i) \leq T_k^i, \tag{3.29}$$

where T_k^i is the total decrease of Θ_{σ_k} defined by (3.24).

We conclude this section by some comments on the cost of the PLS algorithm. The main requirement of this method is the linearization of the constraints (the computation of A, A^-, and Z^-) at the intermediate points x_k^i $(1 \leq i \leq i_k - 1)$. This apparently damning cost must be reappreciated in view of the following two facts. First, we have shown in [18] that it is possible to combine the PLS technique with a suitable update criterion such that, asymptotically, each time the update criterion holds, the PLS algorithm succeeds without intermediate point ($i_k = 1$) and with unit step-size ($\alpha_k = 1$). Therefore, one can expect that in practice very few inner iterations will be necessary in the PLS algorithm. This is confirmed by the limited numerical experiments presented in section 5. Secondly, the work realized during the inner iterations of the PLS algorithm helps to find a better approximation of the solution: the search along the inner direction d_k^i makes the linearization at x_k^i useful. In fact, since d_k^i is close to a standard reduced SQP direction, one could consider all the intermediate iterates x_k^i as "true" iterates. It is a matter of presentation to group in a single iteration all the stages between two matrix updates.

3.4. Successive backtrackings. When a step-size candidate α_k^i is not accepted because inequality (3.22) does not hold, one has to determine the next tangent scaling factor $\tau_k^i > 0$ and the next step-size candidate α_k^{i+1} such that

$$\alpha_k^{i+1} > \alpha_k^i, \quad x_k^{i+1} = p_k^i(\alpha_k^{i+1}) \in \Omega, \quad \text{and} \quad (3.21) \text{ holds with } \alpha = \alpha_k^{i+1}.$$

This cannot be done in an uncontrolled manner. In particular, τ_k^i cannot be arbitrarily small or large, and α_k^{i+1} cannot be chosen too close to α_k^i. In this section, we describe a method for determining α_k^{i+1} that will ensure the finite termination of the search algorithm.

The determination can be divided into two stages. In the *forward* or *extrapolation* stage, a step-size $\alpha_k^{i,1} > \alpha_k^i$ is taken along d_k^i. The *backward* or *interpolation* stage is iterative: as long as (for the current trial with a step-size $\alpha_k^{i,j}$ ($j \geq 1$)) $p_k^i(\alpha_k^{i,j})$ is not in Ω or (3.21) does not hold for $\alpha = \alpha_k^{i,j}$, a new trial is made with a step-size $\alpha_k^{i,j+1} \in (\alpha_k^i, \alpha_k^{i,j})$. By requiring that $\alpha_k^{i,j}$ converges to α_k^i when $j \to \infty$, $p_k^i(\alpha_k^{i,j})$ will be in Ω and (3.21) with $\alpha = \alpha_k^{i,j}$ will hold for some finite index j. We denote by j_i the *first* index j for which this occurs and set

$$\alpha_k^{i+1} = \alpha_k^{i,j_i}.$$

We also suppose that $\{\alpha_k^{i,j}\}_{j\geq 1}$ does not tend too fast to α_k^i: the closer α_k^{i+1} is to α_k^i, the larger j_i must be. The rigorous form of our assumptions follows.

Assumptions 3.7. We suppose that the determination of the tangent scaling factor $\tau_k^i > 0$ and the step-sizes $\alpha_k^{i,j}$ is such that

(i) the sequences $\{\tau_k^i\}_{i\geq 0}$ and $\{1/\tau_k^i\}_{i\geq 0}$ are bounded,

(ii) the sequence $\{\alpha_k^{i,j}\}_{j\geq 1}$ converges to α_k^i,

(iii) if the increasing sequence $\{\alpha_k^i\}_{i\geq 1}$ converges to some step-size $\overline{\alpha}_k$, then

  (a) for any index $j' \geq 1$, there is an index $i' \geq 1$ such that $j_i \geq j'$ for all $i \geq i'$,

  (b) for any $j \geq 1$, the sequence $\{\alpha_k^{i,j}\}_{i\geq 1}$ converges to a step-size $\overline{\alpha}_k^{\infty,j} \neq \overline{\alpha}_k$,

  (c) the sequence $\{\overline{\alpha}_k^{\infty,j}\}_{j\geq 1}$ converges to $\overline{\alpha}_k$.

Assumption 3.7 (iii-a) means that when $\{\alpha_k^i\}_{i\geq 1}$ converges, the number $(j_i - 1)$ of interpolations must go to infinity when $i \to \infty$.

Assumption 3.7 (i) is not difficult to satisfy. On the other hand, an easy way of satisfying Assumptions 3.7 (ii) and (iii), while using its favorite extrapolation and interpolation formulas, is to use some safeguard rules. Here is an example of rules that guarantee Assumptions 3.7 (ii) and (iii).

Example of safeguard rules for $\alpha_k^{i,j}$.

1. Choose $\varepsilon_E > 0$ and $\varepsilon_I \in (0, 1/2)$.
2. Extrapolation safeguard: for $i \geq 0$, choose $\alpha_k^{i,1} \geq \alpha_k^i + \varepsilon_E$.

3. Interpolation safeguard: for $i \geq 0$ and $j \geq 2$, choose

$$\alpha_k^{i,j} \in \left[(1-\varepsilon_I)\alpha_k^i + \varepsilon_I \alpha_k^{i,j-1}, \varepsilon_I \alpha_k^i + (1-\varepsilon_I)\alpha_k^{i,j-1}\right].$$

Let us show that Assumptions 3.7 (ii), (iii-a), and (iii-c) are satisfied if these rules are used. Observe that, for $i \geq 1$ and $j \geq 1$,

$$\alpha_k^i < \alpha_k^{i,j} \leq \alpha_k^i + (1-\varepsilon_I)^{j-1}(\alpha_k^{i,1} - \alpha_k^i). \tag{3.30}$$

Therefore, Assumption 3.7 (ii) is verified. On the other hand, suppose that $\{\alpha_k^i\}_{i \geq 1}$ converges, and choose an index $j' \geq 1$. Then, one can find an index $i' \geq 1$ such that

$$\alpha_k^{i+1} - \alpha_k^i \leq \varepsilon_E \, \varepsilon_I^{j'-1} \quad \forall i \geq i'.$$

As

$$\alpha_k^{i+1} = \alpha_k^{i,j_i} \geq (1-\varepsilon_I^{j_i-1})\alpha_k^i + \varepsilon_I^{j_i-1}\alpha_k^{i,1} \geq \alpha_k^i + \varepsilon_E \, \varepsilon_I^{j_i-1},$$

we have from the previous inequality that

$$\varepsilon_E \, \varepsilon_I^{j_i-1} \leq \varepsilon_E \, \varepsilon_I^{j'-1} \quad \forall i \geq i'.$$

Now, because $\varepsilon_I < 1$, we obtain $j_i \geq j'$ for all $i \geq i'$, which is Assumption 3.7 (iii-a). Finally, Assumption 3.7 (iii-c) is also guaranteed by the above rules as this can be seen by taking the limit on i and then on j in (3.30).

We have not discussed the case of Assumption 3.7 (iii-b), but it also can easily be satisfied by taking for $i \geq 0$, for example,

$$\alpha_k^{i,j} = \begin{cases} \alpha_k^i + \varepsilon_E & \text{if } j = 1, \\ \frac{1}{2}(\alpha_k^i + \alpha_k^{i,j-1}) & \text{if } j \geq 2, \end{cases}$$

which is compatible with the safeguard rules given above. More appropriate interpolation rules would use the known values of Θ_{σ_k} and its directional derivatives.

3.5. Finite termination of the search algorithm.

The next proposition gives conditions that ensure the finite termination of the PLS algorithm described in sections 3.2 and 3.4 at a point x_{k+1} satisfying

$$\Theta_{\sigma_k}(x_{k+1}) \leq \Theta_{\sigma_k}(x_k) + \omega_1 \, \nu_k(\alpha_k), \tag{3.31}$$

$$g_{k+1}^\top Z_k t_k \geq \omega_2 \, g_k^\top Z_k t_k, \tag{3.32}$$

where the function ν_k is defined recursively in step 1 of the algorithm (see (3.25)). Recall that the search path p_k is also defined recursively in step 1 of the algorithm.

PROPOSITION 3.8. *Suppose that f and c are differentiable on Ω, c is a submersion on Ω, and the decomposition of $\mathbb{R}^n$ described in section* 1 *is made with maps Z^- and A^- which are bounded on Ω. Let x_k be a point in Ω and B_k be a symmetric positive definite matrix of order $n-m$. Suppose that the penalty factor σ_k in* (1.12) *satisfies* (2.2). *Then, if the PLS algorithm described in sections* 3.2 *and* 3.4, *with Assumptions* 3.7, *$\omega_1 \in (0,1)$, and $\omega_2 > 0$, is started from x_k, one of the following situations occurs:*

(i) *the algorithm terminates after a finite number of stages with a step-size $\alpha_k > 0$, a point $x_{k+1} \in \Omega$, and a function ν_k satisfying conditions* (3.31) *and* (3.32);

(ii) *the algorithm terminates prematurely with a step-size $\alpha_k > 0$, a point $x_{k+1} \in \Omega$, and a function ν_k satisfying* (3.31) *only, because either the update criterion does not hold at x_k^1 or* (3.23) *fails at $x_k^{i+1} = x_{k+1}$;*

(iii) *the algorithm builds a sequence of points $\{x_k^i\}_{i\geq 1}$ in Ω and either $\Theta_{\sigma_k}(x_k^i)$ tends to $-\infty$ or $\{x_k^i\}_{i\geq 1}$ tends to a point on the boundary of Ω.*

Proof. We have already observed in section 3.3 that if the algorithm terminates, then either situation (i) or (ii) occurs.

Suppose now that the algorithm cycles: a sequence $\{x_k^i\}_{i\geq 1}$ is built in Ω. This can only occur when $g_k \neq 0$, since (3.32) is always satisfied when $t_k = 0$ and (3.31) is satisfied at x_k^1. We have to show that one of the events given in (iii) occurs. We proceed by contradiction, supposing that $\{\Theta_{\sigma_k}(x_k^i)\}_{i\geq 1}$ is bounded from below and that $\{x_k^i\}_{i\geq 1}$ does not converge to a point on the boundary of Ω. We recall that conditions (3.18) are satisfied for all $i \geq 1$.

Step 1. Let us prove that the sequences $\{F_k^i\}_{i\geq 1}$, $\{\nu_k^{i-1}(\alpha_k^i)\}_{i\geq 1}$, and $\{\alpha_k^i\}_{i\geq 1}$ converge, say to $\overline{F}_k$, $\overline{N}_k$, and $\overline{\alpha}_k$, respectively.

The first sequence is increasing and the second is decreasing; hence, from (3.29), they will converge if we prove that $\{T_k^i\}$ is bounded. But this is clear since $T_k^i = \Theta_{\sigma_k}(x_k) - \Theta_{\sigma_k}(x_k^i) \geq 0$ and $\{\Theta_{\sigma_k}(x_k^i)\}$ is supposed to be bounded below.

On the other hand, using the definition (3.28) of F_k^{i+1}, (3.26), (2.2), (3.18d), $g_k^\top Z_k t_k \leq 0$, and (3.18c), we obtain

$$\begin{aligned} F_k^{i+1} &\geq -\,\omega_1 \sum_{l=0}^{i} \tau_k^l \,(\alpha_k^{l+1} - \alpha_k^l)\, g(x_k^l)^\top Z_k t_k \\ &\geq -\,\omega_1\,\omega_2 \left(\sum_{l=1}^{i} \tau_k^l \,(\alpha_k^{l+1} - \alpha_k^l) \right) g_k^\top Z_k t_k. \end{aligned}$$

Then, the boundedness of $\{F_k^i\}_{i\geq 1}$, $g_k^\top Z_k t_k < 0$, and $\omega_1\omega_2 > 0$ imply that

$$\sum_{i\geq 0} \tau_k^i \,(\alpha_k^{i+1} - \alpha_k^i) < +\infty. \tag{3.33}$$

As $\{\tau_k^i\}_{i\geq 0}$ is bounded away from 0 by Assumption 3.7 (i), $\{\alpha_k^i\}_{i\geq 1}$ converges.

Step 2. Let us show that the sequence $\{x_k^i\}_{i\geq 1}$ converges to a point $\overline{x}_k \in \Omega$.

By definition of F_k^{i+1}, $g_k^\top Z_k t_k \leq 0$, (3.18c), (2.2), and (3.18d), we obtain

$$F_k^{i+1} \geq \omega_1\, \overline{\sigma} \sum_{l=0}^{i} (\alpha_k^{l+1} - \alpha_k^l)\, \|c(x_k^l)\|.$$

Since $\{F_k^i\}_{i\geq 1}$ is bounded, we have the convergence of the series

$$\sum_{i\geq 0} (\alpha_k^{i+1} - \alpha_k^i)\, \|c(x_k^i)\| < +\infty. \tag{3.34}$$

Now, by definition of x_k^i,

$$x_k^i = x_k + \sum_{l=0}^{i-1} (\alpha_k^{l+1} - \alpha_k^l)\, \left(\tau_k^l Z^-(x_k^l) Z_k t_k - A^-(x_k^l) c(x_k^l)\right).$$

Using the boundedness of $Z^-(\cdot)$ and $A^-(\cdot)$ on Ω, (3.33), and (3.34), we see that the series in the right-hand side is absolutely convergent when $i \to \infty$. Therefore, the series is convergent and x_k^i converges to a limit point $\bar{x}_k$. By our assumptions, $\bar{x}_k$ cannot be a point on the boundary of Ω; hence, $\bar{x}_k \in \Omega$.

This implies the following convergence when $i \to \infty$ (see (3.24)):

$$T_k^i \to \overline{T}_k = \Theta_{\sigma_k}(x_k) - \Theta_{\sigma_k}(\bar{x}_k).$$

Furthermore, since $\{\tau_k^i\}$ and $\{1/\tau_k^i\}$ are bounded by Assumption 3.7 (i), there is some $\bar{\tau}_k > 0$ and a subsequence $\mathcal{I} \in \mathbb{N}$ such that for $i \to \infty$, $i \in \mathcal{I}$, we have $\tau_k^i \to \bar{\tau}_k$ and (using (3.19), (3.26), and (2.3))

$$\begin{gathered} d_k^i \to \bar{d}_k = \bar{\tau}_k Z^-(\bar{x}_k) Z_k t_k - A^-(\bar{x}_k) c(\bar{x}_k), \\ \Theta'_{\sigma_k}(x_k^i; d_k^i) \to \bar{\tau}_k\, g(\bar{x}_k)^\top Z_k t_k + \lambda(\bar{x}_k)^\top c(\bar{x}_k) - \sigma_k \|c(\bar{x}_k)\| = \Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k). \end{gathered}$$

Step 3. Let us conclude with the expected contradiction.

Define

$$x_k^{i,j} = x_k^i + (\alpha_k^{i,j} - \alpha_k^i)\, d_k^i.$$

By Assumption 3.7 (iii-b), the sequence $\{\alpha_k^{i,j}\}_{i \geq 1}$ converges to a step-size $\bar{\alpha}_k^{\infty,j} \neq \bar{\alpha}_k$. Therefore, for any $j \geq 1$,

$$x_k^{i,j} \to \bar{x}_k^{\infty,j} = \bar{x}_k + (\bar{\alpha}_k^{\infty,j} - \bar{\alpha}_k)\, \bar{d}_k \quad \text{when } i \to \infty \text{ with } i \in \mathcal{I}.$$

Now, for fixed $j \geq 1$, Assumption 3.7 (iii-a) says that $j_i > j$ for sufficiently large i. This means that, for large i, $x_k^{i,j}$ is not accepted in step 2 of the PLS algorithm. Hence, either $x_k^{i,j} \notin \Omega$ or (3.21) is not verified with $\alpha = \alpha_k^{i,j}$. This can be written

$$\begin{aligned} x_k^{i,j} \in \Omega \quad \Longrightarrow \quad \Theta_{\sigma_k}(x_k^{i,j}) &> \Theta_{\sigma_k}(x_k) + \omega_1\, \nu_k^i(\alpha_k^{i,j}) \\ &= \Theta_{\sigma_k}(x_k) + \omega_1\, \nu_k^i(\alpha_k^{i+1}) + \omega_1\, (\alpha_k^{i,j} - \alpha_k^{i+1})\, \Theta'_{\sigma_k}(x_k^i; d_k^i). \end{aligned}$$

Taking the limit on $i \in \mathcal{I}$ in this relation and using $\omega_1 \overline{N}_k \geq -\overline{T}_k$ from (3.29) and the results of step 2, we obtain

$$\begin{aligned} \bar{x}_k^{\infty,j} \in \Omega \quad \Longrightarrow \quad \Theta_{\sigma_k}(\bar{x}_k^{\infty,j}) &\geq \Theta_{\sigma_k}(x_k) - \overline{T}_k + \omega_1\, (\bar{\alpha}_k^{\infty,j} - \bar{\alpha}_k)\, \Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k) \\ &= \Theta_{\sigma_k}(\bar{x}_k) + \omega_1\, (\bar{\alpha}_k^{\infty,j} - \bar{\alpha}_k)\, \Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k). \end{aligned}$$

Hence,

$$\bar{x}_k^{\infty,j} \in \Omega \quad \Longrightarrow \quad \frac{\Theta_{\sigma_k}(\bar{x}_k^{\infty,j}) - \Theta_{\sigma_k}(\bar{x}_k)}{\bar{\alpha}_k^{\infty,j} - \bar{\alpha}_k} \geq \omega_1\, \Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k).$$

Because $\bar{x}_k \in \Omega$, taking the limit in this implication when j tends to infinity gives (with Assumption 3.7 (iii-c)) $\Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k) \geq \omega_1\, \Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k)$. Because $\omega_1 < 1$, we get

$$\Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k) \geq 0.$$

On the other hand,

$$\begin{aligned} \Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k) &= \bar{\tau}_k\, g(\bar{x}_k)^\top Z_k t_k + \lambda(\bar{x}_k)^\top c(\bar{x}_k) - \sigma_k \|c(\bar{x}_k)\| \\ &\leq \bar{\tau}_k\, \omega_2\, g_k^\top Z_k t_k \\ &< 0, \end{aligned}$$

because $\|\lambda(\bar{x}_k)\|_D \leq \sigma_k$ from the limit in (3.18d), $g(\bar{x}_k)^\top Z_k t_k \leq \omega_2\, g_k^\top Z_k t_k$ from the limit in (3.18c), $\bar{\tau}_k \omega_2 > 0$, and $g_k \neq 0$. This inequality contradicts the nonnegativity of $\Theta'_{\sigma_k}(\bar{x}_k; \bar{d}_k)$ obtained above and concludes the proof. □

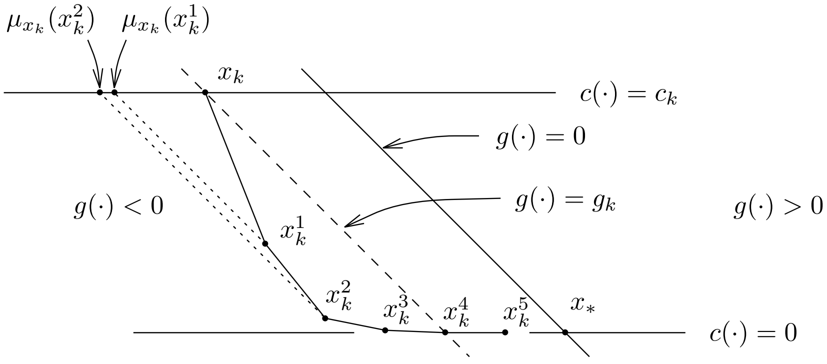


FIG. 3.4. *A difficult case for the PLS.*

3.6. Resetting the PLS. In some cases, the PLS described in sections 3.2 and 3.4 can be trapped in a situation where its behavior is poor. Such a situation may happen when conditions (3.11) do not hold, in particular, when the path $\mu_{x_k} \circ p_k$ is not a descent path for f. Then, the search algorithm may necessitate a large number of inner iterations to satisfy the reduced Wolfe conditions, and the vectors γ_k and δ_k may be erroneous. We show how to improve the PLS algorithm in this situation.

Here is an example of such a situation, in which $n = 2$ and $m = 1$. Take

$$f(x) = \frac{1}{4}\left(x_{(1)} + x_{(2)}\right)^2 \qquad \text{and} \qquad c(x) = e^{x_{(2)}} - 1,$$

where $x_{(i)}$ denotes the ith component of x. The unique solution of this problem is clearly $x_* = 0$. With the following decomposition of $\mathbb{R}^2$,

$$Z^-(x) = e_1 \qquad \text{and} \qquad A^-(x) = e^{-x_{(2)}} e_2,$$

where (e_1, e_2) is the canonical basis of $\mathbb{R}^2$, the reduced gradient is given by $g(x) = (x_{(1)} + x_{(2)})/2$, and the transversal component of the steps are orthogonal to the constraint manifold. The manifolds $c(\cdot) = 0$ and $g(\cdot) = 0$ are the lines $x_{(2)} = 0$ and $x_{(1)} + x_{(2)} = 0$ represented in Figure 3.4.

Now, suppose that the current iterate x_k has coordinates $(-1-\epsilon, 1)$, with $\epsilon > 0$, and that $B_k = I$. Consider an implementation of the PLS in which $\omega_1 = 10^{-4}$, $\omega_2 = 0.9$, $\rho_k^i = 1$, $\tau_k^i = 1$, and the first step-size candidate is $\alpha_k^{i,1} = \alpha_k^i + 1$. If the penalty parameter $\sigma_k = 10$, the step-size $\alpha_k^{i,1}$ is always accepted by the Armijo condition (3.21). When $\epsilon = 0.5$ the search algorithm requires 5 inner iterations. The intermediate points $\{x_k^i\}_{i=1}^5$ are represented in Figure 3.4. By decreasing $\epsilon > 0$, one can obtain as many inner iterations as desired. For example, 21 inner iterations are necessary for $\epsilon = 0.1$, while 2001 are necessary for $\epsilon = 10^{-3}$! The reason is that when ϵ decreases, x_k is closer to the manifold $g(\cdot) = 0$ and the reduced tangent direction $Z_k t_k$ is smaller. Because the iterates go rapidly close to the constraint manifold where the reduced gradient is much more negative than at x_k and because the reduced gradients evaluated at the intermediate points are not used for defining the search directions, the algorithm needs more and more inner iterations to cross the manifold $g(\cdot) = g_k$ (represented by a dashed line in Figure 3.4), beyond which it has to go to satisfy the curvature condition (3.22). If the search path is mapped by μ_{x_k} (see Definition 3.3) on the line $c(\cdot) = c_k$, it is clear from Figure 3.4 that the mapped path starts in the wrong left direction. The correct direction is followed from $\mu_{x_k}(x_k^2)$ only. The basic reason for this behavior is, once again, that reduced methods have no information on the space tangent to the manifold of constant reduced gradient.

In reduced quasi-Newton methods, the update criterion is often a rule that suggests not updating the matrix when the tangential component of the direction is small with respect to its transversal component (see Nocedal and Overton [31] or Gilbert [16]). In the example above, this would result in skipping the update when ϵ is small. By step 3 of the PLS algorithm, the search would be interrupted at x_k^1, avoiding the large number of inner iterations that we have observed. Unfortunately, the implementation of update criteria is often less efficient than expected. Therefore, we propose a modification of the PLS algorithm, such that the situation of the example above is faced with more success. In the modified version, the curvature condition (3.32) is replaced by

$$g_{k+1}^\top Z_k t_k \geq \omega_2 \min_{0\leq i<i_k} g(x_k^i)^\top Z_k t_k. \tag{3.35}$$

The PLS algorithm with this new condition is said to be "with resetting" and it is denoted by "PLS-rst" below. Since inequality (3.35) is less restrictive than (3.32), it is clear that PLS-rst terminates more quickly than PLS. In particular, it still has the finite termination property of Proposition 3.8. Questions concerning the global convergence of the algorithm with PLS and PLS-rst are discussed in the next section. On the example above, this new version of the algorithm terminates in 3, 5, and 204 inner iterations when $\epsilon = 0.5$, 0.1, and 10^{-3}, respectively.

When the PLS is reset at an intermediate point $x_k^{l_k}$, where l_k gives the current arg-minimum in (3.35), the reduced direction $Z_k t_k$ may be very small (this is the case in the example above), so that guessing the correct tangent step-size (or the tangent scaling factor τ_k^i) by using an extrapolation formula may be useful. For example, one can try to use $g(x_k^{i-1})^\top Z_k t_k$ and $g(x_k^i)^\top Z_k t_k$ to evaluate τ_k^{i+1}. The rationale behind this is that, when (3.7) and (3.11) hold, $g(x_k^{i-1})^\top Z_k t_k$ and $g(x_k^i)^\top Z_k t_k$ are derivatives of the function $\alpha \mapsto (f\circ\psi_k)(\alpha Z_k t_k)$. Hence, when $g(x_k^{i-1})^\top Z_k t_k < g(x_k^i)^\top Z_k t_k$, one can use quadratic interpolation to determine τ_k^{i+1}. Using this, the runs with $\epsilon = 0.5$, 0.1, and 10^{-3} terminate now in 3, 4, and 5 inner iterations, respectively.

With PLS-rst, the vectors γ_k and δ_k used to update B_k have to be modified. If l_k denotes the largest index for which the minimum in (3.35) is reached, then it is appropriate to take

$$\gamma_k = g_{k+1} - g(x_k^{l_k}) \qquad \text{and} \qquad \delta_k = \left(\sum_{i=l_k}^{i_k-1} \tau_k^i(\alpha_k^{i+1}-\alpha_k^i)\right) Z_k t_k.$$

Note again that when $\tau_k^i = 1$ for all i, $\delta_k = (\alpha_k - \alpha_k^{l_k})Z_k t_k$, simply. With this choice, $\gamma_k^\top \delta_k > 0$. Note also that these vectors usually put better information into the matrix B_{k+1}, because the new value of δ_k is generally closer to the reduced step from $x_k^{l_k}$ to x_{k+1} than the previous value of δ_k is close to the reduced step from x_k to x_{k+1}. This remark particularly applies to the example above. From Figure 3.4, we have $\gamma_k^{\text{PLS}} = g(x_k^5) - g_k \simeq g(x_k^5) - g(x_k^4) \simeq g(x_k^3) - g(x_k^2) = \gamma_k^{\text{PLS-rst}}$. But $\delta_k^{\text{PLS}} = 5\delta_k^{\text{PLS-rst}}$ and it is clear that $\delta_k^{\text{PLS-rst}}$ corresponds better to $\gamma_k^{\text{PLS}} \simeq \gamma_k^{\text{PLS-rst}}$ than δ_k^{PLS}.

4. Convergence result. In this section, we show that the PLS method of section 3 is able to force convergence of reduced secant algorithms from remote starting points. For this, we shall suppose that the calculation of the reduced matrices keeps the sequences $\{B_k\}$ and $\{B_k^{-1}\}$ bounded. This is a rather strong assumption, but the present state of the convergence theory for constrained problems is not sufficiently developed to have significantly better results. For instance, Byrd and Nocedal [4] analyze the global convergence of reduced quasi-Newton algorithms under conditions that are not known to be guaranteed by the present step-size determination method.

The algorithm we consider is therefore not fully determined, since we shall not be very specific on the way the matrices are updated (in particular, the update criterion will remain unspecified). A possibility is to use the BFGS update formula (1.8), which is always well defined when the PLS succeeds. There is still another facet of the algorithm that must be clarified—this is how the penalty parameter σ_k is updated. We suppose that a rule is chosen such that the following three properties are satisfied ($\overline{\sigma} > 0$ is a constant):

$$(4.1)\qquad \begin{cases} \sigma_k \geq \|\lambda_k\|_D + \overline{\sigma} \quad \forall k \geq 1, \\ \exists \text{ an index } k_1, \ \ \forall k \geq k_1, \ \ \sigma_{k-1} \geq \|\lambda_k\|_D + \overline{\sigma} \quad \Longrightarrow \quad \sigma_k = \sigma_{k-1}, \\ \{\sigma_k\} \text{ is bounded} \quad \Longrightarrow \quad \sigma_k \text{ is updated finitely often.} \end{cases}$$

Many rules can satisfy these conditions. For example, Mayne and Polak [27] suggest taking ($\tilde{\sigma} > 1$):

$$(4.2)\qquad \text{if } \sigma_{k-1} \geq \|\lambda_k\|_D + \overline{\sigma}, \text{ then } \sigma_k = \sigma_{k-1}, \text{ else } \sigma_k = \max(\tilde{\sigma}\sigma_{k-1}, \|\lambda_k\|_D + \overline{\sigma}).$$

We can now outline the algorithm, whose convergence is analyzed in Proposition 4.2. At the beginning, the iteration index k is set to 1 and the constants ω_1 and ω_2 used in the PLS algorithm are chosen in $(0, 1)$. When the kth iteration starts, an iterate $x_k \in \Omega$ is known, as well as a positive definite matrix B_k. Then the PLS technique is used to determine the next iterate x_{k+1} such that $x_{k+1} \in \Omega$ and (3.31) (and possibly (3.32)) hold. Then, the matrix B_k is updated, provided that the PLS algorithm has not been interrupted prematurely by an update criterion or the failure of (3.23). Finally, the penalty parameter is updated according to the rules (4.1).

In unconstrained optimization, the curvature condition corresponding to (3.32) prevents the step-size from being too small, which is important for the global convergence of the algorithm. In constrained problems, this is not necessarily the case, because condition (3.32) ignores the transversal component of the search path. For example, when the objective function f is constant the reduced gradient vanishes and (3.32) is satisfied for any step-sizes, independently of the form of the search path. Therefore, something has to be done such that the first step-size candidate α_k^1 ($\leq \alpha_k$) will not be too small. For the same reason, the first tangent scaling factor τ_k^0 must be chosen bounded away from zero. We gather below additional conditions that the tuning of the PLS algorithm must take into account in order to get global convergence.

Assumptions 4.1. We suppose that the determination of the tangent scaling factors $\tau_k^0 > 0$ and the step-sizes α_k^1 is such that

(i) the sequences $\{\tau_k^0\}_{k\geq 1}$ and $\{1/\tau_k^0\}_{k\geq 1}$ are bounded,

(ii) the sequence $\{\alpha_k^{0,1}\}_{k\geq 1}$ is bounded away from zero,

(iii) there exists a constant $\beta \in (0, 1)$ such that for all $k \geq 1$ and $j \geq 1$, $\alpha_k^{0,j+1} \geq \beta\alpha_k^{0,j}$.

Note that these assumptions are compatible with Assumptions 3.7 and the safeguard rules given afterwards. Assumptions 4.1 (ii) and (iii) can be satisfied, for instance, by using Armijo's backtracking to determine the first step-size candidate α_k^1 from a constant value for $\alpha_k^{0,1}$. In quasi-Newton methods, $\tau_k^0 = 1$ and $\alpha_k^{0,1} = 1$ are recommended.

In Proposition 4.2 below, we suppose that a sequence $\{x_k\}$ is generated in Ω. This implicitly supposes that the PLS algorithm never cycles: situation (i) or (ii) of Proposition 3.8 occurs at each iteration. We denote by $\operatorname{dist}(x, \Omega^c)$ the Euclidean distance between a point x and the complementary set of Ω.

PROPOSITION 4.2. *Suppose that f and c are differentiable on Ω with Lipschitz continuous derivatives, that c is a submersion on Ω, that the map A^- is continuous and bounded on Ω, and that Z^- is bounded on Ω. Suppose also that the algorithm for solving problem* (1.4) *outlined above generates a sequence $\{x_k\}$ in Ω by the PLS method with Assumptions* 3.7 *and* 4.1 *and that the constants ω_1 and ω_2 are taken in* $(0,1)$. *Suppose finally that the symmetric positive definite matrices B_k used in the algorithm are such that $\{B_k\}$ and $\{B_k^{-1}\}$ are bounded. Then, one of the following situations occurs:*
(i) $\{\sigma_k\}_{k\geq 1}$ *is unbounded and* $\{x_k : \sigma_k \neq \sigma_{k-1}\}$ *has no accumulation point in* Ω,
(ii) σ_k *is modified finitely often and one of the following situations occurs:*
(a) $g_k \to 0$ *and* $c_k \to 0$,
(b) $\Theta_{\sigma_k}(x_k) \to -\infty$,
(c) $\operatorname{dist}(x_k, \Omega^c) \to 0$ *for some subsequence of indices* $k \to \infty$.

Proof. First, consider situation (i): $\{\sigma_k\}$ is unbounded. Let $\mathcal{K}$ be the subsequence of indices $\{k : \sigma_k \neq \sigma_{k-1},\ k \geq k_1\}$ (k_1 given by (4.1)). From (4.1),

$$\sigma_{k-1} < \|\lambda_k\|_D + \overline{\sigma} \quad \text{for } k \in \mathcal{K}.$$

As $\{\sigma_k\}_{k\geq k_1}$ is increasing, if it is unbounded, the inequality above shows that the sequence $\{\|\lambda_k\|_D\}_{k\in\mathcal{K}}$ tends to ∞. Then, by continuity of $x \mapsto \lambda(x)$ on Ω, $\{x_k : \sigma_k \neq \sigma_{k-1}\}$ has no accumulation point in Ω.

Suppose now that $\{\sigma_k\}$ is bounded. By (4.1), σ_k is modified finitely often: $\sigma_k = \sigma$ for $k \geq k_2$, say. Suppose also that $\Theta_{\sigma_k}(x_k)$ is bounded from below and that $\{x_k\}$ remains away from Ω^c. We have to prove that situation (ii-a) of the proposition occurs. We denote by C an "absorbing" positive constant independent of k.

From the definition (3.28) of $F_k^{i_k}$, (3.26), the fact that (3.18c) holds for $i = 1, \dots, i_k - 1$, $g_k^\top Z_k t_k \leq 0$, (4.1), (3.18d), and the boundedness of $\{B_k\}$, we have the following for $k \geq k_2$:

$$\begin{aligned}
F_k^{i_k} &= -\,\omega_1 \sum_{i=0}^{i_k-1} (\alpha_k^{i+1} - \alpha_k^i)\Theta'_\sigma(x_k^i; d_k^i) \\
&\geq \omega_1 \sum_{i=0}^{i_k-1} (\alpha_k^{i+1} - \alpha_k^i)\ \big(\omega_2\, \tau_k^i\, g_k^\top B_k^{-1} g_k + \sigma\|c(x_k^i)\|\big) \\
&\geq C \left(\sum_{i=0}^{i_k-1} \tau_k^i\, (\alpha_k^{i+1} - \alpha_k^i)\, \|g_k\|^2 + \sum_{i=0}^{i_k-1} (\alpha_k^{i+1} - \alpha_k^i)\, \|c(x_k^i)\| \right).
\end{aligned} \tag{4.3}$$

As the sequence $\{\Theta_\sigma(x_k)\}_{k\geq k_2}$ decreases and is bounded below, it converges. Then, from (3.31) and $\omega_1\, \nu_k(\alpha_k) \leq -F_k^{i_k}$ (use (3.29) with $i = i_k$), we see that $F_k^{i_k} \to 0$. Therefore, the terms in the right-hand side of (4.3) converge to zero when $k \to \infty$:

$$\left\{ \begin{array}{l}
\displaystyle\sum_{i=0}^{i_k-1} \tau_k^i\, (\alpha_k^{i+1} - \alpha_k^i)\, \|g_k\|^2 \to 0, \\
\displaystyle\sum_{i=0}^{i_k-1} (\alpha_k^{i+1} - \alpha_k^i)\, \|c(x_k^i)\| \to 0.
\end{array} \right. \tag{4.4}$$

The result (ii-a) will be proved if we show that the step-size candidates α_k^1 are bounded away from zero. Indeed, from (4.4) and Assumption 4.1 (i), this implies that $g_k \to 0$ and $c_k \to 0$. We proceed by contradiction, supposing that for some subsequence $\mathcal{K}$ of indices $k \geq k_2$ we have

$$\alpha_k^1 \to 0, \quad \text{when } k \to \infty \text{ in } \mathcal{K}. \tag{4.5}$$

By Assumptions 4.1 (ii) and (iii), we can suppose that, for $k \in \mathcal{K}$, $\alpha_k^1 < \alpha_k^{0,1}$ (therefore $\alpha_k^1 = \alpha_k^{0,j_1}$ for some $j_1 \geq 2$) and $\alpha_k^{0,j_1-1} \leq 1$.

Observe first that we can also suppose that, for $k \in \mathcal{K}$, α_k^{0,j_1-1} is not accepted by the search algorithm because the descent condition (3.21) does not hold for $i = 0$ and $\alpha = \alpha_k^{0,j_1-1}$. Indeed, otherwise we would have a subsequence $\mathcal{K}' \subset \mathcal{K}$ such that

$$x_k^{0,j_1-1} \notin \Omega \quad \text{for } k \in \mathcal{K}'. \tag{4.6}$$

Recall that $r_k = -A_k^- c_k$. We have $x_k^{0,j_1-1} - x_k = \alpha_k^{0,j_1-1}(\tau_k^0 t_k + r_k)$ and, by Assumption 4.1 (iii), $\alpha_k^{0,j_1-1} \leq \alpha_k^1/\beta \leq \alpha_k/\beta$. Then, using the boundedness of $\{\alpha_k^1\}_{k\in\mathcal{K}}$ (due to (4.5)); Assumption 4.1 (i); the boundedness of $\{Z_k^-\}$, $\{B_k^{-1}\}$, and $\{A_k^-\}$; and (4.4); we have for $k \to \infty$ in $\mathcal{K}$

$$\begin{aligned} \|\alpha_k^{0,j_1-1}\, \tau_k^0\, t_k\|^2 &\leq C\, \alpha_k^1\, \tau_k^0\, \|g_k\|^2 \to 0, \\ \|\alpha_k^{0,j_1-1}\, r_k\| &\leq C\, \alpha_k^1\, \|c_k\| \to 0. \end{aligned}$$

Therefore, $(x_k^{0,j_1-1} - x_k) \to 0$ for $k \to \infty$ in $\mathcal{K}$, and (4.6) would imply that $\operatorname{dist}(x_k, \Omega^c)$ tends to 0 for $k \to \infty$ in $\mathcal{K}'$, in contradiction with our assumptions.

Therefore, we can suppose that (3.21) is not satisfied for $i = 0$, $\alpha = \alpha_k^{0,j_1-1}$, and $k \in \mathcal{K}$, i.e.,

$$\Theta_\sigma(x_k^{0,j_1-1}) > \Theta_\sigma(x_k) + \omega_1\, \alpha_k^{0,j_1-1} \left(\tau_k^0\, g_k^\top Z_k t_k + \lambda_k^\top c_k - \sigma \|c_k\|\right). \tag{4.7}$$

We obtain a contradiction with (4.5) by showing that this may not occur for too small α_k^{0,j_1-1}. For this, we expand the left-hand side of (4.7) about x_k.

First, using the Lipschitz continuity of f' on Ω,

$$f(x_k + \alpha\,\tau\, t_k + \alpha\, r_k) \leq f_k + \alpha\,\tau\, g_k^\top Z_k t_k + \alpha\, \lambda_k^\top c_k + C\alpha^2 \left(\tau^2\, \|t_k\|^2 + \|r_k\|^2\right).$$

Similarly, using the Lipschitz continuity of c' on Ω, we get the following for $\alpha \leq 1$:

$$\begin{aligned} \|c(x_k + \alpha\,\tau\, t_k + \alpha\, r_k)\| &\leq \|c_k - \alpha\, c_k\| + C\alpha^2 \left(\tau^2\, \|t_k\|^2 + \|r_k\|^2\right) \\ &= \|c_k\| - \alpha\, \|c_k\| + C\alpha^2 \left(\tau^2\, \|t_k\|^2 + \|r_k\|^2\right). \end{aligned}$$

Grouping these estimates, we obtain the following for $\alpha \leq 1$:

$$\begin{aligned} \Theta_\sigma(x_k + \alpha\,\tau\, t_k + \alpha\, r_k) \leq \Theta_\sigma(x_k) &+ \alpha \left(\tau\, g_k^\top Z_k t_k + \lambda_k^\top c_k - \sigma\, \|c_k\|\right) \\ &+ C\alpha^2 \left(\tau^2\, \|t_k\|^2 + \|r_k\|^2\right). \end{aligned}$$

Using this inequality in (4.7) gives (recall that $\alpha_k^{0,j_1-1} \leq 1$ for $k \in \mathcal{K}$)

$$\begin{aligned} (1-\omega_1)\alpha_k^{0,j_1-1} &\left(\tau_k^0\, g_k^\top B_k^{-1} g_k - \lambda_k^\top c_k + \sigma \|c_k\|\right) \\ &< C(\alpha_k^{0,j_1-1})^2 \Big((\tau_k^0)^2 \|g_k\|^2 + \|c_k\|^2\Big) \quad \text{for } k \in \mathcal{K}. \end{aligned}$$

With the boundedness of $\{B_k\}$, $\{\tau_k^0\}$ and $\{1/\tau_k^0\}$ and the inequalities $\omega_1 < 1$ and $\sigma \geq \|\lambda_k\|_D + \overline{\sigma}$, we obtain

$$\alpha_k^{0,j_1-1}\|g_k\|^2 + \alpha_k^{0,j_1-1}\|c_k\| < C(\alpha_k^{0,j_1-1})^2 \left(\|g_k\|^2 + \|c_k\|^2\right) \quad \text{for } k \in \mathcal{K}.$$

By (4.4) and $\alpha_k^{0,j_1-1} \leq \alpha_k^1/\beta$, $\alpha_k^{0,j_1-1}\|c_k\| \to 0$. Hence, the inequality above gives

$$\alpha_k^{0,j_1-1}\|g_k\|^2 + \alpha_k^{0,j_1-1}\|c_k\| < C(\alpha_k^{0,j_1-1})^2\|g_k\|^2 + C\epsilon_k\alpha_k^{0,j_1-1}\|c_k\| \quad \text{for } k \in \mathcal{K},$$

where $\epsilon_k \to 0$ for $k \in \mathcal{K}$. Finally,

$$\alpha_k^{0,j_1-1}\|g_k\|^2 < C(\alpha_k^{0,j_1-1})^2\|g_k\|^2 \quad \text{for large } k \in \mathcal{K}.$$

Clearly, this strict inequality shows that $\{\alpha_k^{0,j_1-1}\}_{k\in\mathcal{K}}$ is bounded away from zero. As $\alpha_k^1 \geq \beta\alpha_k^{0,j_1-1}$, $\{\alpha_k^1\}_{k\in\mathcal{K}}$ cannot converge to zero, contradicting (4.5).

This contradiction concludes the proof. □

When g is Lipschitz continuous on Ω, Assumptions 4.1 are no longer necessary to prove that $g_k \to 0$. This can be shown by a standard argument, using (3.32) and (4.4). But we were not able to prove that $c_k \to 0$ without these assumptions, for the reasons given above the statement of Assumptions 4.1. On the other hand, once Assumptions 4.1 hold, condition (3.32) is no longer useful for the global convergence (it is not used in the proof above). In this case, if the PLS algorithm is replaced by the PLS-rst method described in section 3.6, the conclusion of Proposition 4.2 still holds.

5. Numerical experiment. The behavior of the PLS technique introduced in section 3 and the reduced quasi-Newton algorithm presented in section 4 have been tested on two model problems with a dimension ranging from $n = 2$ to 500 and a single constraint. They consist in minimizing quadratic functions on the unit sphere. Since there is just one constraint, this problem does not favor reduced SQP methods. A full SQP method should be more efficient on this problem.

The numerical experiments have been done in double precision on a SUN SPARC-station 1, with a program written in Fortran-77.

Test problem I. In the first test problem, the function f to minimize and the constraint function c are defined on $\Omega = \{x \in \mathbb{R}^n : x_{(1)} > 0\}$ by

$$f(x) = \frac{1}{2}\sum_{i=1}^{n}(a_{(i)}x_{(i)} - 1)^2, \qquad c(x) = \frac{1}{2}(\|x\|_2^2 - 1).$$

Here $v_{(i)}$ denotes the ith component of a vector v. The constants $a_{(i)}$ are set to $(n+1-i)/n$ for $1 \leq i \leq n$. The problem is more and more difficult to solve as n increases because the order of the updated matrices and the condition number of the reduced Hessian of the Lagrangian increase with n.

The Jacobian matrix of the constraints $A(x) = x^\top$ is surjective if $x \neq 0$. The matrix $Z^-(x)$, whose columns form a basis of the space tangent to the constraint manifold, and the restoration operator $A^-(x)$ are chosen as follows:

$$Z^-(x) = \begin{pmatrix} -x_{(2)} - \cdots - x_{(n)} \\ x_{(1)}I_{n-1} \end{pmatrix}, \qquad A^-(x) = \frac{x}{\|x\|_2^2}, \tag{5.1}$$

where I_{n-1} is the identity matrix of order $n-1$. These matrices are well defined and injective for $x \in \Omega$. The form of $A^-(x)$ shows that the transversal steps are

Table 5.1
Test problems.

n	$x_{*(1)}$	$\kappa_2(B_*)$
2	0.69	1.
5	0.53	6.
10	0.42	9.
20	0.32	14.
50	0.22	26.
100	0.16	46.
200	0.12	84.
500	0.075	192.

orthogonal (for the Euclidean scalar product) to the space tangent to the constraint manifold.

Table 5.1 gives some information on the problems: n is the number of variables (hence, $n-1$ is the dimension of the constraint manifold and the order of the matrix to update), $x_{*(1)}$ is the first component of the solution, and $\kappa_2(B_*)$ is the ℓ_2 condition number of the reduced Hessian of the Lagrangian at the solution (computed by the LAPACK program DSYEV). Note that although the Hessian of the Lagrangian $L(x, \lambda)$ is a diagonal matrix and $Z^-(x)$ is sparse, the reduced Hessian $Z^-(x)^\top L(x,\lambda)Z^-(x)$ is dense: its (i,j) element is $(a_1^2+\lambda)x_{i+1}x_{j+1} + (a_{i+1}^2+\lambda)x_1^2\delta_{ij}$.

To globalize the algorithm, the exact ℓ_1 penalty function (with the ℓ_1-norm in (1.12)) is used with $\sigma_1 = 2\|\lambda_1\|_\infty$ initially. Next, σ_k is updated by the rule (4.2) with $\overline{\sigma} = \sigma_1/100$ and $\tilde{\sigma} = 2$. The initial point x_1 has its ith component set to $(-1)^{i-1}10$, and the algorithm stops at the point x_k when

$$\|c_k\|_2 \leq 10^{-7}\|c_1\|_2 \qquad \text{and} \qquad \|g_k\|_2 \leq 10^{-7}\|g_1\|_2.$$

The update of the matrix B_k^{-1} is done with the inverse BFGS formula when it is appropriate (this depends on the algorithm and is specified below). The first time this occurs, for $k = k_0$ say, the inverse matrix is first initialized to $\gamma_{k_0}^\top\delta_{k_0}/\|\gamma_{k_0}\|^2 I$ before being updated.

The results of our experiments on test problem I are given in Tables 5.2 to 5.6 and summarized in Table 5.7. Here are some common symbols: "n" is the dimension of the problem, "iter" is the number of iterations, "lin" is the number of times the constraints are linearized, "func" is the number of function calls, "skip" is the number of times the matrix update is skipped, and "$\sigma \nearrow$" is the number of increases of the penalty parameter. The meaning of some other symbols is given below.

To serve as a reference, the first runs have been made with Armijo's backtracking along $d_k = t_k + r_k$ and the *skipping rule*: if at the point found by the search algorithm $\gamma_k^\top\delta_k$ is positive, B_k is updated; otherwise, the update is skipped. This algorithm is denoted by AS-skip. The results are given in Table 5.2.

We see that the number of skips is usually small, except for the cases $n = 20$ and $n = 500$.

In the next experiment, Armijo's backtracking is still used as search technique, but a correction is made to δ_k when $\gamma_k^\top\delta_k$ is not sufficiently positive (the so-called Powell's correction; see Powell [34]): $\tilde{\delta}_k = \theta\delta_k + (1-\theta)B_k^{-1}\gamma_k$, where

$$\theta = \begin{cases} 1 & \text{if } \gamma_k^\top\delta_k \geq 0.2\,\gamma_k^\top B_k^{-1}\gamma_k, \\ 0.8\,\frac{\gamma_k^\top B_k^{-1}\gamma_k}{\gamma_k^\top B_k^{-1}\gamma_k - \gamma_k^\top\delta_k} & \text{otherwise.} \end{cases}$$

Table 5.2
AS-skip: Armijo's search and skipping rule (I).

n	iter	func	skip	$\sigma \nearrow$
2	17	23	1	0
5	55	58	1	1
10	88	93	5	1
20	110	116	31	2
50	82	89	3	3
100	83	95	2	3
200	72	90	3	4
500	91	98	11	5

Table 5.3
AS-Powell: Armijo's search and Powell's correction (I).

n	iter	func	P-cor	$\sigma \nearrow$
2	17	23	1	0
5	55	58	2	1
10	86	89	3	1
20	98	110	14	2
50	74	82	6	3
100	95	118	18	3
200	78	89	4	4
500	104	132	11	5

The update of B_k^{-1} is then made with $(\gamma_k, \tilde{\delta}_k)$ instead of (γ_k, δ_k). This algorithm is denoted by AS-Powell. Table 5.3 shows the results: "P-cor" is the number of Powell's corrections, i.e., the number of times $\theta \neq 1$ in the formula of $\tilde{\delta}_k$ above. We see that this algorithm works slightly better than the method with skipping rule for small n ($n \leq 50$) and slightly worse for larger n. We believe that this may not be fortuitous and may come from update pairs (γ_k, δ_k) of bad quality, in particular of the initial one, which is used to scale the matrix. Indeed, for small n, the effect of an initial pair with wrong information is rapidly compensated by updates with good pairs (this is clearly the case when $n = 2$, since then the matrix to update has order 1 and the update formula is memoryless). On the other hand, from our experience [20], if n is large and if the first pair used to scale the matrix is spoiled, it may take many updates to recover from this bad initial scaling.

For two reasons, we introduce an update criterion in the algorithm AS-Powell. First, we want to see whether an update criterion improves the algorithm by selecting good pairs (γ_k, δ_k) and, second, we want to offer a fairer comparison with algorithms using the PLS, which naturally require update criteria. We take the following inequality as update criterion:

$$\|r_k\|_2 \leq \mu \, \|e^1_{k \ominus 2}\|_2 \, \|t_k\|_2. \tag{5.2}$$

An update is desirable when the inequality holds. In this criterion, μ is a positive constant, $k \ominus 2$ is the index of the last but one iteration at which an update occurred before iteration k (see [16] or [18]), and $e^1_k = \alpha^1_k d_k$. The value used for μ is important for the efficiency of the criterion. In order to get a sufficiently good initial pair, we

TABLE 5.4
AS-Powell-UC: Armijo's search, Powell's correction, and update criterion (I).

n	iter	func	skip	P-cor	$\sigma \nearrow$
2	18	27	3	1	0
5	19	26	7	0	1
10	27	35	5	0	1
20	40	51	8	2	2
50	56	72	8	4	3
100	43	50	4	1	3
200	48	63	5	1	3
500	57	79	10	6	4

TABLE 5.5
PLS: Piecewise line-search and update criterion (I).

n	iter	lin	func	skip	$\sigma \nearrow$
2	16	21	30	3	0
5	19	20	26	7	1
10	27	28	35	5	1
20	37	38	46	7	2
50	51	53	63	7	3
100	43	44	50	7	3
200	47	48	61	5	3
500	43	56	66	10	4

take for μ the quotient

$$\mu = 0.1\,\frac{\|r_1\|_2}{\|e_1^1\|_2\,\|t_1\|_2},$$

so the update criterion cannot be satisfied before a few iterations have been done. This forces the algorithm to choose as its initial scaling pair $(\gamma_{k_0}, \delta_{k_0})$ a better pair than (γ_1, δ_1).

The results of algorithm AS-Powell with this update criterion, denoted as AS-Powell-UC, are given in Table 5.4. They are remarkably better than those of algorithm AS-Powell: the number of iterations and function calls has decreased by 49 % and 43 %, respectively. This confirms our feeling on the importance of selecting good pairs (in particular the first one).

The last two experiments use the PLS technique, provided that the update criterion (5.2) holds. Hence, the update is skipped when the PLS is interrupted by the update criterion or by the test on the penalty parameter (step 5 of the search algorithm). As far as the PLS algorithm is concerned, we have always set $\rho_k^i = 1$ in (3.25), which corresponds to a demanding search. The results with $\rho_k^i = 0$ hardly differ, essentially because the unit step-size is usually accepted by the PLS. The first tangent scaling factor τ_k^0 and the step-size candidates $\alpha_k^{i,1}$ are always set to 1 and $\alpha_k^i + 1$, respectively. Safeguarded quadratic interpolation is used to determine the intermediate step-sizes $\{\alpha_k^{i,j}\}_{j=2}^{j_i}$.

In the first experiment, whose results are given in Table 5.5, the plain PLS method described in sections 3.2 and 3.4 is used with τ_k^i always set to 1 (without tangential extrapolation). A first observation is that the PLS algorithm never cycles, as this is suggested by the theory (Proposition 3.8). Now, comparing the number of linearizations with those of the algorithms with Powell's correction, we observe an important

Table 5.6
PLS-rst: Piecewise line-search (with resetting) and update criterion (I).

n	iter	lin	func	rst	skip	$\sigma \nearrow$
2	17	19	27	1	3	0
5	19	20	26	0	7	1
10	27	28	35	0	5	1
20	37	38	46	0	7	2
50	51	53	65	1	8	3
100	43	44	50	0	7	3
200	47	48	61	0	5	3
500	48	52	65	1	10	4

Table 5.7
Compared performance of the algorithms (I).

Algorithm	iter	lin	func
AS-skip	598	606	662
AS-Powell	607	615	701
AS-Powell-UC	308	316	403
PLS	283	308	377
PLS-rst	289	302	375

improvement with respect to algorithm AS-Powell and a small one with respect to algorithm AS-Powell-UC. The results look quite satisfactory, particularly if we observe that the small improvement with respect to algorithm AS-Powell-UC is due to a very limited use of the PLS technique. Only the cases $n = 2$, $n = 50$, and $n = 500$ use this technique, as this can be seen by a positive number of inner iterations: "lin" − "iter" − 1 > 0. Now the results with $n = 500$ are not very good, since the PLS algorithm requires a great number of inner iterations. By looking more closely at these results, however, we have observed that the deterioration is due to a single iteration and that a phenomenon resembling the one described in the example of section 3.6 occurs.

The last experiment is done with the PLS-rst algorithm of section 3.6. The PLS algorithm is interrupted as soon as condition (3.35) holds. Furthermore, the tangent scaling factor τ_k^i may be different from 1: either τ_k^i is determined by a safeguarded quadratic or cubic extrapolation formula using the values $g(x_k^i)^\top Z_k t_k$ or (when this is unsuccessful, due to the inconsistency of the interpolating values) τ_k^i is doubled at each inner iteration (provided that the descent test (3.18b) has always been verified with $\alpha_k^i = \alpha_k^{i-1,1}$ during the current PLS). The results are given in Table 5.6. The number of iterations with "resettings" are given in a column labeled by 'rst': it is the number of times condition (3.35) differs from (3.32). Of course, only the results of the cases $n = 2$, $n = 50$, and $n = 500$ may change. We see that the very few "resettings" slightly improve the results. We also observe that the number of inner iterations used by the PLS-rst algorithm (= "lin" − "iter" − 1) is now very small.

To summarize, we add up the number of iterations, linearizations, and function calls used by the considered algorithms for all the runs: see Table 5.7 (for every run with AS-skip or AS-Powell, "lin" = "iter" + 1). The results of PLS-rst compare favorably with those of the other techniques.

Test problem II. The second test problem is obtained by changing the objective function in the first test problem. It is now the quadratic form $\frac{1}{2}x^\top Qx - q^\top x$, where

TABLE 5.8
Compared performance of the algorithms (II).

Algorithm	iter	lin	func
AS-skip	1109*	1117*	1294*
AS-Powell	495	503	583
AS-Powell-UC	381	389	490
PLS	375	462	533
PLS-rst	354	382	440

TABLE 5.9
Compared performance for Test problem I *with tangent basis* (5.3).

Algorithm	iter	lin	func	saving in “lin”
AS-skip	453	461	503	24 %
AS-Powell	388	396	440	36 %
AS-Powell-UC	279	287	370	9 %
PLS	254	262	324	15 %
PLS-rst	254	262	324	13 %

TABLE 5.10
Compared performance for Test problem II *with tangent basis* (5.3).

Algorithm	iter	lin	func	saving in “lin”
AS-skip	235	243	298	78 %*
AS-Powell	230	238	303	53 %
AS-Powell-UC	252	260	357	33 %
PLS	226	242	300	48 %
PLS-rst	228	238	295	38 %

$Q_{i,j} = (i+j-1)^{-1}$ and $q_i = n$ for all $i, j \in \{1, \ldots, n\}$. Table 5.8 summarizes the results. The “*” in this table indicates that algorithm AS-skip failed to satisfy the stopping test for $n = 500$. This is due to the fact that the matrix update is very often skipped in this run. The same type of comments as for Test problem I can be given:

- Algorithm AS-Powell works much better than AS-skip;
- The update criterion in AS-Powell-UC improves algorithm AS-Powell significantly;
- The less satisfactory results of the plain PLS algorithm are due to a large number of inner iterations in the PLS (no resettings, see section 3.6);
- The PLS-rst algorithm has the best results, with very few inner iterations.

Change of tangent basis. We would like to mention the results obtained by changing the field of tangent basis Z^-. We now take

$$Z^-(x) = \begin{pmatrix} -x_{(2)}/x_{(1)} - \cdots - x_{(n)}/x_{(1)} \\ I_{n-1} \end{pmatrix}. \tag{5.3}$$

Hence, the elements of the previous matrix Z^- in (5.1) have been divided by $x_{(1)}$. This is motivated by the fact that this new basis satisfies property (3.7) for some parametrization ψ_k, while the basis (5.1) does not (see [19]). All the other parameters of the algorithms have been kept unchanged.

Tables 5.9 and 5.10 give the results corresponding to Test problems I and II. We observe an important improvement in the number of linearizations (last column, i.e., saving in “lin”). Note that this is not due to a change in the conditioning of the problem: since the previous basis has just been divided by $x_{(1)}$, the condition number of the reduced Hessian of the Lagrangian at the solution has not changed. We still do not know whether the fact that the basis (5.3) satisfies (3.7) is a key to explain the improvement (see [19] for a possible explanation).

Comments on the numerical experiments. Among the techniques used to maintain the positive definiteness of the reduced matrices that have been tested (skipping rule, Powell's corrections, PLS technique), the PLS technique appears to be the best one, provided that the method is carefully implemented (PLS-rst version of the algorithm). For the test-problems we considered, two other tools are of great importance for the efficiency of reduced SQP methods: the use of update criteria and a proper choice of the tangent basis field Z^-. It is clear that this small amount of tests impedes from giving final conclusions. More experiments with more realistic problems are necessary before asserting the usefulness of the PLS technique. We have found, however, that these results are encouraging and we believe that this limited number of tests demonstrates the feasibility of the PLS approach.

6. Conclusions.

This paper proposes a method for maintaining the positive definiteness of the matrices in reduced quasi-Newton algorithms for equality constrained optimization. By using a PLS (as opposed to a traditional line-search) technique, which conducts the search of the next iterate along a piecewise linear path, some reduced Wolfe conditions are satisfied whenever desired. One of these conditions is such that between two successive iterates, the function to minimize, reduced to the current manifold, seems to have positive curvature. This allows the algorithm to sustain the positive definiteness of the reduced Hessian approximations from one iteration to the other.

A few numerical experiments have shown that a careful implementation of the technique can do better than other methods, such as the skipping rule or Powell's correction of the BFGS update. This improvement is obtained with reduced methods, despite of their important defect, which is that they have no means to improve the orientation of the transversal component of the step. Because this defect is crucial in the present context (due to the first condition in (3.11)) and because it is not shared with the SQP method, it is expected that the PLS technique could be more clearly efficient when the updated matrices approximate the full Hessian of the augmented Lagrangian. This discussion also raises the question whether an update criterion based on (3.11) rather than on the comparison of the transversal and tangential components of the step can be conceived.

Another feature of the PLS technique is to offer the possibility to have cleaner algorithms. At least, this is an advantage for their analysis. For example, in [18], a strong superlinear convergence result has been proven for an algorithm with PLSs and the update criterion (5.2). It is shown, indeed, that if in the Coleman and Conn reduced algorithm the points $\{x_k^i\}_{k,i}$ converge to a solution satisfying sufficient second order conditions of optimality, then no intermediate point exists eventually ($i_k = 1$ for k large) and the sequence converges q-superlinearly. In this result, the matrices are supposed to be generated by the BFGS formula from any positive definite starting matrix. No other assumptions on the generated matrices are necessary. To our knowledge, this is the first extension of Powell's result (see [32]) to constrained problems.

REFERENCES

[1] P. Armand and J. Ch. Gilbert (1995), *A Piecewise Line-Search Technique for Maintaining the Positive Definiteness of the Updated Matrices in the SQP method*, Rapport de Recherche 2615, INRIA, BP 105, 78153 Le Chesnay, France. Http server: http://www.inria.fr/RRRT/RR-2615.html; ftp server: ftp://ftp.inria.fr/INRIA /publication/RR, file RR-2615.ps.gz (submitted to *Comput. Optim. Appl.*).

[2] L. Armijo (1966), *Minimization of functions having Lipschitz continuous first partial derivatives*, Pacific J. Math., 16, pp. 1–3.

[3] W. M. Boothby (1986), *An Introduction to Differentiable Manifolds and Riemannian Geometry*, 2nd ed., Academic Press, Boston, MA.

[4] R. H. Byrd and J. Nocedal (1991), *An analysis of reduced Hessian methods for constrained optimization*, Math. Programming, 49, pp. 285–323.

[5] R. H. Byrd, R. A. Tapia, and Y. Zhang (1992), *An SQP augmented Lagrangian BFGS algorithm for constrained optimization*, SIAM J. Optim., 2, pp. 210–241.

[6] T. F. Coleman and A. R. Conn (1982), *Nonlinear programming via an exact penalty function: Asymptotic analysis*, Math. Programming, 24, pp. 123–136.

[7] T. F. Coleman and A. R. Conn (1984), *On the local convergence of a quasi-Newton method for the nonlinear programming problem*, SIAM J. Numer. Anal., 21, pp. 755–769.

[8] T. F. Coleman and P. A. Fenyes (1992), *Partitioned quasi-Newton methods for nonlinear constrained optimization*, Math. Programming, 53, pp. 17–44.

[9] L. Conlon (1993), *Differentiable Manifolds – A First Course,* Birkhäuser Boston, Cambridge, MA.

[10] B. N. Pshenichnyi and Yu. M. Danilin (1978), *Numerical Methods for Extremal Problems,* MIR, Moscow.

[11] J. E. Dennis and R. B. Schnabel (1983), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, NJ.

[12] P. Fenyes (1987), *Partitioned Quasi-Newton Methods for Nonlinear Equality Constrained Optimization*, Ph.D. thesis, Department of Computer Science, Cornell University, Ithaca, NY.

[13] R. Fletcher (1980), *Practical Methods of Optimization. Volume* 1 *: Unconstrained Optimization*, John Wiley & Sons, Chichester, UK.

[14] D. Gabay (1982), *Minimizing a differentiable function over a differential manifold*, J. Optim. Theory Appl., 37, pp. 177–219.

[15] D. Gabay (1982), *Reduced quasi-Newton methods with feasibility improvement for nonlinearly constrained optimization*, Math. Programming Study, 16, pp. 18–44.

[16] J. Ch. Gilbert (1988), *Mise à jour de la métrique dans les méthodes de quasi-Newton réduites en optimisation avec contraintes d'égalité*, Modélisation Math. Anal. Numér., 22, pp. 251–288.

[17] J. Ch. Gilbert (1991), *Maintaining the positive definiteness of the matrices in reduced secant methods for equality constrained optimization*, Math. Programming, 50, pp. 1–28.

[18] J. Ch. Gilbert (1993), *Superlinear Convergence of a Reduced BFGS Method with Piecewise Line-Search and Update Criterion*, Rapport de Recherche 2140, INRIA, BP 105, 78153 Le Chesnay, France. Http server: http://www.inria.fr/RRRT/RR-2140.html; ftp server: ftp: //ftp.inria.fr/INRIA/publication/RR, file RR-2140.ps.gz.

[19] J. Ch. Gilbert (1997), *Piecewise line-search techniques for constrained minimization by quasi-Newton algorithms*, in Advances in Nonlinear Programming, Proc. International Conference on Nonlinear Programming, Beijing, China, Kluwer Academic Publishers, Norwell, MA.

[20] J. Ch. Gilbert and C. Lemaréchal (1989), *Some numerical experiments with variable-storage quasi-Newton algorithms*, Math. Programming, 45, pp. 407–435.

[21] P. E. Gill, W. Murray, and M. H. Wright (1981), *Practical Optimization*, Academic Press, New York.

[22] S.-P. Han (1976), *Superlinearly convergent variable metric algorithms for general nonlinear programming problems*, Math. Programming, 11, pp. 263–282.

[23] S.-P. Han and O. L. Mangasarian (1979), *Exact penalty functions in nonlinear programming*, Math. Programming, 17, pp. 251–269.

[24] J.-B. Hiriart-Urruty and C. Lemaréchal (1993), *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, New York.

[25] C. Lemaréchal (1981), *A view of line-searches*, in Optimization and Optimal Control, A. Auslender, W. Oettli, and J. Stoer, eds., Lecture Notes in Control and Information Science 30, Springer-Verlag, Heidelberg, pp. 59–78.

[26] D. C. Liu and J. Nocedal (1989), *On the limited memory BFGS method for large scale optimization*, Math. Programming, 45, pp. 503–520.

[27] D. Q. Mayne and E. Polak (1982), *A superlinearly convergent algorithm for constrained optimization problems*, Math. Programming Study, 16, pp. 45–61.

[28] J. J. Moré and D. C. Sorensen (1984), *Newton's method*, in Studies in Numerical Analysis, G. H. Golub, ed., The Mathematical Association of America, pp. 29–82.

[29] J. J. Moré and D. J. Thuente (1992), *Line Search Algorithms with Guaranteed Sufficient Decrease*, Preprint MCS-P330-1092, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill.

[30] W. Murray and M. H. Wright (1978), *Projected Lagrangian Methods Based on the Trajectories of Penalty and Barrier Functions*, Technical report SOL-78-23, Department of Operations Research, Stanford University, Stanford, CA.

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

[32] M. J. D. Powell (1976), *Some global convergence properties of a variable metric algorithm for minimization without exact line searches*, Nonlinear Programming, SIAM-AMS Proceedings, R. W. Cottle and C. E. Lemke, eds., American Mathematical Society, Providence, RI.

[33] M. J. D. Powell (1978), *Algorithms for nonlinear constraints that use Lagrangian functions*, Math. Programming, 14, pp. 224–248.

[34] M. J. D. Powell (1978), *A fast algorithm for nonlinearly constrained optimization calculations*, in Numerical Analysis, G. A. Watson, ed., Springer, pp. 144–157.

[35] M. J. D. Powell (1985), *The performance of two subroutines for constrained optimization on some difficult test problems*, in Numerical Optimization 1984, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., SIAM, Philadelphia, PA.

[36] R. T. Rockafellar (1970), *Convex Analysis*, Princeton University Press, Princeton, NJ.

[37] L. Schwartz (1992), *Analyse* II: *Calcul Différentiel et Équations Différentielles.* Hermann, Paris, France.

[38] M. Spivak (1979), *Differentiable Geometry, Volume* I, 2nd ed., Publish or Perish, Inc., Houston, TX.

[39] J. Stoer (1984), *Principles of sequential quadratic programming methods for solving nonlinear programs*, in Proc. NATO ASI on Computational Mathematical Programming, Bad Windsheim, Germany.

[40] R. A. Tapia (1977), *Diagonalized multiplier methods and quasi-Newton methods for constrained optimization*, J. Optim. Theory Appl., 22, pp. 135–194.

[41] R. A. Tapia (1988), *On secant updates for use in general constrained optimization*, Math. Comput., 51, pp. 181–202.

[42] R. B. Wilson (1963), *A Simplicial Algorithm for Concave Programming*, Ph.D. thesis, Graduate School of Business Administration, Harvard University, Boston, MA.

[43] P. Wolfe (1969), *Convergence conditions for ascent methods*, SIAM Rev., 11, pp. 226–235.

[44] P. Wolfe (1971), *Convergence conditions for ascent methods* II: *Some corrections*, SIAM Rev., 13, pp. 185–188.

SIAM J. OPTIM.
Vol. 7, No. 3, pp. 814–836, August 1997

GLOBAL CONTINUATION FOR DISTANCE GEOMETRY PROBLEMS*

JORGE J. MORÉ† AND ZHIJUN WU†

Abstract. Distance geometry problems arise in the determination of protein structure. We consider the case where only a subset of the distances between atoms is given and formulate this distance geometry problem as a global minimization problem with special structure. We show that global smoothing techniques and a continuation approach for global optimization can be used to determine global solutions of this problem reliably and efficiently. The global continuation approach determines a global solution with less computational effort than is required by a standard multistart algorithm. Moreover, the continuation approach usually finds the global solution from any given starting point, while the multistart algorithm tends to fail.

Key words. global optimization, continuation methods, smoothing transform, distance geometry problems, macromolecular modeling

AMS subject classifications. 49M37, 65D32, 68Q22, 68Q25, 92C40, 92E10

PII. S1052623495283024

1. Introduction.

Distance geometry is generally associated with the study of the relationship between geometric constraints on the atoms of a molecule and the structure of the molecule. Structures are usually determined with the aid of distance data between the atoms and other geometric constraints (for example, angle constraints), since this information can be obtained from nuclear magnetic resonance (NMR) data. For surveys and reviews of work in this area, see Crippen and Havel [4], Havel [9], Kuntz, Thomason, and Oshiro [16], and Brünger and Nilges [1].

We consider the case where only distance data is available. Moreover, since NMR only yields estimates for a fraction of the distances, we assume that the distances $\delta_{i,j}$ between the (i,j) pair of atoms are only available for a subset $\mathcal{S}$ of the atom pairs. The problem we study is to find positions $x_1, \ldots, x_m$ in $\mathbb{R}^3$ of the atoms in the molecule such that

$$\|x_i - x_j\| = \delta_{i,j}, \qquad (i,j) \in \mathcal{S}. \tag{1.1}$$

If there is no solution $x_1, \ldots, x_m$ to these constraints, then the length specification must be in error. This can happen, for example, if the triangle inequality

$$\delta_{i,j} \le \delta_{i,k} + \delta_{k,j}$$

is violated for atoms $\{i, j, k\}$ with bond length constraints.

Since the data obtained from NMR is inaccurate, distance geometry problems that arise in the determination of protein structure are usually associated with the

*Received by the editors March 18, 1995; accepted for publication (in revised form) February 7, 1996. 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. This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U. S. Department of Energy, under contract W-31-109-Eng-38.

http://www.siam.org/journals/siopt/7-3/28302.html

†Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439 (more@mcs.anl.gov, zwu@mcs.anl.gov).

more general problem of finding positions $x_1, \ldots, x_m$ in $\mathbb{R}^3$ such that

$$l_{i,j} \leq \|x_i - x_j\| \leq u_{i,j}, \qquad (i,j) \in \mathcal{S}, \tag{1.2}$$

where $l_{i,j}$ and $u_{i,j}$ are lower and upper bounds on the distance constraints, respectively. We do not consider the general problem (1.2) because the aim of this paper is to show that algorithms based on the continuation approach for global optimization can be used to determine solutions of (1.1) reliably and efficiently. The techniques of this paper can be extended to (1.2), but the theory is not as elegant.

Saxe [21] proved that the distance geometry problem (1.1) in $\mathbb{R}^d$ is NP-hard. The proof of this result, when all the atoms are restricted to $\mathbb{R}^1$, is obtained by reducing the problem to the set partition problem: given positive integers $s_1, \ldots, s_m$, determine a partition of these integers in sets $\mathcal{S}_1$ and $\mathcal{S}_2$ such that

$$\sum_{i \in \mathcal{S}_1} s_i = \sum_{i \in \mathcal{S}_2} s_i.$$

The proof is instructive. Given an instance of the set partition problem, consider a distance geometry problem in $\mathbb{R}^1$ with $m+1$ atoms, where

$$\delta_{i,i+1} = s_i, \quad 1 \leq i \leq m, \quad \delta_{1,m+1} = 0.$$

If the distance geometry problem (1.1) has a solution, then the constraint $\delta_{1,m+1} = 0$ implies that $x_{m+1} = x_1$, and thus

$$\sum_{i=1}^{m} (x_{i+1} - x_i) = x_{m+1} - x_1 = 0.$$

Since $|x_{i+1} - x_i| = s_i$, the sets $S_1 = \{i : x_{i+1} - x_i \geq 0\}$ and $S_2 = \{i : x_{i+1} - x_i < 0\}$ solve the set partition problem.

We formulate the distance geometry problem (1.1) in terms of finding the global minimum of the function

$$f(x) = \sum_{i,j \in \mathcal{S}} w_{i,j} \left(\|x_i - x_j\|^2 - \delta_{i,j}^2 \right)^2, \tag{1.3}$$

where $w_{i,j}$ are positive weights. Clearly, $x \in \mathbb{R}^n$ solves the distance geometry problem if and only if $f(x) = 0$. We could use any global optimization algorithm (see [20, 12, 5] for global optimization background) in the search for a global minimum of f, but these general algorithms do not take advantage of the structure in the distance geometry problem. Other algorithms used in the solution of distance geometry problems (for example, Hendrickson [10, 11], Havel [9], and Glunt, Hayden, and Raydan [7, 8]) must also rely on general techniques, such as multistarts or simulated annealing, to claim convergence to a global minimizer.

The continuation approach for global optimization hinges on the ability to gradually transform the original function into a smoother function with fewer local minimizers. An optimization algorithm is then applied to the transformed function, tracing their minimizers back to the original function. The idea of transforming a function into a smoother function is appealing; the main approaches include the diffusion equation method of Piela, Kostrowicki, and Scheraga [19], the packet annealing method of Shalloway [24, 23], and the effective energy simulated annealing method of Coleman, Shalloway, and Wu [2, 3]. In the diffusion equation method, the transformation can be written (see [13, 14] for details) in the form

$$\frac{1}{(4\pi\tau)^{n/2}} \int_{\mathbb{R}^n} f(y) \exp\left(-\frac{\|y-x\|^2}{4\tau} \right) dy, \tag{1.4}$$

where τ is a parameter (time). The smoothing properties of this transformation have been studied by the researchers in Scheraga's group, often in connection with the search for the lowest energy conformation of a molecule (see, for example, [13, 14, 15, 22]). The transformation used in the packet annealing method and in the effective energy simulated annealing method can be written in the form

$$\frac{1}{\pi^{n/2}|\mathrm{det}\Lambda|^n} \int_{\mathbb{R}^n} \exp\left(-\frac{f(y)}{\kappa_B t} \right) \exp\left(-\|\Lambda^{-1}(y-x)\|^2 \right) dy, \tag{1.5}$$

where κ_B is the Boltzmann constant, t is a parameter (temperature), and Λ is a nonsingular matrix (the sampling scale). Other transformations used in molecular conformation problems are reviewed by Straub [25]. In this paper we follow the work of Wu [26] by developing the general properties and use of (1.4) in continuation algorithms for the solution of large global optimization problems, since this transformation seems to have the strongest mathematical properties.

We feel that (1.4) is likely to play an important role, not only in the molecular conformation problem, but in the solution of a wide variety of global optimization problems. For this reason section 2 introduces the term *Gaussian transform* to denote this transformation. We also illustrate the smoothing properties of the general Gaussian transform on a simple two-dimensional problem. This example also provides motivation for the continuation approach.

Section 3 presents some of the more interesting properties of the Gaussian transform. We study, in particular, the computation of the Gaussian transform for the decomposable functions. This is an important class of functions because many of the functions that arise in applications are decomposable. This class of functions was introduced by Wu [26] under the term *generalized multilinear functions*; we are using the term *decomposable* to avoid confusion with the use of multilinear for a function that is linear in each argument.

Our approach for solving the distance geometry problem is outlined in sections 4 and 5. We compute the Gaussian transform of function (1.3) as a special case of more general results in section 4, while section 5 presents the basic ideas behind global continuation algorithms. We concentrate on an approach based on choosing a predetermined sequence of smoothing parameters, since this approach already brings out the power of the continuation algorithm. In future work we plan to address more sophisticated approaches for choosing the smoothing parameters.

In section 6 we consider a typical distance geometry problem and compare a basic global continuation algorithm with a multistart method for global optimization. We are interested in the solution of problems with a large number of atoms, and thus we performed our numerical testing on the Argonne IBM SP system. This system has 128 nodes, where each node is an IBM RS/6000-370 with 128 MB of memory. One of our main conclusions from the numerical results is that the continuation algorithm usually finds a solution of the distance geometry problem (1.1) from any given starting point. On the other hand, the local minimization algorithm used in the multistart methods is unreliable as a method for determining global solutions. As a result, the multistart method becomes increasingly unreliable and expensive as the number of atoms increases. We also show that the continuation approach determines a global solution with less computational effort than is required by the local minimization algorithm.

2. Continuation for global optimization. In the continuation approach for global optimization, the original function is gradually transformed into a smoother function with fewer local minimizers. An optimization algorithm is then applied to the transformed function, tracing the minimizers back to the original function. In this section we define the transformation and provide motivation for the continuation approach.

The transformed function depends on a parameter λ that controls the degree of smoothing. The original function is obtained if $\lambda = 0$, while smoother functions are obtained as λ increases.

Definition 2.1. *The Gaussian transform* $\langle f \rangle_\lambda$ *of a function* $f : \mathbb{R}^n \mapsto \mathbb{R}$ *is*

$$\langle f \rangle_\lambda(x) = \frac{1}{\pi^{n/2}\lambda^n} \int_{\mathbb{R}^n} f(y) \exp\left(-\frac{\|y-x\|^2}{\lambda^2}\right) dy. \tag{2.1}$$

We are using the term Gaussian transform because we can view $\langle f \rangle_\lambda(x)$ as the expected value of $f(x)$ with respect to the Gaussian density function

$$\rho_\lambda(y) = \frac{1}{\pi^{n/2}\lambda^n} \exp\left(-\frac{\|y\|^2}{\lambda^2}\right). \tag{2.2}$$

The value $\langle f \rangle_\lambda(x)$ of the Gaussian transformation is a weighted average of $f(x)$ in a neighborhood of x, with the relative size of this neighborhood controlled by the parameter λ. The size of the neighborhood decreases as λ decreases so that when $\lambda = 0$, the neighborhood is the center x. The Gaussian transform can also be viewed as the convolution of the function f with the Gaussian density function. We explore additional properties of the Gaussian transformation in the next section.

We illustrate the transformation process with a function that is a linear combination of four Gaussian functions. In this section we discuss the transformation in terms of the global maximization problem because visualization is easier in this case. Note, however, that in the rest of the paper we deal with the global minimization problem.

The function that we use to illustrate the transformation process appears in the top left corner of Figure 2.1. This function is of the general form

$$f(x) = \sum_{i=1}^{4} \alpha_i \exp\left(\frac{\|x-x_i\|^2}{\sigma_i^2}\right), \tag{2.3}$$

where $\sigma_i = 0.5$ for $1 \le i \le 4$, $\alpha_1 = 1.5$, and $\alpha_i = 1$ for $i = 2, 3, 4$; the centers x_i are the vertices of the square $[-0.5, 0.5] \times [-0.5, 0.5]$. As can be seen in Figure 2.1, the function has four maximizers in $[-2, 2] \times [-2, 2]$. The Gaussian transforms of (2.3) for three values of λ also appear in Figure 2.1. The top right corner corresponds to $\lambda = 0.2$, and in the bottom row we have $\lambda = 0.3, 0.4$.

Figure 2.1 clearly shows that the original function is gradually transformed into a smoother function with fewer local maximizers and that the smoothing increases as λ increases. We can view the Gaussian transform of a function as a coarse approximation to the original function, with small and narrow maximizers being removed while the overall structure of the function is maintained. This property allows an optimization

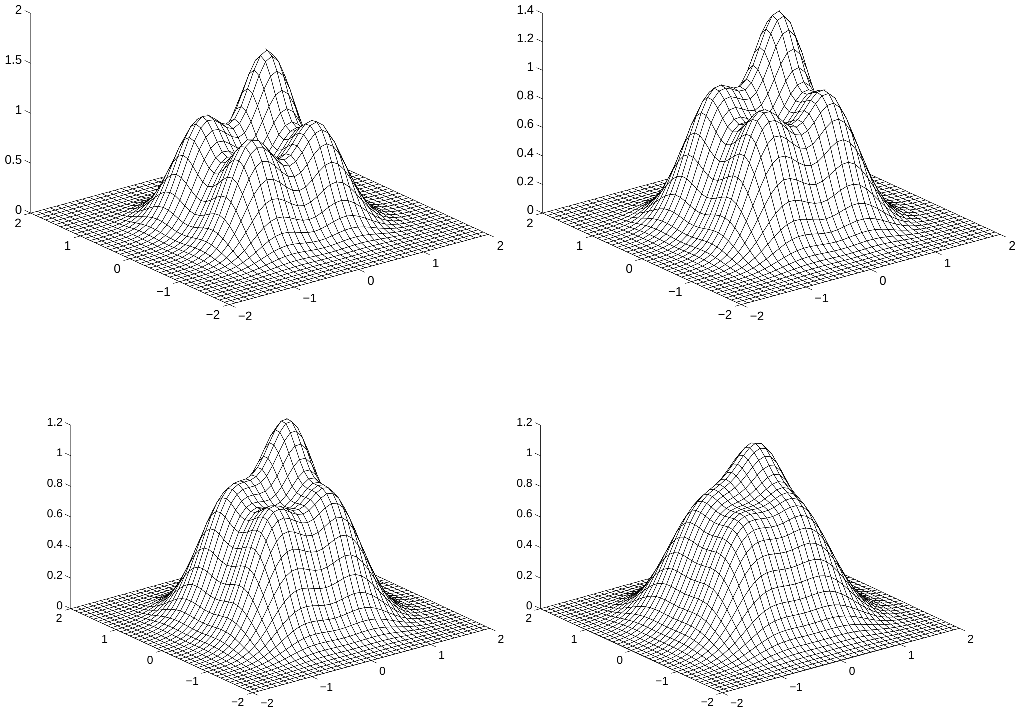


FIG. 2.1. *The Gaussian transform of a function. The original function (*$\lambda = 0$*) is in the top left corner, with* $\lambda = 0.2$ *in the top right corner,* $\lambda = 0.3$ *in the bottom left corner, and* $\lambda = 0.4$ *in the bottom right corner.*

procedure to skip less interesting local maximizers and to concentrate on regions with average high function values, where a global maximizer is most likely to be located.

Another point that is apparent from Figure 2.1 is that a continuation process based on the Gaussian transform will find the global maximizer. In general, we cannot expect that the continuation process will succeed on an arbitrary function. In particular, the Gaussian transform eliminates tall, narrow hills; hence, if the global maximizer lies in one of these hills, the continuation approach is likely to fail. Determining broad regions of maximal value is often of more interest than determining tall, narrow hills, so this characteristic of the continuation approach should be viewed as a strength, rather than a weakness.

3. The Gaussian transform.

We have defined the Gaussian transform for a function $f : \mathbb{R}^n \mapsto \mathbb{R}$ by (2.1). In many cases, it is preferable to make the change of variables $y \mapsto x + \lambda u$ in (2.1) to obtain

$$\langle f \rangle_\lambda(x) = \frac{1}{\pi^{n/2}} \int_{\mathbb{R}^n} f(x + \lambda u) \exp\left(-\|u\|^2\right)\, du. \tag{3.1}$$

In this section we explore some of the properties of this transformation.

The Gaussian transform is defined for a large class of functions. In particular, the transformation is defined if f is continuous almost everywhere and if

$$|f(x)| \le \beta_1 \exp(\beta_2 \|x\|) \tag{3.2}$$

for positive constants β_1 and β_2. These assumptions guarantee that f is bounded on compact sets but allow for unbounded f on $\mathbb{R}^n$. In the development that follows, we assume that f satisfies assumptions (3.2).

An important property of this transformation is that $\langle f\rangle_\lambda$ is a linear operator in the sense that

$$\langle \alpha f\rangle_\lambda = \alpha\langle f\rangle_\lambda, \qquad \langle f_1 + f_2\rangle_\lambda = \langle f_1\rangle_\lambda + \langle f_2\rangle_\lambda$$

for any scalar α and functions f_1 and f_2. Also note that the Gaussian transform of the identity function is unity; this result depends on the identity

$$\int_{-\infty}^{+\infty} \exp\left(-\xi^2\right)\, d\xi = \pi^{1/2}.$$

More generally, if $\mu_1 \le f(x) \le \mu_2$ for all $x \in \mathbb{R}^n$, then $\mu_1 \le \langle f\rangle_\lambda(x) \le \mu_2$ also holds for all $x \in \mathbb{R}^n$. In particular, this shows that if f is bounded below, then $\langle f\rangle_\lambda$ is also bounded below.

THEOREM 3.1. *The Gaussian transform $\langle f\rangle_\lambda$ is a continuous function.*

Proof. The proof is a direct consequence of general results (see, for example, Lang [17, Chapter 13]) on the continuity of functions of the form

$$x \mapsto \int_{\mathbb{R}^n} h(x, y)\, dy,$$

where the mapping h is continuous in x and integrable in y. ◻

Theorem 3.1 helps to support our claim that $\langle f\rangle_\lambda$ is a smoother version of f. Indeed, Theorem 3.1 is a special case of a more general result that establishes $\langle f\rangle_\lambda$ as an infinitely differentiable function. This result can be established by showing that the mapping h defined by

$$h(x, y) = f(y)\rho_\lambda(x - y),$$

where ρ_λ is given by (2.2), is infinitely differentiable with respect to x, and all the derivatives are integrable.

We now show that if f is convex, the Gaussian transform is also a convex function. This property is reassuring because it shows that the transformation does not introduce difficulties if none exist.

THEOREM 3.2. *If $f : \mathbb{R}^n \mapsto \mathbb{R}$ is convex, then $\langle f\rangle_\lambda$ is also convex.*

Proof. The result follows from (3.1) because the convexity of f implies that

$$f(\alpha x_1 + (1-\alpha)x_2 + \lambda u) \le \alpha f(x_1 + \lambda u) + (1-\alpha) f(x_2 + \lambda u), \qquad 0 \le \alpha \le 1$$

for any x_1 and x_2 in $\mathbb{R}^n$. ◻

A serious drawback to the general use of the Gaussian transform for minimization is that computing $\langle f\rangle_\lambda$ for a general function defined on $\mathbb{R}^n$ is not practical because this requires the computation of n-dimensional integrals. However, there is a large class of functions for which the computation of the Gaussian transform is reasonable.

DEFINITION 3.3. *A function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is decomposable if f can be written in the form*

$$f(x) = \sum_{k=1}^{m} f_k(x), \qquad f_k(x) = \prod_{j=1}^{n} f_{k,j}(x_j) \tag{3.3}$$

Table 3.1
The Gaussian transformation of elementary functions.

$f(x)$	$\langle f\rangle_\lambda(x)$
x	x
x^2	$x^2+\frac{1}{2}\lambda^2$
$\sin(x)$	$\sin(x)\exp(-\frac{1}{4}\lambda^2)$
$\cos(x)$	$\cos(x)\exp(-\frac{1}{4}\lambda^2)$
$\exp(x)$	$\exp(x)\exp(\frac{1}{4}\lambda^2)$

for some set of functions $\{f_{k,j}\}$, where $f_{k,j}:\mathbb{R}\mapsto\mathbb{R}$.

This class of functions was introduced by Wu [26] under the term generalized multilinear functions; we are using decomposable to avoid confusion with the use of multilinear for a function that is linear in each argument.

The decomposable functions are of interest with respect to the Gaussian transform because computing the Gaussian transform of a decomposable function requires the computation of only one-dimensional integrals. Indeed, a computation shows that if f is defined by (3.3), then

$$\langle f\rangle_\lambda(x)=\sum_{k=1}^{m}\left(\prod_{j=1}^{n}\langle f_{k,j}\rangle_\lambda(x_j)\right).$$

Thus, computing $\langle f\rangle_\lambda$ for a decomposable function requires the computation of only the one-dimensional integrals for each $\langle f_{k,j}\rangle_\lambda$.

Table 3.1 shows the Gaussian transformation of several elementary functions determined by Kostrowicki and Piela [13]. We will justify the correctness of the entries later; here we note that the Gaussian transform of any decomposable function with component functions drawn from this table can be calculated explicitly. For example, using these results, we can show that if f is the general quadratic

$$f(x)=\tfrac{1}{2}x^TQx+c^Tx$$

for some $Q\in\mathbb{R}^{n\times n}$ and $c\in\mathbb{R}^n$, then

$$\langle f\rangle_\lambda(x)=\tfrac{1}{2}x^TQx+c^Tx+\tfrac{1}{4}\lambda^2\left(\sum_{i=1}^{n}q_{i,i}\right). \tag{3.4}$$

In particular, this shows that $\langle f\rangle_\lambda(x)=f(x)$ for linear functions.

Table 3.1 includes only the most commonly occurring functions; there are many other functions with an easily computable Gaussian transform. For example,

$$\langle f\rangle_\lambda(x)=\frac{1}{(\lambda^2+1)^{1/2}}\exp\left(-\frac{x^2}{(\lambda^2+1)}\right)$$

is the Gaussian transform of $f(x)=\exp(-x^2)$.

In addition to quadratic functions, the decomposable functions include the polynomial functions, that is, functions that are linear combinations of terms of the form

$$x_1^{p_1} x_2^{p_2} \cdots x_n^{p_n}$$

for arbitrary integer powers $p_i \geq 0$. The following result is needed to compute $\langle f \rangle_\lambda$ for a polynomial function.

THEOREM 3.4. *If $f : \mathbb{R} \mapsto \mathbb{R}$ is the monic polynomial $f(x) = x^k$, then*

$$\langle f \rangle_\lambda(x) = \sum_{l=0}^{\lfloor k/2 \rfloor} \left(\frac{k!}{(k-2l)!\; l!} \right) \left(\frac{\lambda}{2} \right)^{2l} x^{k-2l}.$$

Proof. Since f is a polynomial, we can expand $f(x + \lambda u)$ in (3.1) and obtain that

$$\langle f \rangle_\lambda(x) = \frac{1}{\pi^{1/2}} \sum_{j=0}^{k} f^{(j)}(x) \frac{\lambda^j}{j!} \int_{\mathbb{R}} u^j \exp\left(-\|u\|^2\right)\, du$$

and, since the integrals with odd powers vanish by symmetry,

$$\langle f \rangle_\lambda(x) = \frac{1}{\pi^{1/2}} \sum_{l=0}^{\lfloor k/2 \rfloor} f^{(2l)}(x) \frac{\lambda^{2l}}{(2l)!} \int_{\mathbb{R}} u^{2l} \exp\left(-\|u\|^2\right)\, du.$$

We can complete the proof if we show that

$$\frac{1}{\pi^{1/2}} \int_{\mathbb{R}} u^{2l} \exp\left(-\|u\|^2\right)\, du = \frac{(2l)!}{4^l l!}. \tag{3.5}$$

This identity can be established by defining I_{2l} as the integral in (3.5) and noting that integration by parts yields

$$I_{2l} = \frac{2l-1}{2} I_{2l-2} = \frac{(2l)(2l-1)}{4l} I_{2l-2}.$$

An induction argument, based on this relationship and using the result $I_0 = 1$, shows that (3.5) holds, and thus completes the proof. □

Theorem 3.4 was obtained by Kostrowicki and Piela [13], but it had a completely different approach. We will elaborate on this point below.

We can extend Theorem 3.4 by noting that if f is analytic, the Taylor series of $f(x + \lambda u)$ as a function of u converges for all λu. Thus we can proceed as in the proof of Theorem 3.4 to obtain

$$\langle f \rangle_\lambda(x) = \frac{1}{\pi^{1/2}} \sum_{l=0}^{+\infty} f^{(2l)}(x) \frac{\lambda^{2l}}{(2l)!} \int_{\mathbb{R}} u^{2l} \exp\left(-\|u\|^2\right)\, du.$$

Hence, (3.5) shows that

$$\langle f \rangle_\lambda(x) = \sum_{l=0}^{+\infty} \frac{1}{l!} f^{(2l)}(x) \left(\frac{\lambda}{2} \right)^{(2l)}. \tag{3.6}$$

This relationship holds, in particular, for the functions in Table 3.1. A short computation shows that this expression justifies the entries in this table.

Expression (3.6) was used by Piela, Kostrowicki, and Scheraga [19] to define the transformation for the diffusion equation method. A disadvantage of this definition is that it requires an analytic f, while (3.1) requires only the integrability of f. On the other hand, as we have noted, this expression is quite useful for determining the Gaussian transform of several important functions. In particular, Kostrowicki and Piela [13] obtained Theorem 3.4 with this approach.

The Gaussian transform for functions that are related by a scaling or a translation of the variables can be computed by noting that if

$$f_0(x) = f(\alpha x - x_0)$$

for some scalar α and vector x_0, then

$$\langle f_0 \rangle_\lambda(x) = \langle f \rangle_{\alpha\lambda}(\alpha x - x_0).$$

For example, if $f(x) = \sin(\alpha x)$, then

$$\langle f \rangle_\lambda(x) = \sin(\alpha x) \exp\left(-\tfrac{1}{4}(\alpha\lambda)^2\right).$$

As noted by Piela, Kostrowicki, and Scheraga [19], this result suggests that $\langle f \rangle_\lambda$ tends to dampen the high-frequency components in a function, since if α is large, then the exponential term produces a larger damping effect. See Wu [26, section 4] for a discussion of the effect of the Gaussian transform on the high-frequency components of a general function.

We have defined the Gaussian transform of a real-valued function $f : \mathbb{R}^n \mapsto \mathbb{R}$ by (3.1), but this definition extends immediately to vector-valued functions. This remark is of interest because in addition to transforming the function, we could also transform the gradient and the Hessian of f. We now show that the Gaussian transform of the gradient (Hessian) is the gradient (Hessian) of $\langle f \rangle_\lambda$. This result can be deduced by differentiating under the integral sign in (3.1) to obtain that

$$\nabla \langle f \rangle_\lambda(x) = \frac{1}{\pi^{n/2}} \int_{\mathbb{R}^n} \nabla f(x + \lambda u) \exp\left(-\|u\|^2\right) du = \langle \nabla f \rangle_\lambda(x), \tag{3.7}$$

which is the desired result for the gradient. If we repeat the process, we obtain that

$$\nabla^2 \langle f \rangle_\lambda(x) = \frac{1}{\pi^{n/2}} \int_{\mathbb{R}^n} \nabla^2 f(x + \lambda u) \exp\left(-\|u\|^2\right) du = \langle \nabla^2 f \rangle_\lambda(x), \tag{3.8}$$

so that the Gaussian transform of the Hessian matrix is the Hessian of $\langle f \rangle_\lambda$.

We guarantee the validity of differentiating under the integral sign in (3.8) by assuming that $\nabla^2 f$ is continuous almost everywhere and that

$$\|\nabla^2 f(x)\| \le \gamma_1 \exp(\gamma_2 \|x\|) \tag{3.9}$$

holds for some positive constants γ_1 and γ_2. This result requires a technical lemma.

LEMMA 3.5. *If $f : \mathbb{R}^n \mapsto \mathbb{R}$ is twice differentiable on $\mathbb{R}^n$ and* (3.9) *holds for some positive constants γ_1 and γ_2, then*

$$\|\nabla f(x)\| \le 2\beta_1 \exp\left(\beta_2 \|x\|\right), \qquad |f(x)| \le 3\beta_1 \exp\left(\beta_2 \|x\|\right),$$

where $\beta_1 \ge \max\{\gamma_1, \|\nabla f(0)\|, |f(0)|\}$ and $\beta_2 \ge 2 + \gamma_2$.

Proof. The standard estimate

$$\|\nabla f(x) - \nabla f(0)\| \leq \sup_{0\leq\tau\leq 1} \|\nabla^2 f(\tau x)\| \, \|x\|,$$

together with the bound $\|x\| \leq \exp(\|x\|)$, implies that

$$\|\nabla f(x)\| \leq \|\nabla f(0)\| + \gamma_1 \exp\Bigl(\gamma_2\|x\|\Bigr)\|x\| \leq \beta_1 + \beta_1 \exp\Bigl((1+\gamma_2)\|x\|\Bigr),$$

and thus

$$\|\nabla f(x)\| \leq 2\beta_1 \exp\Bigl((1+\gamma_2)\|x\|\Bigr),$$

which is clearly of the desired form. We complete the proof by using this inequality and repeating the above argument, but with ∇f replaced by f. In this case we obtain

$$|f(x)| \leq |f(0)| + 2\beta_1 \exp\Bigl((1+\gamma_2)\|x\|\Bigr)\|x\| \leq \beta_1 + 2\beta_1 \exp\Bigl((2+\gamma_2)\|x\|\Bigr),$$

as desired. □

We now show that assumption (3.9) guarantees that (3.7) and (3.8) hold.

THEOREM 3.6. *If $f : \mathbb{R}^n \mapsto \mathbb{R}$ is twice continuously differentiable almost everywhere on $\mathbb{R}^n$ and* (3.9) *holds for some positive constants γ_1 and γ_2, then*

$$\nabla\langle f\rangle_\lambda(x) = \langle\nabla f\rangle_\lambda(x), \qquad \nabla^2\langle f\rangle_\lambda(x) = \langle\nabla^2 f\rangle_\lambda(x).$$

Proof. Assumption (3.9) guarantees that the function

$$u \mapsto \nabla^2 f(x+\lambda u)\exp\left(-\|u\|^2\right)$$

is bounded by an integrable function for any fixed x and λ. The validity of (3.8) now follows from standard results that guarantee differentiation under the integral sign (see, for example, Lang [17, Chapter 13]). Lemma 3.5 shows that the same argument can be used to validate (3.7). □

Theorem 3.6 was stated informally by Wu [26]; the above argument supplies the pieces needed to give a formal proof of this result. Theorem 3.6 is of interest from a computational viewpoint because optimization algorithms usually require the gradient of $\langle f\rangle_\lambda$; Newton methods also require the Hessian matrix. This result shows that the gradient and Hessian of $\langle f\rangle_\lambda$ are also smooth functions in the sense that they are obtained by transforming the gradient and Hessian of f, respectively.

In this section we have concentrated on obtaining explicit expressions for the Gaussian transform of various functions. We have also experimented with other approaches. In one of the approaches, the Gaussian transform is approximated by a Gaussian quadrature. This approach hinges on the ability to evaluate Gaussian integrals efficiently with ORTHOPOL (Gautschi [6]). Another approach is based on approximating the function by a decomposable function and using the Gaussian transform of the decomposable function as an approximation to the Gaussian transform of the original function. We plan to pursue these approaches in future work.

4. The Gaussian transform for the distance geometry problem.

Our continuation algorithms for the distance geometry problem are based on the function

$$f(x) = \sum_{i,j\in\mathcal{S}} w_{i,j}\left(\|x_i - x_j\|^2 - \delta_{i,j}^2\right)^2, \tag{4.1}$$

where $w_{i,j}$ are positive weights and $\delta_{i,j}$ are distances. Computing the Gaussian transform of (4.1) is not difficult because f is decomposable. In fact, f is a polynomial function in the components of x. The development below shows that f has considerable structure and that this structure can be used to simplify the computation for the Gaussian transform.

In the standard formulation of the distance geometry problem, the components x_i of x belong to $\mathbb{R}^3$. We assume that $x_i \in \mathbb{R}^p$ because this assumption does not lead to extra complications. We thus consider the general problem where f is of the form

$$f(x) = \sum_{i,j \in \mathcal{S}} w_{i,j} h_{i,j}(x_i - x_j) \tag{4.2}$$

and $h_{i,j} : \mathbb{R}^p \mapsto \mathbb{R}$ is defined by

$$h_{i,j}(x) = \left(\|x\|^2 - \delta_{i,j}^2 \right)^2 . \tag{4.3}$$

The following result shows that computing the Gaussian transform of (4.2) requires only the Gaussian transform on $h_{i,j}$.

THEOREM 4.1. *If $f : \mathbb{R}^n \mapsto \mathbb{R}$ and $h : \mathbb{R}^p \mapsto \mathbb{R}$ are related by*

$$f(x) = h(P^T x)$$

for some matrix $P \in \mathbb{R}^{n \times p}$ such that $P^T P = \sigma^2 I$, then

$$\langle f \rangle_\lambda (x) = \langle h \rangle_{\sigma\lambda} (P^T x).$$

Proof. Define $Q \in \mathbb{R}^{n \times (n-p)}$ such that

$$R = \frac{1}{\sigma} \begin{pmatrix} P & Q \end{pmatrix}$$

is an orthogonal matrix. By definition,

$$\langle f \rangle_\lambda (x) = \frac{1}{\pi^{n/2}} \int_{\mathbb{R}^n} h(P^T x + \lambda P^T u) \exp\left(-\|u\|^2 \right) \, du,$$

so if we make the change of variables $u \mapsto Rv$ in (3.1), we obtain

$$\langle f \rangle_\lambda (x) = \frac{1}{\pi^{n/2}} \int_{\mathbb{R}^n} h(P^T x + \lambda P^T R v) \exp\left(-\|v\|^2 \right) \, dv,$$

since R is an orthogonal matrix. Now note that $P^T R = \sigma \, (I \quad 0)$, and thus the above integral reduces to an integral over $\mathbb{R}^p$; that is,

$$\langle f \rangle_\lambda (x) = \frac{1}{\pi^{p/2}} \int_{\mathbb{R}^p} h(P^T x + \lambda \sigma v) \exp\left(-\|v\|^2 \right) \, dv = \langle h \rangle_{\sigma\lambda} (P^T x). \qquad \square$$

The application of Theorem 4.1 to the distance geometry problem requires that we specify how the vectors x_i are related to x. Let the ith component of the vector x_j be the $c(i,j)$ components of x. In other words, $c(i,j)$ specifies how the components of x_j are stored in $x \in \mathbb{R}^n$. Another way of defining $c(i,j)$ is by the relationship

$$[x]_{c(i,j)} = [x_j]_i .$$

With this choice we can define $P \in \mathbb{R}^{n \times p}$ by

$$P = \left(e_{c(1,i)} - e_{c(1,j)}, \ldots, e_{c(p,i)} - e_{c(p,j)}\right)$$

and obtain $P^T x = x_i - x_j$. In particular, $P^T P = \sigma^2 I$, where $\sigma^2 = 2$.

As an application of these results, note that Theorem 4.1 implies that

$$\langle f \rangle_\lambda(x) = \langle h \rangle_{\sqrt{2}\lambda}(x_i - x_j)$$

is the Gaussian transform of $f(x) = h(x_i - x_j)$. We can apply this result to the distance geometry problem, where h is given by (4.3), by computing the Gaussian transform of the functions $f_1 : \mathbb{R}^p \mapsto \mathbb{R}$ and $f_2 : \mathbb{R}^p \mapsto \mathbb{R}$ defined by

$$f_1(x) = \|x\|^2, \qquad f_2(x) = \|x\|^4.$$

Since f_1 is a quadratic,

$$\langle f_1 \rangle_\lambda(x) = \|x\|^2 + \tfrac{1}{2} p \lambda^2 \tag{4.4}$$

is just a special case of (3.4). We now claim that Theorem 3.4 shows that

$$\langle f_2 \rangle_\lambda(x) = \|x\|^4 + [3 + (p-1)]\lambda^2 \|x\|^2 + \tfrac{1}{4} p(p+2)\lambda^4. \tag{4.5}$$

We prove (4.5) by noting that

$$\left(\sum_{i=1}^{p} x_i^2 \right)^2 = \sum_{i=1}^{p} x_i^4 + \sum_{i \neq j}^{p} \left(x_i^2 x_j^2 \right),$$

and thus Theorem 3.4 implies that

$$\langle f_2 \rangle_\lambda(x) = \sum_{i=1}^{p} \left(x_i^4 + 3\lambda^2 x_i^2 + \tfrac{3}{4}\lambda^4 \right) + \sum_{i \neq j}^{p} \left((x_i^2 + \tfrac{1}{2}\lambda^2)(x_j^2 + \tfrac{1}{2}\lambda^2) \right).$$

Identity (4.5) is now a direct consequence of this expression.

THEOREM 4.2. *If $h : \mathbb{R}^p \mapsto \mathbb{R}$ is defined by*

$$h(x) = \left(\|x\|^2 - \delta^2 \right)^2,$$

then

$$\langle h \rangle_\lambda(x) = h(x) + [3 + (p-1)]\lambda^2 \|x\|^2 + \tfrac{1}{4} p(p+2)\lambda^4 - p\delta^2\lambda^2.$$

Proof. Since

$$h(x) = f_2(x) - 2\delta^2 f_1(x) + \delta^4,$$

the result follows from (4.4) and (4.5). □

The computation of the Gaussian transform for the distance geometry problem now follows from the results that we have obtained.

THEOREM 4.3. *If $f : \mathbb{R}^n \mapsto \mathbb{R}$ is defined by* (4.2) *and* (4.3)*, then*

$$\langle f \rangle_\lambda(x) = f(x) + \sum_{i,j \in \mathcal{S}} \left(2w_{i,j} [3 + (p-1)]\lambda^2 \|x_i - x_j\|^2 \right) + \gamma,$$

where γ is the constant

$$\gamma = \sum_{i,j \in \mathcal{S}} \left(p(p+2)\lambda^4 - 2p\delta_{i,j}^2 \lambda^2 \right) w_{i.j}.$$

Proof. Recall that we can write f in the form

$$f(x) = \sum_{i,j \in \mathcal{S}} w_{i,j} h_{i,j}(P_{i,j} x),$$

where $P_{i,j}^T P_{i,j} = \sigma^2 I$ with $\sigma^2 = 2$. □

Theorem 4.3 shows that the Gaussian transform of the distance geometry function defined by (4.2) and (4.3) can be computed quite easily. Moreover, this result also shows that the gradient and the Hessian matrix of the Gaussian transform are also readily computable at a fractional increase in cost.

We conclude this section by discussing the relationship between Theorem 4.1 and the anisotropic Gaussian transform defined by Wu [26]. Given a nonsingular matrix $\Lambda \in \mathbb{R}^{n \times n}$, the anisotropic Gaussian transform of f is defined by

$$\langle f \rangle_\Lambda (x) = \frac{1}{\pi^{n/2} |\det \Lambda|} \int_{\mathbb{R}^n} f(y) \exp\left(-\|\Lambda^{-1}(y-x)\|^2 \right) \, dy. \tag{4.6}$$

Clearly, this transformation generalizes Definition 2.2, where $\Lambda = \lambda I$.

From a computational viewpoint, the anisotropic transformation is important when Λ is a diagonal matrix, and it is closely related to the isotropic transformation when f is a decomposable function. In particular, if f is defined by (3.3) and $\Lambda = \operatorname{diag}(\lambda_j)$, then

$$\langle f \rangle_\Lambda (x) = \sum_{k=1}^{m} \left(\prod_{j=1}^{n} \langle f_{k,j} \rangle_{\lambda_j}(x_j) \right).$$

The following result, a generalization of Theorem 4.1, provides further motivation for the anisotropic transformation.

THEOREM 4.4. *If $f : \mathbb{R}^n \mapsto \mathbb{R}$ and $h : \mathbb{R}^p \mapsto \mathbb{R}$ are related by*

$$f(x) = h(P^T x)$$

for some matrix $P \in \mathbb{R}^{n \times p}$ such that $P^T P = D^T D$ where D is a diagonal matrix, then

$$\langle f \rangle_\lambda (x) = \langle h \rangle_{\lambda D} (P^T x).$$

Proof. The proof follows that of Theorem 4.1. In this case we define $Q \in \mathbb{R}^{(n-p) \times n}$ such that

$$R = \begin{pmatrix} PD^{-1} & Q \end{pmatrix}$$

is an orthogonal matrix and obtain that

$$\langle f \rangle_\lambda (x) = \frac{1}{\pi^{p/2}} \int_{\mathbb{R}^p} h(P^T x + \lambda D v) \exp\left(-\|v\|^2 \right) \, dv.$$

The result now follows from the definition of the anisotropic transformation because the change of variables $y \mapsto x + Du$ in (4.6) shows that

$$\langle h \rangle_D(x) = \frac{1}{\pi^{n/2}} \int_{\mathbb{R}^n} h(x + Du) \exp\left(-\|u\|^2\right)\, du$$

is the anisotropic transformation of h. □

5. Continuation algorithms.

The basic idea behind the continuation approach is to trace a curve $\{x(\lambda) : \lambda \geq 0\}$, where each $x(\lambda)$ is a minimizer of $\langle f \rangle_\lambda$. In the simplest approach we choose a sequence $\{\lambda_k\}$ of smoothing parameters that converges to zero and compute a minimizer x_k of each $\langle f \rangle_{\lambda_k}$. A more sophisticated approach is to rely on a differential equation to trace the curve. For this approach, we define $h : \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}$ by

$$h(x, \lambda) = \langle f \rangle_\lambda(x) \tag{5.1}$$

and note that, since $x(\lambda)$ is a stationary point of $\langle f \rangle_\lambda$,

$$\partial_x h[x(\lambda), \lambda] = 0.$$

We now differentiate with respect to λ to obtain

$$\partial_{xx} h[x(\lambda), \lambda] x'(\lambda) + \partial_{\lambda x} h[x(\lambda), \lambda] = 0.$$

This differential equation, together with an initial value x_0, defines a curve if the coefficient matrix $\partial_{xx} h[x(\lambda), \lambda]$ is nonsingular. In this paper we concentrate on the approach based on choosing a predetermined sequence of smoothing parameters, since this approach already brings out the power of continuation algorithms.

We wish to analyze the ideal situation where we are able to determine a global minimizer x_k of $\langle f \rangle_{\lambda_k}$ for some sequence $\{\lambda_k\}$ converging to zero. This requires that we show that the function $h : \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}$ defined by (5.1) is continuous on $\mathbb{R}^n \times \mathbb{R}$. Without loss of generality, we show continuity at $(x^*, 0)$. We had previously noted the continuity of h with respect to x and λ; we now establish the joint continuity with respect to (x, λ).

LEMMA 5.1. *Assume that $f : \mathbb{R}^n \mapsto \mathbb{R}$ is continuous on $\mathbb{R}^n$ and satisfies* (3.2). *If $\{x_k\}$ converges to x^* and $\{\lambda_k\}$ converges to zero, then*

$$\lim_{k \to +\infty} \langle f \rangle_{\lambda_k}(x_k) = f(x^*).$$

Proof. Let B_r be the ball of radius r centered at the origin, and let C_r be the complement of B_r, that is,

$$C_r = \{x \in \mathbb{R}^n : \|x\| > r\}.$$

We first show that for any $\epsilon > 0$ we can choose $r > 0$ and k_0 so that

$$\int_{C_r} |f(x_k + \lambda_k u) - f(x^*)| \exp\left(-\|u\|^2\right)\, du \leq \epsilon, \qquad k \geq k_0. \tag{5.2}$$

Assumption (3.2) implies that there is a constant $\mu > 0$ such that

$$|f(x_k + \lambda_k u) - f(x^*)| \leq \mu \exp\left(\lambda_k \|u\|\right),$$

and since $\lambda \|u\| \le \frac{1}{2}\|u\|^2$ for $\lambda \le \frac{1}{2}$ and $\|u\| \ge 1$,

$$\int_{C_r} |f(x_k + \lambda_k u) - f(x^*)| \exp\left(-\|u\|^2\right)\, du \le \mu \int_{C_r} \exp\left(-\tfrac{1}{2}\|u\|^2\right)\, du$$

if $\lambda_k \le \frac{1}{2}$ and $r \ge 1$. This inequality proves (5.2) because, if r is sufficiently large, the integral of $\exp\left(-\frac{1}{2}\|u\|^2\right)$ over C_r is arbitrarily small. Now note that the continuity of f at x^* shows that for given r and k_0 we can choose $k_1 \ge k_0$ so that

$$\int_{C_r} |f(x_k + \lambda_k u) - f(x^*)| \exp\left(-\|u\|^2\right)\, du \le \epsilon, \qquad k \ge k_1.$$

This inequality and (5.2) imply that

$$|\langle f \rangle_{\lambda_k}(x_k) - f(x^*)| \le 2\epsilon, \qquad k \ge k_1,$$

which is the desired result. □

A variation on Lemma 5.1 would be to show that the gradient and Hessian matrix of h are continuous. The proof of this variation would be entirely similar to the one for Lemma 5.1.

THEOREM 5.2. *Assume that $f : \mathbb{R}^n \mapsto \mathbb{R}$ is continuous on $\mathbb{R}^n$ and satisfies* (3.2). *Let $\{\lambda_k\}$ be any sequence converging to zero. If x_k is a global minimizer of $\langle f \rangle_{\lambda_k}$ and $\{x_k\}$ converges to x^*, then x^* is a global minimizer of f.*

Proof. Since x_k is a global minimizer of $\langle f \rangle_{\lambda_k}$,

$$\langle f \rangle_{\lambda_k}(x_k) \le \langle f \rangle_{\lambda_k}(x), \qquad x \in \mathbb{R}^n.$$

Lemma 5.1 now implies that $f(x^*) \le f(x)$ for any $x \in \mathbb{R}^n$. Hence, x^* is a global minimizer of f. □

Given λ_k, we need an algorithm to determine a minimizer x_k of $\langle f \rangle_{\lambda_k}$. A trust region version of Newton's method based on the work of Moré and Sorensen [18] is an attractive choice because it has strong global and local convergence properties.

At each iteration of a trust region Newton method for the minimization of $f : \mathbb{R}^n \mapsto \mathbb{R}$, we have an iterate x_k, a bound Δ_k, a scaling matrix D_k, and a quadratic model $q_k : \mathbb{R}^n \mapsto \mathbb{R}$ of the possible reduction $f(x_k + w) - f(x_k)$ for $\|D_k w\| \le \Delta_k$. The developments in section 4 show that the gradient and Hessian matrix easily can be obtained for the distance geometry problem. Thus

$$q_k(w) = \nabla f(x_k)^T w + \tfrac{1}{2} w^T \nabla^2 f(x_k) w$$

is our choice for the quadratic model.

An important ingredient in a trust region method is the choice of step s_k. In general, s_k is an approximate solution to the trust region subproblem

$$\min \left\{ q_k(w) : \|D_k w\| \le \Delta_k \right\}$$

with $q_k(s_k) < 0$. We use the algorithm described by Moré and Sorensen [18] because it provides an approximate global solution to the subproblem. In particular, if x_k is a saddle point so that $\nabla f(x_k) = 0$ and $\nabla^2 f(x_k)$ is indefinite, we still have $q_k(s_k) < 0$.

Given the step s_k, the test for acceptance of the trial point $x_k + s_k$ depends on a parameter $\eta_0 > 0$. The following algorithm summarizes the main computational steps:

```
For $k = 0, 1, \ldots,$ maxiter
    Compute the quadratic model $q_k$.
    Compute a scaling matrix $D_k$.
    Compute an approximate solution $s_k$ to the trust region subproblem.
    Compute the ratio $\rho_k$ of actual to predicted reduction.
    Set $x_{k+1} = x_k + s_k$ if $\rho_k \geq \eta_0$; otherwise set $x_{k+1} = x_k$. Update $\Delta_k$.
```

Given a step s_k such that $\|D_k s_k\| \leq \Delta_k$ and $q_k(s_k) < 0$, the rules for updating the iterate x_k and the bound Δ_k depend on the ratio

$$\rho_k = \frac{f(x_k + s_k) - f(x_k)}{q_k(s_k)}$$

of the actual reduction in the function to the predicted reduction in the model. See, for example, Moré and Sorensen [18] for details on these rules.

The trust region method outlined above is attractive for the distance geometry problem provided the number of molecules m is moderate, say $m \leq 50$. For larger problems we still can use the trust region method, provided that the set $\mathcal{S}$ in (4.1) is sparse and the computation of the step s_k makes use of sparsity. We plan to address this case in future work.

6. Numerical results. We present numerical results for two model problems based on a molecule with $m = s^3$ atoms located in the three-dimensional lattice

$$\{(i_1, i_2, i_3) : 0 \leq i_1 < s,\ 0 \leq i_2 < s,\ 0 \leq i_3 < s\}$$

for some integer $s \geq 1$. Figure 6.1 shows a molecule with 64 atoms ($s = 4$). For both model problems we consider a distance geometry problem of the form

$$\|x_i - x_j\| = \delta_{i,j}, \qquad (i,j) \in \mathcal{S}, \tag{6.1}$$

where $\mathcal{S}$ is a subset of the pairwise distances $\delta_{i,j}$ between atoms i and j.

Reasonable choices of the set $\mathcal{S}$ depend on the source of the data. NMR data usually produces information for atoms located in chains of relatively close atoms. Distance data for atoms that are far away is available, but it tends to be less accurate. Other sources of distance geometry problems yield data for all atoms. For example, the embed algorithm (see Crippen and Havel [4] and Havel [9]) approaches the general distance geometry problem (1.2) by solving a sequence of exact distance geometry problems where all pairwise distances are included.

Both model problems that we consider try to capture various features in distance data from applications. The first model problem has distance data for both near and relatively far away atoms, while the second problem only has distance data for nearby atoms.

In the first model problem we specify an ordering for the atoms in this molecule by letting atom i be the atom at position (i_1, i_2, i_3), where

$$i = 1 + i_1 + s i_2 + s^2 i_3,$$

and define the set $\mathcal{S}$ in terms of an integer r by

$$\mathcal{S} = \{(i,j) : |i - j| \leq r\}. \tag{6.2}$$

With this definition, the set $\mathcal{S}$ is sparse in the sense that it contains about rm pairs out of a possible m^2 pairs. Figure 6.2 shows an 8-atom problem defined by a sparse $\mathcal{S}$ with $r = 3$.

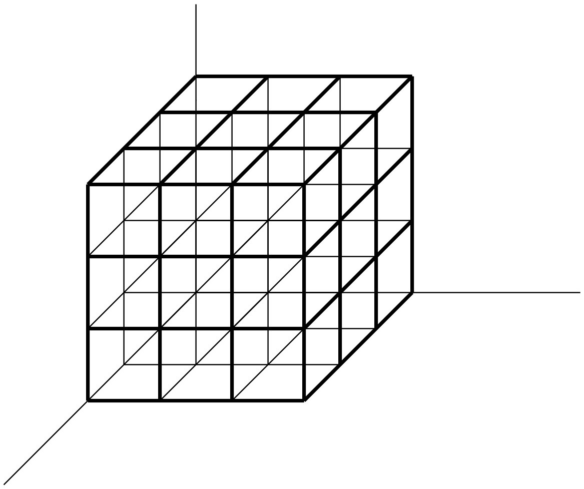


FIG. 6.1. *An example lattice structure of* 64 *atoms.*

Our construction shows that the distance geometry problem defined by (6.1) and (6.2) always has at least one solution for any value of r. Since in our numerical results we choose $r = s^2$, there is data for atoms at positions (i_1, i_2, i_3) and (j_1, j_2, j_3) only if $|i_3 - j_3| \leq 1$. This means that these atoms must be either on the same plane perpendicular to the i_3-axis or on adjacent planes that are one unit apart and perpendicular to the i_3-axis. Thus, in this model problem there is distance data for both near and relatively far away atoms.

We attack the distance geometry problem by using the global continuation approach to obtain a global minimum of the function

$$f(x) = \sum_{(i,j)\in\mathcal{S}} (\|x_i - x_j\|^2 - \delta_{i,j}^2)^2, \tag{6.3}$$

where $\delta_{i,j}$ is the distance between atoms i and j in the lattice. We need the Gaussian transform of f and, for the trust region Newton method, the gradient and Hessian matrix of the transform. Theorem 4.3 shows that

$$\langle f \rangle_\lambda(x) = \sum_{(i,j)\in\mathcal{S}} \big((\|x_i - x_j\|^2 - \delta_{i,j}^2)^2 + 10\lambda^2 \|x_i - x_j\|^2\big) + \gamma \tag{6.4}$$

is the Gaussian transform of f, where γ is a constant. The gradient and Hessian matrix can be obtained from this expression.

The parameter λ in the Gaussian transform (6.4) must be chosen with care. In particular, if λ is chosen so that the element functions

$$h_{i,j}(x) = (\|x_i - x_j\|^2 - \delta_{i,j}^2)^2 + 10\lambda^2 \|x_i - x_j\|^2$$

in (6.4) are convex, then minimization of $\langle f \rangle_\lambda$ is obtained only if all the atoms collapse into one atom; that is, all global minimizers of (6.4) have $x_1 = \cdots = x_m$. We elaborate on this remark when we discuss the implementation of the continuation algorithm. To determine λ so that $h_{i,j}$ is convex, consider the function

$$h(r) = \big(r^2 - \delta^2\big)^2 + 10\lambda^2 r^2.$$

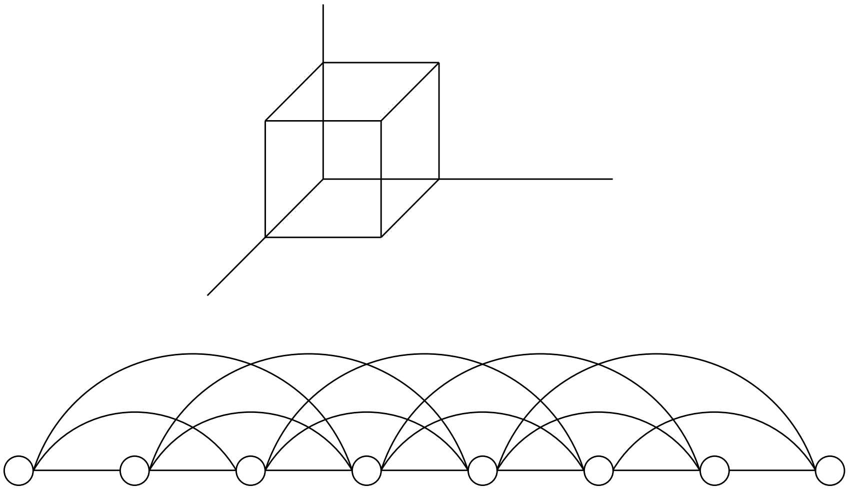


FIG. 6.2. *An example lattice structure with sparse distance constraints.*

Note that if h is increasing and convex, then $x \mapsto h(\|x_i - x_j\|)$ is convex, and thus $h_{i,j}$ is convex. We choose λ so that h is increasing and convex by noting that

$$h'(r) = 4r^3 + 4r(5\lambda^2 - \delta^2), \qquad h''(r) = 12r^2 - 4\delta^2 + 20\lambda^2$$

implies that we must have $\lambda \geq (\frac{1}{5})^{1/2}\delta$. These computations also show that if

$$\lambda \geq (\tfrac{1}{5})^{1/2} \max\{\delta_{i,j} : (i,j) \in \mathcal{S}\},$$

then the Gaussian transform $\langle f \rangle_\lambda$ in (6.4) is convex.

In this paper we have shown that the continuation method has strong theoretical properties. We now use numerical results to show that the continuation method is superior to the multistart approach, a standard procedure for finding the global minimizer of f.

We are interested in the solution of problems with a large number of atoms, and thus we performed our numerical testing on the Argonne IBM SP system. This system has 128 nodes, where each node is an IBM RS/6000-370 with 128 MB of memory. We defer a discussion of the parallel aspects of our approach to future work.

In the multistart method we choose a random starting point x_s and use the trust region method from this starting point to determine a local minimizer x_s^*. If x_s^* satisfies

$$\Big| \|x_i - x_j\| - \delta_{i,j} \Big| \leq \epsilon, \qquad (i,j) \in \mathcal{S} \tag{6.5}$$

for some tolerance ϵ, then x_s^* is declared to be a solution to the distance geometry problem (6.1), and we terminate the multistart method. If x_s^* does not satisfy (6.5), we repeat the procedure with another starting point. The multistart method fails if (6.5) is not satisfied after trying ten starting points.

The global continuation method that we use is similar to the multistart method, except that the continuation algorithm of section 5 is used to determine a local minimizer x_s^* of f. We start the continuation algorithm with the random starting point x_s and $\lambda_0 > 0$. We compute p major iterations, where p is the number of continuation

Table 6.1

Performance for the multistart and continuation method ($\lambda_0 = 0.5$, $p = 10$).

		Multistart		Continuation		Multistart		Continuation	
m	r	$nfev$	$ngev$	$nfev$	$ngev$	$nfev$	$ngev$	$nfev$	$ngev$
27	9	573	472	255	216	273	229	221	188
64	16	$^{\texttt{F}}$1211	1009	886	710	1102	917	863	698
125	25	1810	1461	390	304	1600	1324	410	322
216	36	$^{\texttt{F}}$3397	2782	550	421	$^{\texttt{F}}$3416	2802	446	337
		$x_s \in \mathbf{rand}(B)$				$x_s \in 2\,\mathbf{rand}(B)$			

steps. The kth major iterate x_k is computed by applying a trust region algorithm, with x_{k-1} as a starting point, to the transformed function $\langle f \rangle_{\lambda_k}$, where

$$\lambda_k = \left(1 - \frac{k}{p}\right)\lambda_0 .$$

Since $\lambda_p = 0$, the final major iterate x_p is a local minimizer of f, so we set $x_s^* = x_p$.

In Table 6.1 we present the results obtained by the global continuation method and the multistart method on two sets of starting points. The number of molecules in these tables are of the form $m = s^3$ for $3 \le s \le 6$. The parameter r in (6.2) was set to $r = s^2$.

Since the solutions of the distance geometry problems defined by (6.1) and (6.2) lie in

$$B = \left\{ x \in \mathbb{R}^n : 0 \le x_i \le s - 1 \right\},$$

it is reasonable to choose the starting points randomly in B by setting each component of the starting point to a random number in $(0, s-1)$. In Table 6.1 we present results when the starting point is chosen randomly in B and $2B$.

For these results we used $\lambda_0 = 0.5$ and $p = 10$ continuation steps. An automatic choice of λ_0 for these problems is not clear; in particular, below we point out that $\lambda_0 = 0.5$ is too small for some problems. Also note that if we choose λ_0 large, then $\langle f \rangle_{\lambda_0}$ is convex, and thus the local minimizer x_s^* obtained by the continuation approach is independent of the random starting point x_s. This is clearly an undesirable situation. We plan to address the automatic selection of λ_0 in future work.

Performance is measured in terms of the number of function and gradient evaluations, `nfev` and `ngev`, used to find a global minimizer. The results marked by $^{\texttt{F}}$ are the cases where no global minimizer was found after trying 10 starting points.

We have not included execution times in Table 6.1 because the distance geometry problems under consideration give rise to sparse minimization problems, but the algorithm that we have used does not take advantage of sparsity. Our concern in this paper is mainly with the ability of the continuation method to solve these problems with a reasonable number of function and gradient evaluations. In future work we will consider problems with more atoms and the use of algorithms that take advantage of sparsity.

These results show that the continuation method finds a global minimizer in all cases and with fewer function and gradient evaluations than the multistart method.

Table 6.2
Probability of success when the constraint set $\mathcal{S}$ is defined by (6.2).

m	r	Multistart	Continuation	Multistart	Continuation
27	9	10%	100%	60%	100%
64	16	0%	70%	10%	50%
125	25	10%	100%	10%	100%
216	36	0%	100%	0%	100%
		$x_s \in \mathbf{rand}(B)$		$x_s \in 2\,\mathbf{rand}(B)$	

Moreover, the performance of the continuation method seems to be relatively insensitive to the choice of starting point. The multistart method, on the other hand, requires a large number of function and gradient evaluations to determine a global minimizer and is unable to find a global minimizer for problems with $m = 216$ atoms. Also note that the performance of the multistart method seems to be sensitive to the choice of starting point.

The reliability of the continuation and multistart methods can be measured by the probability of success of these methods, that is, the percentage of successful runs (the global minimizer is found) in all 10 starting points. The results in Table 6.2 clearly show that the multistart method had little success in finding a global minimizer, especially for problems with $m \geq 64$ atoms. However, the continuation method succeeded 100% in most of the cases. Even for $m = 64$, the probability of success is much higher for the continuation method.

One might wonder why the continuation method was not able to find the global minimizer for $m = 64$ in all 10 runs. A simple answer to this question is that the initial $\lambda_0 = 0.5$ value was too small for smoothing the function in this problem. Therefore, we repeated the runs for the problem with $m = 64$ atoms but with $\lambda_0 = 1$ and $p = 20$. The continuation method then found the global minimizer for all 10 starting points.

The results in Table 6.3 compare the average performance of the multistart and the continuation method when $\lambda_0 = 1$ and $p = 20$. When $m = 64$ and $x_s \in \mathbf{rand}(B)$, the multistart method fails in all cases, so the results in Table 6.3 measure the effort required to find a local minimizer. In contrast, the continuation method succeeds in all cases, so the results measure the effort required to find a global minimizer. This is interesting because, in general, we expect the effort required to find a global minimizer to be much larger than the effort needed to find a local minimizer. A similar conclusion is reached when $m = 64$ and $x_s \in 2\,\mathbf{rand}(B)$, since in this case the multistart method only succeeds in one case. When $m = 216$, the additional effort (measured by the number of function and gradient evaluations) required to find a global minimizer is less than 30% of the effort required to find a local minimizer.

We conclude this section by presenting our numerical results for the second model problem. In this problem distance data is generated for all pairs of atoms in

$$\mathcal{S} = \left\{ (i,j) : \|x_i - x_j\| \leq r^{1/2} \right\}. \tag{6.6}$$

Distance data defined by (6.6) is similar to distance data defined by (6.2) with $r = s^2$ because if $(i,j) \in \mathcal{S}$, where $\mathcal{S}$ is defined by (6.2), then

$$\|x_i - x_j\| \leq \sqrt{1 + 2(s-1)^2} \leq (2r)^{1/2}.$$

Table 6.3
Average performance for the multistart and continuation method ($\lambda_0 = 1$, $p = 20$).

		Multistart		Continuation		Multistart		Continuation	
m	r	$nfev$	$ngev$	$nfev$	$ngev$	$nfev$	$ngev$	$nfev$	$ngev$
27	9	61.2	50.5	251.1	211.1	57.4	98.7	240.9	200.9
64	16	121.1	100.9	267.9	212.4	118.3	98.7	272.2	217.0
125	25	241.2	197.3	328.4	249.2	212.1	176.5	344.9	265.5
216	36	339.7	278.2	446.9	340.8	341.6	280.2	472.6	361.7
		$x_s \in \mathbf{rand}(B)$				$x_s \in 2\,\mathbf{rand}(B)$			

Table 6.4
Probability of success when the constraint set $\mathcal{S}$ is defined by (6.6) *with* $r = 2$.

m	Multistart	Continuation	Multistart	Continuation
27	0%	100%	0%	90%
64	0%	90%	0%	60%
125	0%	60%	0%	60%
216	0%	30%	0%	30%
	$x_s \in \mathbf{rand}(B)$		$x_s \in 2\,\mathbf{rand}(B)$	

A difference between both definitions of $\mathcal{S}$ is that (6.6) includes all nearby atoms, while (6.2) includes some of the nearby atoms and some relatively far away atoms. Also note that if $r = 1$ then the set $\mathcal{S}$ defined by (6.6) has roughly $6m$ pairs; with $r = 2$ there are about $18m$ pairs, while with $r = 3$ there are about $26m$ pairs. On the other hand, the set $\mathcal{S}$ defined by (6.2) has about rm pairs. Thus, for the same value of r, the set $\mathcal{S}$ defined by (6.6) has more elements than (6.2).

We measured the reliability of the continuation and multistart methods on the second model problem by recording the percentage of successful runs in all 10 starting points. The results obtained were highly dependent on the choice of r. For example, the multistart and global continuation methods were 100% successful in all cases when $r = 1$, but as shown by Table 6.4, the multistart method fails in all cases when $r = 2$. The global continuation method always finds the solution, but the reliability is not as good as in Table 6.2.

The difficulty of the second model problem continues to vary as r increases. We give an impression of the variability in these results by briefly discussing the reliability of the multistart and global continuation methods. We restrict the discussion to the case where $x_s \in 2\mathbf{rand}(B)$, but similar results are obtained when $x_s \in \mathbf{rand}(B)$.

The reliability of the multistart method changes dramatically for $2 \le r \le 5$. Table 6.4 shows that the multistart method fails in all cases for $r = 2$. For $r = 3$, on the other hand, the multistart method only failed on half the cases, with most of the failures when $m = 216$ (four failures for $m = 27$, five for $m = 125$, and ten for $m = 216$). For $r = 4$, the multistart method only succeeds in two cases (when $m = 27$) and fails in all cases for $r = 5$. In general, the reliability of the multistart method decreases as the number of atoms increases.

The reliability of the global continuation method also changes for $2 \leq r \leq 5$, but the variability is not dramatic. Table 6.4 shows that the global continuation method failed in sixteen cases when $r = 2$. For $r = 3$ there were eight failures (two failures when $m = 124$ and six with $m = 216$), while for $r = 4$ there were two failures (one failure each for $m = 64$ and $m = 216$), but for $r = 5$ there were five failures (four failures for $m = 64$ and one failure for $m = 124$). The reliability of the global continuation method does not seem to depend on the number of atoms.

The results obtained with the second model problem show that the multistart method tends to fail, with the reliability decreasing as the number of atoms increase. The global continuation method, on the other hand, finds a solution for all problems with at least 30% reliability. Thus, our results with the second model problem confirm the general reliability conclusion obtained with the first problem.

Acknowledgment. Special thanks go to Paul Bash, Danny Ripoll, and Tamar Schlick for the time that they spent talking to us about problems in biology and the relevant literature. Thanks also go to Tom Coleman and Bruce Hendrickson for sharing their experiences with the distance geometry problem. Finally, Bruce Hendrickson and Lucjan Piela deserve special thanks for their careful reading of this manuscript.

REFERENCES

[1] A. T. Brünger and M. Nilges, *Computational challenges for macromolecular structure determination by X-ray crystallography and solution NMR-spectroscopy*, Q. Rev. Biophys., 26 (1993), pp. 49–125.

[2] T. F. Coleman, D. Shalloway, and Z. Wu, *Isotropic effective energy simulated annealing searches for low energy molecular cluster states*, Comput. Optim. Appl., 2 (1993), pp. 145–170.

[3] T. F. Coleman, D. Shalloway, and Z. Wu, *A parallel build-up algorithm for global energy minimizations of molecular clusters using effective energy simulated annealing*, J. Global Optim., 4 (1994), pp. 171–185.

[4] G. M. Crippen and T. F. Havel, *Distance Geometry and Molecular Conformation*, John Wiley & Sons, New York, 1988.

[5] C. Floudas and P. Pardalos, eds., *Recent Advances in Global Optimization*, Princeton University Press, Princeton, NJ, 1992.

[6] W. Gautschi, *Algorithm* 726 : ORTHOPOL – *A package of routines for generating orthogonal polynomials and Gauss-type quadrature rules*, ACM Trans. Math. Software, 20 (1994), pp. 21–62.

[7] W. Glunt, T. L. Hayden, and M. Raydan, *Molecular conformation from distance matrices*, J. Comp. Chem., 14 (1993), pp. 114–120.

[8] W. Glunt, T. L. Hayden, and M. Raydan, *Preconditioners for distance matrix algorithms*, J. Comp. Chem., 15 (1994), pp. 227–232.

[9] T. F. Havel, *An evaluation of computational strategies for use in the determination of protein structure from distance geometry constraints obtained by nuclear magnetic resonance*, Prog. Biophys. Mol. Biol., 56 (1991), pp. 43–78.

[10] B. A. Hendrickson, *The Molecule Problem: Determining Conformation from Pairwise Distances*, Ph.D. thesis, Cornell University, Ithaca, New York, 1991.

[11] B. A. Hendrickson, *The molecule problem: Exploiting structure in global optimization*, SIAM J. Optim., 5 (1995), pp. 835–857.

[12] R. Horst and H. Tuy, *Global Optimization*, Springer-Verlag, Berlin, New York, 1990.

[13] J. Kostrowicki and L. Piela, *Diffusion equation method of global minimization: Performance for standard functions*, J. Optim. Theory Appl., 69 (1991), pp. 269–284.

[14] J. Kostrowicki, L. Piela, B. J. Cherayil, and H. A. Scheraga, *Performance of the diffusion equation method in searches for optimum structures of clusters of Lennard–Jones atoms*, J. Phys. Chem., 95 (1991), pp. 4113–4119.

[15] J. Kostrowicki and H. A. Scheraga, *Application of the diffusion equation method for global optimization to oligopeptides*, J. Phys. Chem., 96 (1992), pp. 7442–7449.

[16] I. D. Kuntz, J. F. Thomason, and C. M. Oshiro, *Distance geometry*, in Methods in Enzymology, N. J. Oppenheimer and T. L. James, eds., vol. 177, Academic Press, New York, San Diego, 1993, pp. 159–204.

[17] S. Lang, *Real Analysis*, 2nd ed., Addison-Wesley, Reading, MA, 1983.

[18] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[19] L. Piela, J. Kostrowicki, and H. A. Scheraga, *The multiple-minima problem in the conformational analysis of molecules: Deformation of the protein energy hypersurface by the diffusion equation method*, J. Phys. Chem., 93 (1989), pp. 3339–3346.

[20] A. H. G. Rinnooy Kan and G. T. Timmer, *Global optimization*, in Optimization, G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, eds., North–Holland, Amsterdam, 1989, pp. 631–662.

[21] J. B. Saxe, *Embeddability of weighted graphs in k-space is strongly NP-hard*, in Proc. 17th Allerton Conference in Communications, Control, and Computing, Monticello, IL, 1979, pp. 480–489.

[22] H. A. Scheraga, *Predicting three-dimensional structures of oligopeptides*, in Reviews in Computational Chemistry, K. B. Lipkowitz and D. B. Boyd, eds., vol. 3, VCH Publishers, New York, 1992, pp. 73–142.

[23] D. Shalloway, *Application of the renormalization group to deterministic global minimization of molecular conformation energy functions*, J. Global Optim., 2 (1992), pp. 281–311.

[24] D. Shalloway, *Packet annealing: A deterministic method for global minimization, application to molecular conformation*, in Recent Advances in Global Optimization, C. Floudas and P. Pardalos, eds., Princeton University Press, Princeton, NJ, 1992, pp. 433–477.

[25] J. E. Straub, *Optimization techniques with applications to proteins*, in Recent Developments in Theoretical Studies of Proteins, R. Elber, ed., World Scientific, Singapore, 1996, pp. 137–196.

[26] Z. Wu, *The effective energy transformation scheme as a special continuation approach to global optimization with application to molecular conformation*, SIAM J. Optim., 6 (1996), pp. 748–768.

SIAM J. Optim.
Vol. 7, No. 3, pp. 837–852, August 1997

013

COMPUTATIONAL DESIGN OF OPTIMAL OUTPUT FEEDBACK CONTROLLERS[*]

T. RAUTERT[†] AND E. W. SACHS[‡]

Abstract. We consider the problem of designing feedback control laws when a complete set of state variables is not available. For linear autonomous systems with quadratic performance criterion, the design problem consists of choosing an appropriate matrix of feedback gains according to a certain objective function. In the literature, the performance of quasi-Newton methods has been reported to be substandard. We try to explain some of these observations and to propose structured quasi-Newton updates. These methods, which take into account the special structure of the problem, show considerable improvement in the convergence. Using test examples from optimal output feedback design, we also can verify these results numerically.

Key words. feedback control, quasi-Newton methods

AMS subject classifications. 49N35, 49N10, 65K10, 93D22, 93D52

PII. S1052623495290441

1. Introduction.

The computational design of feedback controllers has been an active research area of the control community for several decades. Since the resulting optimization problems are of considerable importance, there exist several special purpose algorithms developed by engineers to obtain a numerical solution. These can be found in books and review articles like [1] or [9]. By the mid 1980s, these algorithms had been refined using techniques from mathematical optimization such as step size rules or iterative solution of Newton steps. Also, there were attempts to use quasi-Newton methods, but these were reported to perform worse than the known algorithms designed by engineers in the field. It is the goal of this paper to follow up on this discrepancy and explain why this observation is reasonable. From this understanding we give a quasi-Newton method which is an extension of the classical engineering algorithms and therefore combines both favorable features.

In section 2 we derive and state the optimal design problem which we want to consider in this paper. The resulting optimization problem is to minimize

$$J(F) = tr\ [K(F)P],$$

where the variable F is a matrix. $K(F)$ is the solution of a matrix equation

$$K(F)\ [A+BFC]\ +\ [A+BFC]^T\ K(F) +\ C^T F^T RFC\ +\ Q\ =\ 0.$$

All other quantities are constant matrices and are defined and explained in section 2. A discussion of the output feedback problem and additional references appears in [2].

Section 3 deals with the derivation of the differentiability of the mappings involved and the proper formulation of the necessary optimality conditions. The first and second derivatives of the objective function are given in several lemmas.

For a review of the existing algorithms which treat this problem we refer to an excellent survey article (see [9]), where many references can be found. Section 4

[*] Received by the editors August 4, 1995; accepted for publication (in revised form) February 15, 1996.

http://www.siam.org/journals/siopt/7-3/29044.html

[†] Universität Trier, FB IV – Mathematik, D–54286 Trier, Germany.

[‡] Universität Trier, FB IV – Mathematik and "Graduiertenkolleg Mathematische Optimierung," D–54286 Trier, Germany (sachs@uni-trier.de).

presents the algorithm developed by Anderson and Moore in 1971 for the optimal design problem. For completeness we give the results on the descent property of this method and the convergence result using a step size rule. Similar results have already been credited in [9] to Halyo and Broussard as well as Mäkilä.

Section 5 contains convergence rate estimates for the Anderson–Moore algorithms in terms of the distance of the iterates to the solution. To our knowledge, only results on the convergence rate of the function values are known. We show linear convergence under a certain assumption on the Hessian of J. Furthermore, we can show that the convergence rate is quadratic for so-called observable systems; cf. [9].

In section 6 we give a structured quasi-Newton update based on the work of Dennis and Walker. We show that this type of update is an extension of the Anderson–Moore algorithm. If we do not update at all in our method, then the Anderson–Moore algorithm results. We prove that this method converges at a superlinear rate of convergence instead of the linear rate of the Anderson–Moore method.

Finally, in section 7 we use standard test examples from the engineering literature to support our theoretical results. It shows that the new method is superior to the Anderson–Moore method due to its improved local convergence properties.

2. Optimal design problem.

In this section we give a simple motivation for the optimal design problem. It can be found in most books on systems theory but might not be so well known in the area of mathematical optimization.

We consider the linear quadratic control system defined by

$$\begin{array}{rcl} \dot{x}(t) & = & Ax(t) + Bu(t)\,,\ \ x(0) = x_0, \\ y(t) & = & Cx(t), \end{array} \tag{1}$$

where

$x(t) \in \mathbb{R}^n$ state vector, $u(t) \in \mathbb{R}^p$ input vector, $y(t) \in \mathbb{R}^r$ output vector.

The objective function is given by

$$J_{x_0}(F) = \int_0^\infty \left[x(t)^T Q x(t) + u(t)^T R u(t) \right] dt. \tag{2}$$

With regard to the matrices, we make the following assumptions.

Assumption 1. Let $A, Q \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, $C \in \mathbb{R}^{r\times n}$, $R \in \mathbb{R}^{p\times p}$ be given with Q positive semidefinite, R positive definite, and C having maximal rank with $r \le n$.

It is our understanding that this assumption is true throughout the paper.

Our goal is to design a feedback control which is given by the time-invariant linear output feedback matrix F:

$$u(t) = Fy(t), \quad F \in \mathbb{R}^{p\times r}. \tag{3}$$

If we substitute (3) into the system equation (1), we obtain

$$\dot{x}(t) = (A + BFC)x(t)\,,\ \ x(0) = x_0. \tag{4}$$

In order to maintain stability of the system, F should be chosen from the set of stabilizing feedback gains

$$D_s = \{F \in \mathbb{R}^{p\times r} |\ Re\{\lambda(A + BFC)\} < 0\}\,, \tag{5}$$

where $\{\lambda(A+BFC)\}$ are the eigenvalues of $A+BFC$. Equation (3) can be used to rewrite the objective function (2) as

$$J_{x_0}(F) = \int_0^\infty x(t)^T[Q + (FC)^T R(FC)]x(t)\ dt. \tag{6}$$

In order to reformulate the objective function further, we recall the *Lyapunov equation.*

Theorem 2.1 (see [8, Theorem 3.2.1]). *Let $F \in D_s$ be given. Then there exists a unique solution $K(F) \in \mathbb{R}^{n\times n}$, $K(F)$ positive semidefinite, of*

$$K(F)\ [A+BFC]\ +\ [A+BFC]^T\ K(F) +\ C^T F^T RFC\ +\ Q\ =\ 0. \tag{7}$$

Furthermore, if $Q\ +\ C^T F^T RFC$ is positive definite then $K(F)$ is also positive definite.

It is easy to check that from (4) and (7),

$$\frac{d}{dt}x^T Kx = -x^T[Q + (FC)^T R(FC)]x.$$

This implies, with $F \in D_s$, that

$$\int_0^\infty x(t)^T[Q + (FC)^T R(FC)]x(t)\ dt = x^T(0)Kx(0)\ -\ \lim_{t\to\infty} x^T(t)Kx(t) = x_0^T Kx_0.$$

Theorem 2.2. *If $F \in D_s$, the objective function* (2) *can be rewritten as*

$$J_{x_0}(F) = x_0^T K(F)x_0, \tag{8}$$

where $K(F) \in \mathbb{R}^{n\times n}$ solves the Lyapunov equation (7).

Since the objective function depends on x_0, let x_0 be a random variable, uniformly distributed over the unit sphere. Then we arrive with a covariance matrix $P \in \mathbb{R}^{n\times n}$, P positive definite, at the optimal output feedback design problem.

$$\text{(D)}\quad \text{Minimize}\quad J(F) = E(J_{x_0}(F)) = tr\ [K(F)P]\quad \text{subject to}\quad F \in D_s\quad \text{and}\quad (7).$$

The function $K(F)$, defined by the Lyapunov equation (7), is continuous in F; hence, the objective function (8) also is continuous. In [10] the set of stabilizing feedback gains D_s (5) is replaced by the level set

$$N(F_0) = \{F \in D_s \mid J(F) \le J(F_0)\}.$$

It can be shown (see, e.g., [10]) that it is compact for any given matrix $F_0 \in D_s$. Using the theorem of Bolzano–Weierstrass, this implies the following theorem.

Theorem 2.3 (see [10, Lemma 2.1]). *Given a matrix $F_0 \in D_s$, the optimal output feedback design problem* (D) *has a solution in the level set $N(F_0)$.*

3. Derivatives of the objective function and optimality conditions. In the following theorem we state the most important properties of the objective function's first and second derivatives. They lead to a necessary optimality condition which is exploited in the development of algorithms to solve the optimal output feedback design problem.

First we show the differentiability of K depending on F.

LEMMA 3.1. *Let $F \in D_s$. Then $K(F)$ defined in* (7) *is differentiable and $K'(F)\ dF, dF \in \mathbb{R}^{p\times r}$ is given as the solution of the following Lyapunov equation:*

$$\begin{aligned} &K'(F)\ dF[A+BFC] + [A+BFC]^T K'(F)\ dF \\ &\qquad = -C^T dF^T[B^T K(F) + RFC] - [B^T K(F) + RFC]^T dFC. \end{aligned} \tag{9}$$

Proof. Define

$$\psi(K,F) = K\ [A+BFC]\ +\ [A+BFC]^T\ K +\ C^T F^T RFC\ +\ Q.$$

Since $F \in D_s$ and D_s is an open set, we know by Theorem 2.1 that $\psi_K(K,F)$ given by

$$\psi_K(K,F)dK = dK\ [A+BFC]\ +\ [A+BFC]^T\ dK$$

is surjective. The implicit function theorem then yields the differentiability of $K(F)$ and also, with

$$\psi_F(K,F)dF = KBdFC + [BdFC]^T K + C^T dF^T RFC + C^T F^T RdFC,$$

the representation of $K'(F)\ dF$. □

In order to write the derivative $J'(F)\, dF$ for $dF \in \mathbb{R}^{p\times r}$ in a gradient form, we introduce a gradient of J using the trace of a matrix:

$$J'(F)\, dF = tr[\nabla J(F)^T dF], \quad \text{where} \quad \nabla J(F) \in \mathbb{R}^{p\times r}\,.$$

THEOREM 3.2. *The objective function* (2) *is differentiable on the set of stabilizing feedback gains D_s. The gradient of J at F is given by*

$$\nabla J(F) = 2[B^T K(F) + RFC]L(F)C^T,$$

where $L(F) \in \mathbb{R}^{n\times n}$ solves the Lyapunov equation

$$L(F)[A+BFC]^T + [A+BFC]L(F) + P = 0\,. \tag{10}$$

Proof Using Lemma 3.1 we have

$$J'(F)\ dF = tr[K'(F)\ dF\ P],$$

where $K'(F)\ dF$ solves the Lyapunov equation (9). Since $L(F)$ also solves a Lyapunov equation, we can proceed as follows. Multiply (10) by $K'(F)\ dF$ and (9) by $L(F)$ and then take the trace of the left-and right-hand side of both equations. This yields the following identity and completes the proof:

$$tr\ [K'(F)\ dF\ P] = tr\ [2(B^T K(F) + RFC)L(F)C^T dF]. \qquad \square$$

The result of Theorem 3.2 also yields the necessary optimality condition for the design problem (D).

THEOREM 3.3. *If $F_* \in D_s$ solves the design problem* (D), *then*

$$[B^T K(F_*) + RF_* C]L(F_*)C^T = 0\,, \tag{11}$$

where $K(F_*)$ and $L(F_*)$ are solutions of (7) *and* (10), *respectively.*

In case of observability, an interesting connection can be drawn between the design problem (D) and the Riccati equation.

ASSUMPTION 2. For the dimension of the matrices, let $r = n$ and let C^{-1} exist.

Then we can show the following corollary.

COROLLARY 3.4. *If the system* (1) *is observable, i.e., Assumption* 2 *holds, then* (7)*,* (10)*, and* (11) *are equivalent to solve the Riccati equation*

$$K_*A + AK_*^T - K_*BR^{-1}B^TK_* + Q = 0$$

and to set

$$F_* = -R^{-1}B^TK_*C^{-1}.$$

Proof. If C^{-1} exists, then (11) reduces to

$$F_*C = -R^{-1}B^TK_*. \tag{12}$$

Hence, (10) is redundant. If we substitute (12) into (7), then we obtain

$$\begin{aligned} 0 &= K_*(A - BR^{-1}B^TK_*) + (A - BR^{-1}B^TK_*)^TK_* + K_*BR^{-1}B^TK_* + Q \\ &= K_*A + AK_*^T - K_*BR^{-1}B^TK_* + Q\,. \qquad \square \end{aligned}$$

For higher order methods, second derivative information is an essential part in the development and analysis of algorithms.

THEOREM 3.5. *The objective function* (2) *is twice differentiable, and the second derivative is Lipschitz continuous on the set of stabilizing feedback gains D_s. The second derivative, applied to a direction $dF \in \mathbb{R}^{p\times r}$, is given by*

$$\begin{aligned} J''(F)(dF, dF) &= 2\,tr\left[dF^TRdFCL(F)C^T\right] + 2\,tr\left[dF^TB^TK'(F)\ dFL(F)C^T\right] \\ &\quad + 2\,tr\left[dF^T(B^TK(F) + RFC)L'(F)\ dFC^T\right] \\ &= 2\,tr\left[dF^TRdFCL(F)C^T\right] + 4\,tr\left[dF^TB^TK'(F)\ dFL(F)C^T\right]\,, \end{aligned} \tag{13}$$

where $K'(F)\ dF = K'(F)dF \in \mathbb{R}^{n\times n}$ and $L'(F)\ dF = L'(F)dF \in \mathbb{R}^{n\times n}$ solve the Lyapunov equations

$$\begin{aligned} &K'(F)\ dF[A + BFC] + [A + BFC]^TK'(F)\ dF \\ &\qquad = -C^TdF^T[B^TK(F) + RFC] - [B^TK(F) + RFC]^TdFC \end{aligned} \tag{14}$$

and

$$\begin{aligned} &L'(F)\ dF[A + BFC]^T + [A + BFC]L'(F)\ dF \\ &\qquad = -BdFCL(F) - L(F)C^TdF^TB^T. \end{aligned} \tag{15}$$

Proof With the same arguments as in Lemma 3.1, one can show that $L(F)$ is differentiable. The derivative $L'(F)\ dF$ satisfies (15). Then we obtain from Theorem 3.2 that

$$\begin{aligned} J''(F)(dF, dF) = 2\ tr\quad &[(B^TK'(F)\ dF + RdFC)L(F)C^TdF \\ &+ (B^TK(F) + RFC)L'(F)\ dFC^TdF]. \qquad \square \end{aligned}$$

Given an increment $dF \in \mathbb{R}^{p\times r}$, we can approximate the objective function J at a point F by the quadratic model

$$\begin{aligned} J(F+dF)-J(F) &= J'(F)\,dF + \frac{1}{2}\,J''(F)(dF,dF) \\ &= tr\left[dF^T\nabla J(F)\right] + q_1(F,dF) + q_2(F,dF)\ , \end{aligned} \tag{16}$$

where the quadratic terms are split into

$$\begin{aligned} q_1(F,dF) &= tr\left[dF^T R dF C L(F) C^T\right] \quad \text{and} \\ q_2(F,dF) &= tr\left[dF^T B^T K'(F)(dF) L(F) C^T\right] \\ &\quad + tr\left[dF^T(B^T K(F) + RFC) L'(F)(dF) C^T\right]\ . \end{aligned}$$

The term q_1 is always positive for nonzero dF as the following lemma shows.

LEMMA 3.6. *Let $F \in D_s$. Then there exists $\mu > 0$ such that we have, for all $dF \in \mathbb{R}^{p\times r}$,*

$$tr\left[dF^T R dF C L(F) C^T\right] \ge \mu \|dF\|^2.$$

Proof. We note that $F \in D_s$ implies by Theorem 2.1 that $L(F)$ is positive definite. By Assumption 1 the matrix $V := CL(F)C^T$ is also positive definite. The matrix $U := dF^T R dF$ is positive semidefinite. By Theorem 7.4.10 in [7] there exists for positive semidefinite matrices $U, V \in \mathbb{R}^{r\times r}$ a permutation π of the indices $1, ..., r$ such that

$$tr\ [UV] = \sum_{i=1}^{r} \lambda_i \mu_{\pi(i)},$$

where $\lambda_i, \mu_i \ge 0$ are the eigenvalues of U, V, respectively. This implies

$$tr\ [UV] = tr\ [(dF^T R dF)(CL(F)C^T)] > 0$$

unless all eigenvalues of $dF^T R dF$ vanish. In this case we obtain $R^{1/2}dF = 0$ and, hence, $dF = 0$. □

Dealing with matrices $F \in \mathbb{R}^{p\times r}$ has the disadvantage that one cannot express $\nabla^2 J(F)$ in an explicit form; see Theorem 3.5. If we write F as a column vector $vec[F] \in \mathbb{R}^{p\cdot r}$ and define a function $j : \mathbb{R}^{p\cdot r} \longrightarrow \mathbb{R}$ such that $j(vec[F]) = J(F)$, we are able to calculate the Hessian of j using the Kronecker product $\otimes$.

LEMMA 3.7. *The Hessian of the function $j(vec[F])$ described above is given by*

$$H(F) = 2 \cdot \left\{\left(CL(F)C^T \otimes R\right) + H_1(F)^T + H_1(F)\right\},$$

where $H_1(F)$ can be expressed with an $n^2 \times n^2$-permutation-matrix $P(n,n)$ as

$$\begin{aligned} H_1(F) = &\left[C \otimes \left(B^T K(F) + RFC\right)\right] \left([I \otimes [A+BFC]] + [[A+BFC] \otimes I]\right)^{-1} \\ &(I_{n\cdot n} + P(n,n)) \left[L(F)C^T \otimes B\right]. \end{aligned} \tag{17}$$

According to Lemma 3.6, the matrix $CL(F)C^T \otimes R$ is positive definite. The remaining term of the Hessian $H_1(F) + H_1(F)^T$ may be indefinite, which leads to the following conclusion.

COROLLARY 3.8. *If $\|H_1(F)\|$ is sufficiently small, then $H(F)$ is positive definite.*

4. Anderson–Moore algorithm. One of the algorithms that has been used successfully in the past is the Anderson–Moore method [1]. We rewrite the necessary optimality conditions (11) for the design problem (D) as

$$F = -R^{-1}[B^T K(F)L(F)C^T][CL(F)C^T]^{-1}.$$

Based on this equation, we can formulate a fixed point iteration (the so-called Anderson–Moore algorithm).

Anderson–Moore Algorithm (1971).

Given F

- Solve (7) for $K(F)$
- Solve (10) for $L(F)$
- Set $$\begin{aligned} F_+ &= -R^{-1}[B^T K(F)L(F)C^T][CL(F)C^T]^{-1} \\ &= F - \frac{1}{2}R^{-1}\nabla J(F)[CL(F)C^T]^{-1} \end{aligned}$$

We refine the algorithm and enhance its global convergence properties by the incorporation of the following step size rule:

$$\begin{aligned} F_+ &= F - \alpha(F + R^{-1}[B^T K(F)L(F)C^T][CL(F)C^T]^{-1}) \\ &= F - \frac{\alpha}{2}R^{-1}\nabla J(F)[CL(F)C^T]^{-1}. \end{aligned} \tag{18}$$

The most appealing aspect of the Anderson–Moore algorithm is the fact that each step is very simple to compute once the gradient of the objective function has been evaluated. It is evident from (18) that the Anderson–Moore algorithm is different from the gradient method. Nevertheless, it is a descent method which is well known. We give a refined version of this statement which will be used later.

Lemma 4.1. *The direction for the Anderson–Moore algorithm*

$$-F - R^{-1}[B^T K(F)L(F)C^T][CL(F)C^T]^{-1} = -\frac{1}{2}R^{-1}\nabla J(F)[CL(F)C^T]^{-1}$$

is a descent direction for $J(F)$ at $F \in D_s$ unless $\nabla J(F) = 0$. In particular, there exists $\nu > 0$ such that

$$tr\ [\nabla J(F)^T (R^{-1}\nabla J(F)[CL(F)C^T]^{-1}) \geq \nu\ \|\nabla J(F)\|^2. \tag{19}$$

Proof. A matrix $S \in \mathbb{R}^{p\times r}$ is a descent direction if $tr\ [\nabla J(F)^T S] < 0$. In this special case we have

$$tr\ [\nabla J(F)^T S] = -\frac{1}{2} tr\ [\nabla J(F)^T R^{-1}\nabla J(F)[CL(F)C^T]^{-1}].$$

The proof of (19) follows along the same arguments as in the proof of Lemma 3.6. □

Lemma 4.1 enables us to use classical convergence results from optimization for descent methods to derive convergence. In [11], the Anderson–Moore step was combined with an Armijo step size rule, and a proof of convergence was given in [9].

Theorem 4.2. *Consider a sequence of iterates $\{F_k\} \subset D_s$ as in* (18)*, where the step size is determined by an Armijo rule. Then we obtain*

$$\lim_{k\to\infty} \|\nabla J(F_k)\| = 0.$$

Proof. Since J is bounded from below and is differentiable with a Lipschitz-continuous derivative, we can invoke standard convergence theorems for descent methods with descent directions S_k; see, e.g., [4] to obtain

$$\lim_{k\to\infty} \frac{tr\ [\nabla J(F_k)^T S_k]}{\|\nabla J(F_k)\|} = 0.$$

Using (19) in Lemma 4.1, we can derive the desired statement of the theorem. □

Next we prove estimates on the rate of convergence.

5. Rate of convergence for the Anderson–Moore method. If we look at the statement in Corollary 3.8, we expect a linear rate of convergence if the term $H_1(F_*)$ is sufficiently small in norm. As we will show below, this rate estimate can be refined substantially, and the quadratic rate behavior which is observed in some instances can be explained along the same lines also.

We look at an alternative approach to derive the Anderson–Moore algorithm. Instead of using all second order terms of J, we use the following quadratic model for the objective function:

$$J(F + dF) \quad \approx \quad q_{AM}(F, dF) = J(F) + tr[dF^T \nabla J(F)] + q_1(F, dF)\ .$$

This approximation uses only the first quadratic term q_1 of the Hessian (16), while the other quadratic terms are neglected. The relation to the Anderson–Moore algorithm is clear from the following lemma.

LEMMA 5.1. *For given $F \in D_s$, the minimization problem*

$$\min_{dF} q_{AM}(F, dF) \tag{20}$$

has the unique solution

$$G_{AM} = -\frac{1}{2} R^{-1}(\nabla J(F))[CL(F)C^T]^{-1}\ . \tag{21}$$

Proof. The necessary optimality condition for (20) is

$$\nabla J(F) + 2RdFCL(F)C^T = 0,$$

and since R and $CL(F)C^T$ are positive definite, this yields the expression (21) and its uniqueness. □

The step computed by G_{AM} in (21) is the same as the Anderson–Moore step in (18). The fact that certain parts of the Hessian are neglected in the Anderson–Moore method and that q_1 is positive definite allows us to derive convergence rate results similar to the Gauß–Newton method (cf. [4]).

THEOREM 5.2. *Let $\{F_k\}$ be the sequence of feedback gains determined by the Anderson–Moore algorithm.*

(i) *Let $\delta, \lambda_1, \lambda_2 > 0$, with λ_1 the smallest eigenvalue of R, and let λ_2 be the minimum of the smallest eigenvalues of $CL(F)C^T$ for F in a ball $B_\delta(F_*)$. If*

$$\begin{aligned} &\|B^T K'(F_k)(F_* - F_k)L(F_k)C^T + \left[B^T K(F_k) + RF_kC\right] L'(F_k)(F_* - F_k)C^T\| \\ &\qquad \le \sigma\|F_* - F_k\| \end{aligned} \tag{22}$$

with $0 < \sigma < \lambda_1\lambda_2$ *for* $F_k \in B_\epsilon(F_*)$, *then there exists* $\epsilon > 0$ *such that*

$$\|F_{k+1} - F_*\| \leq \frac{1}{2\lambda_1\lambda_2}\left(\frac{L_{\nabla J}}{2}\epsilon + \sigma\right)\|F_k - F_*\| < \|F_k - F_*\| \ ,$$

i.e., the Anderson–Moore method has a linear rate of convergence in $B_\epsilon(F_*)$.

(ii) *If*

$$q_2(F_*, dF) = 0 \quad \forall\ dF \in \mathbb{R}^{p\times r} \ , \tag{23}$$

then there are $c, \delta > 0$ *such that the Anderson–Moore algorithm converges q-quadratically in* $B_\delta(F_*)$; *i.e.,*

$$\|F_{k+1} - F_*\| \leq c\|F_k - F_*\|^2 \quad \textit{for} \quad F_k \in B_\delta(F_*) \ .$$

Proof. We first show Theorem 5.2 (ii), then (i).

Choose $\delta > 0$ such that $\nabla J(F), \nabla^2 J(F), K(F), L(F), K'(F)$, and $L'(F)$ are Lipschitz continuous in a ball $B_\delta(F_*)$. If $F_k \in B_\delta(F_*)$, the following holds (using (23)):

$$\begin{aligned}
&\|F_{k+1} - F_*\| = \|F_k - \tfrac{1}{2}R^{-1}(\nabla J(F_k))\left[CL(F_k)C^T\right]^{-1} - F_*\| \\
= \ & \|R^{-1}\left\{R(F_k - F_*)\left[CL(F_k)C^T\right] + \tfrac{1}{2}(\nabla J(F_*) - \nabla J(F_k))\right\}\left[CL(F_k)C^T\right]^{-1}\| \\
\leq \ & \tfrac{1}{2}\cdot\|R^{-1}\|\ \|\left[CL(F_k)C^T\right]^{-1}\|\ \|\nabla J(F_*) - \nabla J(F_k) - 2\cdot\left[R(F_* - F_k)CL(F_k)C^T\right]\| \\
\leq \ & \tfrac{1}{2}\cdot\|R^{-1}\|\ \|\left[CL(F_k)C^T\right]^{-1}\|\ \|\nabla J(F_*) - \nabla J(F_k) - \nabla^2 J(F_k)(F_* - F_k)\| \\
& +\|B^TK'(F_k)(F_* - F_k)L(F_k)C^T + \left[B^TK(F_k) + RF_kC\right]L'(F_k)(F_* - F_k)C^T \\
& -(B^TK'(F_*)(F_* - F_k)L(F_*)C^T + \left[B^TK(F_*) + RF_*C\right]L'(F_*)(F_* - F_k)C^T)\|.
\end{aligned} \tag{24}$$

Since $\nabla^2 J$ is Lipschitz continuous in $B_\delta(F_*)$, there is $L_{\nabla J} > 0$ such that the first term of the sum can be bounded by a quadratic term in $\|F_* - F_k\|$. Since K', L', K, and L are Lipschitz continuous, there is $L > 0$ such that

$$\|F_{k+1} - F_*\| \leq \frac{1}{2}\|R^{-1}\|\ \|\left[CL(F_k)C^T\right]^{-1}\|\left(\frac{L_{\nabla J}}{2} + L\right)\|F_k - F_*\|^2.$$

Let λ_1 be the smallest eigenvalue of R and let λ_2 be the smallest eigenvalues of $CL(F)C^T$. This yields the q-quadratic convergence of the Anderson–Moore method:

$$\|F_{k+1} - F_*\| \leq \frac{1}{2\lambda_1\lambda_2}\left(\frac{L_{\nabla J}}{2} + L\right)\|F_k - F_*\|^2 = c\|F_k - F_*\|^2 \ .$$

To show (i), note that the last term in (24) can be estimated using condition (22) by $\sigma\|F_* - F_k\|$. Inserted into equation (24), this yields

$$\begin{aligned}
\|F_{k+1} - F_*\| &\leq \frac{1}{2}\cdot\|R^{-1}\|\ \|\left[CL(F_k)C^T\right]^{-1}\|\left(\frac{L_{\nabla J}}{2}\|F_k - F_*\|^2 + \sigma\|F_k - F_*\|\right) \\
&\leq \frac{1}{2\lambda_1\lambda_2}\left(\frac{L_{\nabla J}}{2}\|F_k - F_*\|^2 + \sigma\|F_k - F_*\|\right).
\end{aligned}$$

Let $\epsilon < \min\{\delta, (4\lambda_1\lambda_2 - 2\sigma)/L_{\nabla J}\}$, and we then obtain the desired result:

$$\|F_{k+1} - F_*\| \leq \|F_k - F_*\| \frac{1}{2\lambda_1\lambda_2} \left(\frac{L_{\nabla J}}{2}\epsilon + \sigma \right) < \|F_k - F_*\| \; . \qquad \square$$

In particular, when the system (1) is observable (a case in which solving the design problem is equivalent to solving the Riccati equation as we showed in Corollary 3.4), the Anderson–Moore algorithm converges quadratically (cf. [9], p. 660).

COROLLARY 5.3. *Let the system* (1) *be observable, i.e., Assumption* 2 *holds. Then the Anderson–Moore algorithm converges q-quadratically in the neighborhood of a solution of the design problem* (D)*.*

Proof. In the case of observability, the necessary optimality condition for the design problem reduces to $B^T K(F_*) + RF_*C = 0$. If we insert this into (14), we obtain $K'(F_*)(dF) = 0$, and thus $q_2(F_*, dF) = 0$. Hence, the condition (23) in Theorem 5.2 is fulfilled and, therefore, the algorithm has a quadratic rate of convergence. $\square$

It is also possible to rewrite the design problem in form of a nonlinear least-squares problem.

LEMMA 5.4. *The design problem* (D) *can be expressed as a nonlinear least-squares problem by*

$$J(F) = \|K(F)^{\frac{1}{2}} P^{\frac{1}{2}}\|_F^2 \tag{25}$$

with the Frobenius norm $\|\cdot\|_F$.

Proof

$$J(F) = tr\,[K(F)P] = tr\,[K(F)^{\frac{T}{2}} K(F)^{\frac{1}{2}} P^{\frac{1}{2}} P^{\frac{T}{2}}] = \|K(F)^{\frac{1}{2}} P^{\frac{1}{2}}\|_F^2. \qquad \square$$

Since it is well known that the Gauß–Newton method converges at a quadratic rate in the zero residual case, one might suspect that the Anderson–Moore algorithm is identical with the Gauß–Newton method. However, using the formulation (25), there is no equivalence between Gauß–Newton method for the nonlinear least-squares problem and the Anderson–Moore algorithm. The derivative $R'(F)$ of the residual $R(F) = K(F)^{\frac{1}{2}} P^{\frac{1}{2}}$ in (25) is given by

$$R'(F)dF = \frac{1}{2} K(F)^{-\frac{1}{2}} K'(F) dF P^{\frac{1}{2}}.$$

In the Gauß–Newton approach, we would use as a quadratic approximation for $J(F)$

$$(R'(F)dF)^T R'(F)dF = \frac{1}{4} tr\ [dF^T K'(F) K(F)^{-1} K'(F) dF P].$$

For example, in the case of observability, this quadratic term vanishes at F_* and, hence, no quadratic rate of convergence could be shown. Furthermore, the residual in (25) is never zero, so the quadratic convergence for the Gauß–Newton method in the zero residual case cannot be applied.

In this framework, we also want to mention Newton's method, which minimizes a quadratic approximation of the objective function J

$$\min q_N(dF) = J(F) + \nabla J(F)(dF) + \frac{1}{2} J''(F)(dF, dF) \, .$$

The necessary optimality condition for this minimization problem is

$$\begin{aligned} &[B^T K(F) + RFC] L(F) C^T + R\ dFCL(F)C^T \\ &+ B^T K'(F)(dF) L(F) C^T + [B^T K(F) + RFC] L'(F)(dF) C^T = 0, \end{aligned} \tag{26}$$

where $K'(F)dF$ and $L'(F)dF$ are given by the Lyapunov equations (15) and (14). This coupled system of three equations adds substantially to the computational cost compared to the Anderson–Moore algorithm. Using these equations, we can formulate Newton's algorithm.

Newton's Method.

Given F

- Solve (7) for $K(F)$
- Solve (10) for $L(F)$
- Solve (26) simultaneously with (14) and (15) for dF
- Set $F_+ = F + \alpha\, dF$

It has been suggested that (26) can be solved with an iterative scheme like the conjugate gradient method (or similar scheme), but we still have to solve two Lyapunov equations for $K'(F)dF$ and $L'(F)dF$ in each cg-step, which makes the computation of a Newton-step rather expensive.

In optimization, quasi-Newton methods are often used instead, but these methods have been reported to exhibit substandard performance. We will show in the next section how to develop successful applications of quasi-Newton methods for this class of problems.

6. Structured quasi-Newton method.

Although there are special cases where the Anderson–Moore method converges quite fast, in general, its rate of convergence is linear and sometimes rather slow. On the other hand, it is globally a rather robust optimization algorithm where the descent direction is especially tuned to the application problem.

General purpose quasi-Newton methods such as BFGS or DFP do not perform as well as usual; see, e.g., [11]. Therefore, we want to develop a quasi-Newton technique which is particularly suited for this application.

The idea is based on structured quasi-Newton updates. Certain parts of the Hessian are relatively easy to compute (such as q_1 in (16)), while others are not (such as q_2). Therefore, we distinguish in the Hessian between a part that is exactly available, denoted by $D(F)$, and a part that has to be approximated by a quasi-Newton update A. We split the Hessian $J''(F) = D(F) + A(F)$ in such a way that for $A(F) = 0$, the algorithm reduces to the Anderson–Moore method. In this way the resulting algorithm can be viewed as an extension of the Anderson–Moore method.

For a description of the method, let the matrices $\nabla J(F), S_c, Y_c^{\#}$ be given and let the approximation of the second derivative be

$$J''(F) \approx A + D(F), \quad A, D(F) : \mathbb{R}^{p\times r} \to \mathbb{R}^{p\times r},$$

where $D(F)S = RS\left[CL(F)C^T\right],\ S \in \mathbb{R}^{p\times r}$.

Structured Quasi-Newton Updates.

- For given F_c, A_c solve

$$(A_c + D(F_c))S_c = -\nabla J(F_c).$$

- Set $F_+ = F_c + S_c$ and

$$Y_c^{\#} = \nabla J(F_+) - \nabla J(F_c) - D(F_+)S_c \tag{27}$$

$$\text{or} \quad Y_c^{\#} = \left[B^T\left(K(F_+) - K(F_c)\right)L(F_+)C^T \right.$$
$$\left. +(B^TK(F_+) + RF_+C)\left(L(F_+) - L(F_c)\right)C^T\right]. \tag{28}$$

- Update A by defining A_+X for all $X \in \mathbb{R}^{p\times r}$

$$A_+X = A_cX + \frac{tr\ [(Y_c^\# - A_cS_c)^T X]}{tr\ [S_c^T(Y_c^\# - A_cS_c)]}(Y_c^\# - A_cS_c).$$

This update is the SR-1-update, but the PSB-update

$$\begin{aligned} A_+X = A_cX &+ \frac{tr\ [S_c^T X]}{tr\ [S_c^T S_c]}(Y_c^\# - A_cS_c) \\ &+ \left(\frac{tr\ [(Y_c^\# - A_cS_c)^T X]}{tr\ [S_c^T S_c]} - \frac{tr\ [(Y_c^\# - A_cS_c)^T S_c]}{(tr\ [S_c^T S_c])^2}\right) S_c \end{aligned} \tag{29}$$

also has been used in our numerical experiments.

The choice of $Y_c^\#$ in (27) corresponds to what Dennis and Walker [5] call the "default choice." The version (28), which has been slightly more successful in our tests, approximates $K'(F_+)(S_c)$ and $L'(F_+)(S_c)$ in q_2 by $K(F_+)-K(F_c)$ and $L(F_+)-L(F_c)$, respectively.

Based on the results of Dennis and Walker [5], we can now show that the algorithm described above has a local q-superlinear rate of convergence.

THEOREM 6.1. *Let F_* be the solution of the design problem* (D)*, let the Hessian $J''(F_*)$ be nonsingular, and choose $Y_k^\#$ as in* (27) *or* (28)*. Then there are $\epsilon > 0$ and $\delta > 0$ such that for $\|F_0 - F_*\| \le \epsilon$ and $\|A_0 - (H_1(F_*) + H_1(F_*)^T)\| \le \delta$ (cf.* (17)*), the sequence $\{F_k\}$ generated by the structured quasi-Newton method described above using either the Broyden-, or the PSB-update is well defined and converges q-superlinearily.*

Proof. This proof is based on Theorem 11.4.1 in [4] (respectively, Theorem 3.3 in [5]). We restrict ourselves to show that the conditions of this theorem are fulfilled. For the Broyden-update ($\mathcal{A} = \mathbb{R}^{n\times n}$), as well as for the PSB-update ($\mathcal{A} = \{M \in \mathbb{R}^{n\times n} : M = M^T\}$), we have (cf. Lemma 3.7)

$$H(F_*) - D(F_*) = H_1(F_*) + H_1(F_*)^T \in \mathcal{A}\ ,$$

since this matrix is always symmetric. Hence, condition (11.4.2) in Theorem 11.4.1 in [4] holds.

If $Y_c^\#$ has been chosen as in (27), the assertion follows from Theorem 11.4.1 in [4]. If $Y_c^\#$ has been chosen as in (28), we have to show that there are $\alpha > 0$ and $p > 0$ such that condition (11.4.3) in Theorem 11.4.1 in [4] is fulfilled. For the Broyden- and the PSB-update, it is sufficient to show that we can choose $\alpha > 0$ and $p > 0$ in a way that for all F_c, F_+ in a neighborhood of F_*,

$$\|Y_c^\# - A_*S_c\| \le \alpha\,(\max\{\|F_c - F_*\|, \|F_+ - F_*\|\})^p\,\|S_c\| \tag{30}$$

(cf. [5], p. 963 f., Lemma 3.2). We have from (28) and $A_* = H_1(F_*) + H_1(F_*)^T$ that

$$\begin{aligned} \|Y_c^\# - A_*S_c\| &= \|B^T\,(K(F_+) - K(F_k))\,L(F_+)C^T - B^TK'(F_*)(F_+ - F_c)L_*C^T \\ &\quad + (B^TK(F_+) + RF_+C)\,(L(F_+) - L(F_k))\,C^T \\ &\quad - (B^TK_* + RF_*C)L'(F_*)(F_+ - F_c)C^T\| \\ &\le \|B^T\,\{K(F_+) - K(F_c) - K'(F_*)(F_+ - F_c)\}\,L_*C^T\| \\ &\quad + \|B^T(K(F_+) - K(F_c))(L(F_+) - L_*)C^T\| \\ &\quad + \|[B^TK_* + RF_*C]\,\{L(F_+) - L(F_c) - L'(F_*)(F_+ - F_c)\}\,C^T\| \\ &\quad + \|\left([B^TK(F_+) + RF_+C] - [B^TK_* + RF_*C]\right)(L(F_+) - L(F_c))\|\ . \end{aligned}$$

From this equation it is easy to derive the desired estimate. □

7. Numerical results. We tested the algorithm on examples which are used in the literature for test purposes. Consider the fourth-order system (cf. [3]) $n = 4, p = 2, r = 3$, and

$$A = \begin{pmatrix} -0.03700 & 0.01230 & 0.00055 & -1.00000 \\ 0.00000 & 0.00000 & 1.00000 & 0.00000 \\ -6.37000 & 0.00000 & -0.23000 & 0.06180 \\ 1.25000 & 0.00000 & 0.01600 & -0.04570 \end{pmatrix},$$

$$B = \begin{pmatrix} 0.000840 & 0.000236 \\ 0.000000 & 0.000000 \\ 0.080000 & 0.804000 \\ -0.086200 & -0.066500 \end{pmatrix},$$

$$C = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad F_* = \begin{pmatrix} 0.398 & 1.593 & 7.852 \\ -1.258 & -3.482 & -5.004 \end{pmatrix},$$

$$R = I, \quad P = Q = I.$$

The Lyapunov equations are solved numerically using the Bartels–Stewart algorithm. Starting data are $F_0 = 0$, which is a stabilizing feedback gain. We list the norm of the gradients for four methods in Table 1. In the quasi-Newtonm method, we started the update procedure after six Anderson–Moore steps. We include the same strategy also for Newton's method in the column denoted by A–M/Newton.

TABLE 1
Comparison of four methods.

k	Anderson–Moore	Newton	Struct. quasi-Newton	A–M/Newton
1	0.8503D+07	0.8503D+07	0.8503D+07	0.8503D+07
2	0.2869D+04	0.3779D+07	0.2870D+04	0.2870D+04
3	0.4515D+02	0.1679D+07	0.4515D+02	0.4515D+02
4	0.3810D+02	0.7463D+06	0.3810D+02	0.3810D+02
5	0.1955D+02	0.3316D+06	0.1955D+02	0.1955D+02
6	0.4724D+01	0.1472D+06	0.4724D+01	0.4724D+01
7	0.3235D+01	0.6535D+05	0.3235D+01	0.3235D+01
8	0.2237D+01	0.2901D+05	0.6874D+00	0.1571D+01
9	0.1579D+01	0.1221D+05	0.7435D+00	0.6805D+00
10	0.1133D+01	0.5151D+04	0.2594D+00	0.1843D+00
11	0.8250D+00	0.2185D+04	0.5243D–01	0.2893D–01
12	0.6070D+00	0.9343D+03	0.2168D–01	0.9271D–04
13	0.4504D+00	0.4025D+03	0.9107D–03	0.1099D–08
14	0.3365D+00	0.1846D+03	0.2465D–04	
15	0.2527D+00	0.8375D+02	0.2642D–05	
16	0.1905D+00	0.3663D+02	0.2077D–07	
17	0.1442D+00	0.1499D+02		
18	0.1092D+00	0.5269D+01		
19	0.8304D–01	0.1268D+01		
20	0.6320D–01	0.1329D+00		
21	0.4815D–01	0.1691D–02		
22	0.3672D–01	0.3116D–06		
·	·			
60	0.1428D–05			

We observe clearly the linear convergence of the Anderson–Moore method and the superlinear convergence of the structured quasi-Newton algorithm. In spite of his local quadratic rate of convergence, Newton's method needs more steps to achieve the termination criterion $\|\nabla J(F)\| < 10^{-5}$ than the structured quasi-Newton method. In terms of the CPU-time (on a DEC 3000 workstation) that is needed until the termination criterion is reached, the superiority of the structured quasi-Newton method is even more striking:

- 0.145 seconds for the structured quasi-Newton method,
- 0.263 seconds for Newton's method,
- 0.387 seconds for the Anderson–Moore method.

If we change the previous example by setting $C = I_4$, we obtain an observable system. Using the stabilizing feedback gain

$$F_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

to start, we obtain the solution

$$F_* = \begin{bmatrix} -1.862 & 0.180 & 0.701 & 6.407 \\ 3.939 & -0.928 & -1.554 & -2.993 \end{bmatrix}.$$

Table 2 lists the result of the Anderson–Moore method and shows a quadratic rate of convergence according to Corollary 5.3.

Table 2
Quadratic convergence of Anderson–Moore.

step	$J(F)$	$\|\nabla J(F)\|$
1	0.3113D+05	0.8503D+07
2	0.7583D+03	0.2810D+04
3	0.3973D+03	0.2075D+02
4	0.2202D+03	0.1591D+02
5	0.1611D+03	0.1326D+02
6	0.1515D+03	0.3427D+01
7	0.1512D+03	0.1220D+00
8	0.1512D+03	0.1650D–03
9	0.1512D+03	0.3184D–09

To illustrate once more the advantages of structured quasi-Newton methods, consider the following example from Horisberger and Bélanger [6]:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -20 & -4.2 & 0 & 4.45 & 12.5 & 0 & 100 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4.7 & 8.35 & 0 & -1.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -3.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 10.9 & 0 & 0 & -2.55 & -250 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 5.9 & 0 & 0 & -1.39 & 0 & 0 & -3700 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 3.3 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0.66 & 0 & 1.2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0.66 & 0 & 1.2 \end{bmatrix}, P = I_9, Q = I_9, R = I_1.$$

Starting with

$$F_0 = \left[\begin{array}{cccc} 10.62 & 3.63 & 372.7 & 37.82 \end{array} \right],$$

we obtained

$$F_* = \left[\begin{array}{cccc} 0.0309 & 2.6422 & 63.7458 & 2.1004 \end{array} \right]$$

and the following iterations for Newton's method, the Anderson–Moore algorithm, and structured quasi-Newton methods (see Table 3).

TABLE 3
Comparison with structured quasi-Newton.

k	Anderson–Moore	Newton	Struct. quasi-Newton	A–M/Newton
1	0.7414D+05	0.7414D+05	0.7414D+05	0.7414D+05
2	0.8759D+05	0.2541D+06	0.8759D+05	0.8759D+05
3	0.9096D+05	0.9653D+05	0.9096D+05	0.9096D+05
4	0.1132D+06	0.3524D+05	0.1132D+06	0.1132D+06
5	0.9806D+05	0.2865D+05	0.9806D+05	0.9806D+05
6	0.4420D+06	0.2815D+05	0.4420D+06	0.4420D+06
7	0.2481D+06	0.2792D+05	0.2481D+06	0.4420D+06
8	0.9253D+05	0.3450D+05	0.2266D+06	0.1276D+06
9	0.2855D+06	0.3221D+05	0.1442D+06	0.1086D+06
10	0.1158D+06	0.3660D+06	0.2071D+07	0.9464D+05
.	.	.	.	.
20	0.4508D+06	0.4804D+05	0.2209D+06	0.2075D–04
.	.	.	.	
25	0.9670D+05	0.2119D+04	0.4320D+03	
26	0.1745D+06	0.5602D+02	0.2452D+03	
27	0.2759D+06	0.8048D+00	0.5247D+01	
28	0.9467D+05	0.2368D–03	0.1401D+00	
29	0.1497D+06	0.1923D–05	0.4371D–02	
30	0.2224D+06		0.5522D–05	
.	.			
1000	0.9331D+03			

This example shows that the Anderson–Moore method can perform very poorly due to its slow linear convergence. The structured quasi-Newton method was again the best, and even if it needed one step more than Newton's method to reach the termination criterion $\|\nabla J(F)\| < 10^{-4}$, it performed considerably faster in terms of CPU-time (0.8 seconds vs. 1.1 seconds on a DEC 3000).

It has been noted in the literature that the convergence of the Anderson–Moore algorithm can often be improved by checking the condition number of $C\ L(F)\ C^T$. If the condition number is high, one adds a regularization term and replaces its inverse in the Anderson–Moore algorithm by $(C\ L(F)\ C^T + \alpha I)^{-1}$. We tried this strategy on the last example and found that it does not improve the behavior of the Anderson–Moore method. The condition number is only in the range of 10^5; however, the Hessian $\nabla^2 J(F)$ is indefinite initially, which causes the slow behavior, because the Anderson–Moore algorithm uses part of the Hessian information as noted above.

REFERENCES

[1] B. D. O. ANDERSON AND J. B. MOORE, *Linear Optimal Control*, Prentice–Hall, Englewood Cliffs, NJ, 1971.

[2] D. S. Bernstein, *Some open problems in matrix theory arising in linear systems and control*, Linear Algebra Appl., (1992), pp. 409–432.

[3] S. S. Choi and H. R. Sirisena, *Computation of optimal output feedback gains for linear multivariable systems*, IEEE Trans. Automat. Control, 19 (1974), pp. 257–258.

[4] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1983.

[5] J. E. Dennis and H. F. Walker, *Convergence theorems for least change secant update methods*, SIAM J. Numer. Anal., 18 (1981), pp. 949–987.

[6] H. P. Horisberger and P. Bélanger, *Solution of the optimal constant output feedback problem by conjugate gradients*, IEEE Trans. Automat. Control, 26 (1974), pp. 434–435.

[7] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.

[8] D. H. Jacobson, D. H. Martin, M. Pachter, and T. Geveci, *Extensions of Linear-Quadratic Control Theory*, Lecture N. Control Inform. Sciences 27, Springer, Berlin, Heidelberg, New York, 1980.

[9] P. M. Mäkilä and H. T. Toivonen, *Computational methods for parametric LQ problems – A survey*, IEEE Trans. Automat. Control, 32 (1987), pp. 658–671.

[10] H. T. Toivonen, *A globally convergent algorithm for the optimal constant output feedback problem*, Int. J. Control, 41 (1985), pp. 1589–1599.

[11] H. T. Toivonen and P. M. Mäkilä, *A descent Anderson–Moore algorithm for optimal decentralized control*, Automatica, 21 (1985), pp. 734–744.

SIAM J. OPTIM.
Vol. 7, No. 3, pp. 853–870, August 1997

AN ϵ-RELAXATION METHOD FOR SEPARABLE CONVEX COST NETWORK FLOW PROBLEMS*

DIMITRI P. BERTSEKAS[†], LAZAROS C. POLYMENAKOS[†], AND PAUL TSENG[‡]

Abstract. We propose a new method for the solution of the single commodity, separable convex cost network flow problem. The method generalizes the ϵ-relaxation method developed for linear cost problems and reduces to that method when applied to linear cost problems. We show that the method terminates with a near optimal solution, and we provide an associated complexity analysis. We also present computational results showing that the method is much faster than earlier relaxation methods, particularly for ill-conditioned problems.

Key words. network optimization, ϵ-relaxation, network flows

AMS subject classifications. 90C35, 90C25

PII. S1052623495285886

1. Introduction.

We consider a directed graph with node set $\mathcal{N} = \{1, \dots, N\}$ and arc set $\mathcal{A} \subset \mathcal{N} \times \mathcal{N}$. The number of nodes is N and the number of arcs is denoted by A. We denote by x_{ij} the *flow* of the arc (i, j), and we refer to the vector $x = \{x_{ij} \mid (i, j) \in \mathcal{A}\}$ as the flow vector. The separable convex cost network flow problem is

$$\text{(P)} \qquad \text{minimize} \quad \sum_{(i,j)\in\mathcal{A}} f_{ij}(x_{ij})$$

$$\text{(1)} \qquad \text{subject to} \quad \sum_{\{j \mid (i,j)\in\mathcal{A}\}} x_{ij} - \sum_{\{j \mid (j,i)\in\mathcal{A}\}} x_{ji} = s_i \qquad \forall\, i \in \mathcal{N},$$

where s_i are given scalars and $f_{ij} : \Re \to (-\infty, \infty]$ are given convex, closed, proper functions (extended real valued, lower semicontinuous, not identically taking the value ∞). We refer to problem (P) as the *primal* problem. We have implicitly assumed that there exists at most one arc in each direction between any pair of nodes, but this assumption is made for notational convenience and easily can be dispensed with. A flow vector x with $f_{ij}(x_{ij}) < \infty$ for all $(i, j) \in \mathcal{A}$, which satisfies the conservation of flow constraint (1), is called *feasible*. For a given flow vector x, the *surplus* of node i is defined as the difference between the supply s_i and the net outflow from i:

$$\text{(2)} \qquad g_i = s_i + \sum_{\{j \mid (j,i)\in\mathcal{A}\}} x_{ji} - \sum_{\{j \mid (i,j)\in\mathcal{A}\}} x_{ij}.$$

We will assume that *there exists at least one feasible flow vector x such that*

$$\text{(3)} \qquad f_{ij}^-(x_{ij}) < \infty \ \text{ and } \ f_{ij}^+(x_{ij}) > -\infty \qquad \forall\, (i, j) \in \mathcal{A},$$

where $f_{ij}^-(x_{ij})$ and $f_{ij}^+(x_{ij})$ denote the left and right directional derivatives of f_{ij} at x_{ij} [Roc84, p. 329].

*Received by the editors May 8, 1995; accepted for publication (in revised form) January 18, 1996. This research was supported by the NSF under grants CCR-9103804 and 9300494-DMI.
http://www.siam.org/journals/siopt/7-3/28588.html

[†] Department of Electrical Engineering and Computer Science, M.I.T., Rm. 35-210, Cambridge, MA 02139 (dimitrib@mit.edu, lcpolyme@lids.mit.edu).

[‡] Department of Mathematics, University of Washington, Seattle, WA 98195 (tseng@math.washington.edu).

There is a well-known duality framework for this problem, primarily developed by Rockafellar [Roc70] and discussed in several texts; see, e.g., [Roc84], [BeT89]. This framework involves a Lagrange multiplier p_i for the ith conservation of flow constraint (1). We refer to p_i as the *price* of node i and to the vector $p = \{p_i \mid i \in \mathcal{N}\}$ as the *price vector*. The *dual* problem is as follows:

$$\text{(D)} \qquad \begin{array}{ll} \text{minimize} & q(p) \\ \text{subject to} & \text{no constraint on } p, \end{array}$$

where the dual functional q is given by

$$q(p) = \sum_{(i,j)\in\mathcal{A}} q_{ij}(p_i - p_j) - \sum_{i\in\mathcal{N}} s_i p_i,$$

and q_{ij} is related to f_{ij} by the conjugacy relation

$$q_{ij}(t_{ij}) = \sup_{x_{ij}\in\Re} \{x_{ij}t_{ij} - f_{ij}(x_{ij})\}.$$

We will assume throughout that f_{ij} *is such that* q_{ij} *is real valued for all* $(i, j) \in \mathcal{A}$. This is true, for example, if each function f_{ij} takes the value ∞ outside some compact interval.

It is known (see [Roc84, p. 360]) that under our assumptions, both the primal problem (P) and the dual problem (D) have optimal solutions, and their optimal costs are the negatives of each other. The standard optimality conditions for a feasible flow-price vector pair (x, p) to be primal and dual optimal are

$$f_{ij}^-(x_{ij}) \le p_i - p_j \le f_{ij}^+(x_{ij}) \qquad \forall\ (i,j) \in \mathcal{A}.$$

These, known as the *complementary slackness conditions* (CS conditions), may be represented explicitly as

$$(x_{ij}, p_i - p_j) \in \Gamma_{ij} \qquad \forall\ (i,j) \in \mathcal{A},$$

where

$$\Gamma_{ij} = \left\{(x_{ij}, t_{ij}) \in \Re^2 \mid f_{ij}^-(x_{ij}) \le t_{ij} \le f_{ij}^+(x_{ij})\right\}$$

is the *characteristic curve* associated with arc (i, j) as shown in Fig. 1.

The traditional methods for solving the problem of this paper for the case of linear arc cost functions (when each f_{ij} is linear on some closed interval and is ∞ outside the interval) are primal and dual cost improvement methods, which iteratively improve the primal or the dual cost function. Recently, methods based on the auction approach have gained attention, following the original proposal of the auction algorithm for the assignment problem [Ber79] and the ϵ-relaxation method [Ber86a], [Ber86b]. These methods may not improve the primal or the dual cost at any iteration, and they are based on a relaxed version of the CS conditions, called ϵ-complementary slackness (ϵ-CS). Their worst-case computational complexity, when properly implemented, is excellent; see [Gol87] (see also [BeE88], [BeT89], [GoT90]). Their practical performance is also very good, particularly for special classes of problems such as assignment and max-flow. Furthermore, these methods are well suited for parallel implementation (see [BCE95], [LiZ91], [NiZ93]). We will focus on extending one such method, the ϵ-relaxation method, to the general convex cost case.

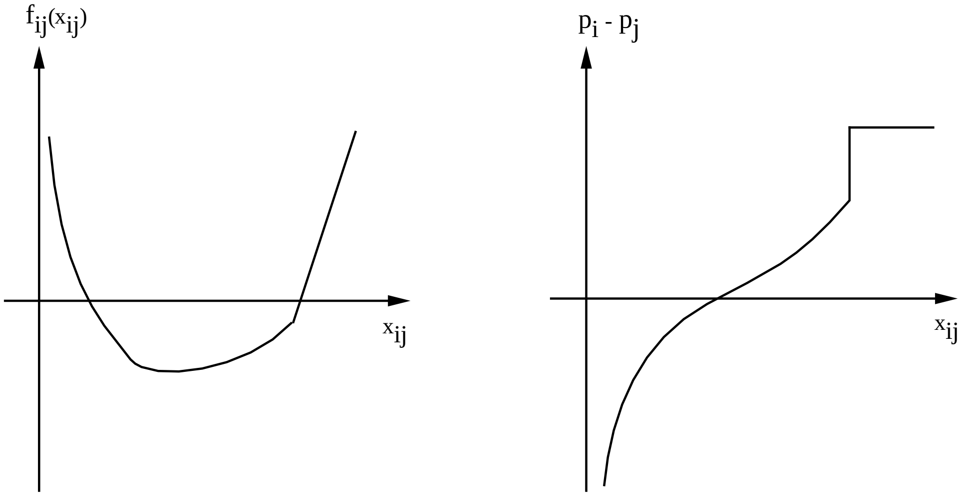


FIG. 1. *A cost function f_{ij} and its corresponding characteristic curve.*

One possibility for dealing with the convex cost case is to use efficient ways to reduce the problem to an essentially linear cost problem by piecewise linearization of the arc cost functions; see [Mey79], [KaM84], [Roc84]. Another possibility is to use differentiable unconstrained optimization methods, such as coordinate descent [BHT87], conjugate gradient [Ven91], and adaptations of other nonlinear programming methods [HaH93], [Hag92] or fixed point methods [BeE87], [TBT90]. However, these methods require that the dual cost function be differentiable, which essentially amounts to the primal cost functions being strictly convex. A more general alternative, which applies to nondifferentiable dual cost functions as well, is to use an extension of the primal or dual cost improvement methods developed for the linear cost case. In particular, there have been proposals of primal cost improvement methods in [Wei74] and more recently in [KaM93]. There have also been proposals of dual cost improvement methods: see the fortified descent method [Roc84] that extends the primal–dual method of Ford and Fulkerson [FoF62] and the relaxation method of [BHT87] that extends the corresponding linear cost relaxation method of [Ber85] and [BeT88]. These methods, together with the price vector, maintain a flow vector that satisfies the ϵ-CS conditions and progressively work towards primal feasibility. The flow vector becomes feasible at termination.

In this paper we develop and analyze the first extension of an auction method, the ϵ-relaxation method, to the convex arc cost case.[1] (An analogous extension of the auction/sequential shortest path method given in [Ber92], which has also been incorporated in the latest release of the RELAX code [BeT94], is developed in the Ph.D. thesis of the second author [Pol95].) Our method is based on the ϵ-CS conditions first introduced in [BHT87] for the case of convex arc costs. In particular, we say that the flow vector x and the price vector p satisfy ϵ-CS if and only if (see Fig. 2)

$$(4) \qquad f_{ij}(x_{ij}) < \infty \quad \text{and} \quad f_{ij}^{-}(x_{ij}) - \epsilon \le p_i - p_j \le f_{ij}^{+}(x_{ij}) + \epsilon \qquad \forall\ (i,j) \in \mathcal{A}.$$

[1] We have learned that the same method was independently developed and analyzed by De Leone, Meyer, and Zakarian [DMZ95]. The results of their computational tests qualitatively agree with ours.

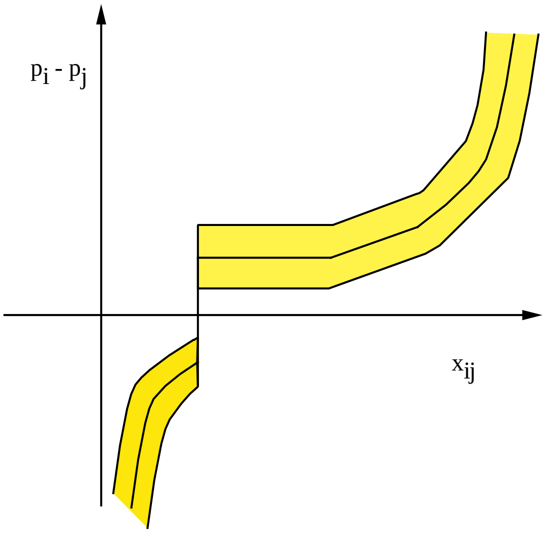


FIG. 2. *A visualization of the ϵ-CS conditions as a cylinder around the characteristic curve (bold line). The shaded area represents flow-price differential pairs that satisfy the ϵ-CS conditions.*

It was shown in [BHT87] that if a feasible flow-price vector pair (x, p) satisfies ϵ-CS, then x and p are primal and dual optimal, respectively, within a factor that is proportional to ϵ (see Proposition 6). Our method is similar to the ϵ-relaxation method for linear cost network flow problems. It iteratively modifies the price vector while effecting attendant flow changes that maintain the ϵ-CS conditions. The method terminates with a feasible flow-price vector pair which, however, satisfies ϵ-CS rather than CS. There is a fundamental difference from the other dual descent methods for nondifferentiable dual cost problems: the price changes are made exclusively along coordinate directions, that is, one price at a time, and a price change need not improve the dual cost. However, because the flow-price vector pair (x, p) maintained by the ϵ-relaxation method satisfies ϵ-CS rather than CS, there is more freedom in adjusting the flow-price vector pair towards feasibility, even though the pair obtained when the method terminates is optimal only within a factor proportional to ϵ.

The method of this paper essentially provides a mechanism to move around the ϵ-CS diagram of Fig. 2 while changing one price at a time and works towards primal feasibility. There is a variety of mechanisms for effecting such price changes and Fig. 3 illustrates some of the possibilities. In particular, by starting from a point on the characteristic curve of arc (i, j) we can follow any direction around that point and change the price p_i or the price p_j and/or the flow x_{ij} simultaneously until $(x_{ij}, p_i - p_j)$ is either on the characteristic curve or is within a distance of ϵ above or below the characteristic curve of arc (i, j). For example, if node i has positive surplus, by increasing the flow of an outgoing arc (i, j) or by decreasing the flow of an incoming arc (j, i) the surplus of i will be decreased, while the surplus of j will be increased by an equal amount. This is the basic mechanism for moving flow from nodes of positive surplus to nodes of negative surplus, thus working towards primal feasibility. It is possible, however, that node i has positive surplus, while the flow of none of the outgoing arcs (i, j) can be increased and the flow of none of the incoming arcs (j, i) can be decreased without violating the ϵ-CS conditions. In this case, the method increases the price of node i in order to “make room” in the ϵ-CS diagram

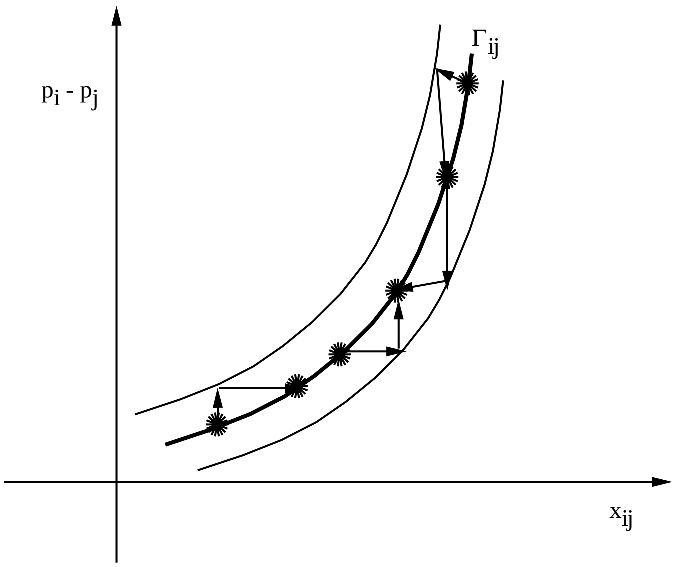


FIG. 3. *Starting from any point on the characteristic curve (dark points) of arc (i, j), a new point on the characteristic curve can be obtained in a variety of ways. The figure depicts a few such examples where the flow and price differential for arc (i, j) are changed simultaneously according to some linear relation.*

for a subsequent flow change.

The paper is organized as follows. In section 2 we present the ϵ-relaxation method extended to solve convex cost problems. In section 3, we show that this method terminates with a near optimal flow-price vector pair, and in section 4 we provide a complexity analysis for the method. In section 5, we describe a version of this method that uses both price increase and decrease steps, and in section 6 we report our computational experience with the methods of sections 2 and 5 on some convex linear/quadratic cost problems. Our test results show that, on problems where some (possibly all) arcs have strictly convex cost, the new method outperforms, often by an impressive margin, earlier relaxation methods. Significantly, our method seems to be minimally affected by ill-conditioning in the dual problem. We do not know of any other method for which this is true.

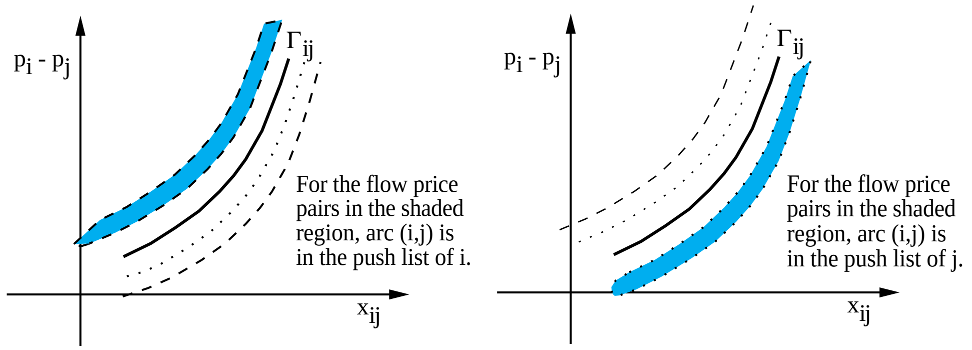
2. The ϵ-relaxation method. For a flow-price vector pair (x, p) satisfying ϵ-CS, we define for each node $i \in \mathcal{N}$ its *push list* as the set of arcs

$$(5)\ \ \big\{(i,j)\mid \epsilon/2 < p_i - p_j - f^+_{ij}(x_{ij}) \le \epsilon\big\} \cup \big\{(j,i)\mid -\epsilon \le p_j - p_i - f^-_{ji}(x_{ji}) < -\epsilon/2\big\}.$$

Figure 4 illustrates when an arc (i, j) is in the push list of i and when it is in the push list of j. We note that a more general definition of the push list can be given by replacing the term $\epsilon/2$ with $\beta\epsilon$, where β is a scalar with $0 < \beta < 1$. The subsequent analysis applies, with minor modifications, to the corresponding version of the ϵ-relaxation method to be given shortly.

An arc (i, j) [or (j, i)] in the push list of i is said to be *unblocked* if there exists a $\delta > 0$ such that

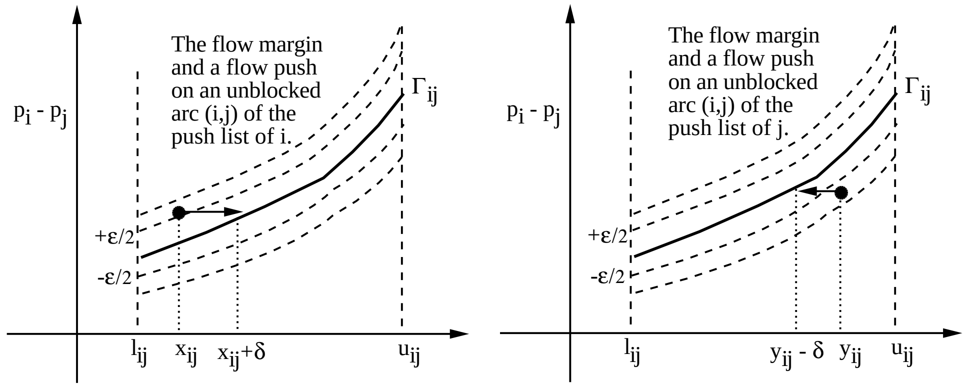
$$p_i - p_j \ge f^+_{ij}(x_{ij} + \delta)$$



For the flow price pairs in the shaded region, arc (i,j) is in the push list of i.

For the flow price pairs in the shaded region, arc (i,j) is in the push list of j.

FIG. 4. *A visualization of the conditions satisfied by a push list arc. The shaded area represents flow-price differential pairs corresponding to a push list arc.*



The flow margin and a flow push on an unblocked arc (i,j) of the push list of i.

The flow margin and a flow push on an unblocked arc (i,j) of the push list of j.

FIG. 5. *The flow margin of an unblocked push list arc.*

(or $p_j - p_i \leq f^-_{ji}(x_{ji} - \delta)$, respectively). For an unblocked push list arc, the supremum of δ for which the above relation holds is called the *flow margin* of the arc. The flow margin of an arc (i, j) is illustrated in Fig. 5. An important property is noted in the following proposition.

PROPOSITION 1. *The arcs in the push list of a node are unblocked.*

Proof. Assume that for an arc $(i, j) \in \mathcal{A}$ we have

$$p_i - p_j < f^+_{ij}(x_{ij} + \delta) \quad \forall\, \delta > 0.$$

Since the function f^+_{ij} is right continuous, this yields

$$p_i - p_j \leq \lim_{\delta \downarrow 0} f^+_{ij}(x_{ij} + \delta) = f^+_{ij}(x_{ij}),$$

and thus (i, j) cannot be in the push list of node i. A similar argument proves that an arc $(j, i) \in \mathcal{A}$ such that

$$p_j - p_i > f^-_{ji}(x_{ji} - \delta) \quad \forall\, \delta > 0$$

cannot be in the push list of node i. □

For a given pair (x, p) satisfying ϵ-CS, consider an arc set $\mathcal{A}^*$ that contains all push list arcs oriented in the direction of flow change. In particular, for each arc (i, j)

in the push list of a node i we introduce an arc (i, j) in $\mathcal{A}^*$, and for each arc (j, i) in the push list of node i we introduce an arc (i, j) in $\mathcal{A}^*$. The set of nodes $\mathcal{N}$ and the set $\mathcal{A}^*$ define the *admissible graph* $G^* = (\mathcal{N}, \mathcal{A}^*)$. Note that an arc can be in the push list of at most one node, so the admissible graph is well defined.

The ϵ-relaxation method starts with a flow-price vector pair (x, p) satisfying ϵ-CS and is such that the corresponding admissible graph G^* is acyclic. One possibility is to select an initial price vector p^0 and to set the initial arc flow for every arc $(i, j) \in \mathcal{A}$ to

$$x_{ij} = \sup\{\xi \mid f_{ij}^+(\xi) \leq p_i^0 - p_j^0 - \epsilon/2\}. \tag{6}$$

It can be seen with this choice that ϵ-CS is satisfied for every arc $(i, j) \in \mathcal{A}$ and that the initial admissible graph is empty and thus acyclic.

In the typical iteration of the method, we select a node i with positive surplus, and we perform one or more of the following two operations:

(a) a *price rise* on node i, which increases the price p_i by the maximum amount that maintains ϵ-CS, while leaving all arc flows unchanged,

(b) a *flow push* (also called a *δ-flow push*) along an arc (i, j) (or along an arc (j, i)), which increases (i, j) (or decreases (j, i)) by an amount $\delta \in (0, g_i]$, while leaving all node prices unchanged.

The iteration is as follows.

Typical Iteration of the ϵ-Relaxation Method.

Step 1: Select a node i with positive surplus g_i; if no such node exists, terminate the method.

Step 2: If the push list of i is empty, go to Step 3. Otherwise, choose an arc from the push list of i and perform a δ-flow push towards the opposite node j, where

$$\delta = \min\{g_i, \text{ flow margin of arc}\}.$$

If the surplus of i becomes zero, go to the next iteration; otherwise go to Step 2.

Step 3: Increase the price p_i by the maximum amount that maintains ϵ-CS. Go to the next iteration.

We make the following observations about the ϵ-relaxation method:

1. The method preserves ϵ-CS and the prices are monotonically nondecreasing. This is evident from the initialization and Step 3 of the method.
2. Once the surplus of a node becomes nonnegative, it remains nonnegative for all subsequent iterations. The reason is that a flow push at a node i cannot make the surplus of i negative (cf. Step 2) and cannot decrease the surplus of neighboring nodes.
3. If at some iteration a node has negative surplus, then its price must be equal to its initial price. This is a consequence of observation 2 above and the fact that price changes occur only on nodes with positive surplus.

3. Termination of the ϵ-relaxation method. To prove the termination of the ϵ-relaxation method of section 2, we first prove that the total number of price rises that the method can perform is bounded.

Proposition 2. *Each price rise increment in the ϵ-relaxation method is at least $\epsilon/2$.*

Proof. We first note that a price rise on a node i occurs only when its push list is empty. Thus, for every arc $(i, j) \in \mathcal{A}$ we have $p_i - p_j - f_{ij}^+(x_{ij}) \leq \epsilon/2$, and for every arc $(j,i) \in \mathcal{A}$ we have $p_j - p_i - f_{ji}^-(x_{ji}) \geq -\epsilon/2$. This implies that all elements of the sets of positive numbers

$$S^+ = \left\{ p_j - p_i + f_{ij}^+(x_{ij}) + \epsilon \mid (i,j) \in \mathcal{A} \right\},$$

$$S^- = \left\{ p_j - p_i - f_{ji}^-(x_{ji}) + \epsilon \mid (j,i) \in \mathcal{A} \right\}$$

are greater than or equal to $\epsilon/2$. Since a price rise at i increases p_i by the increment $\gamma = \min\{S^+ \cup S^-\}$, the result follows. □

The following proposition bounds the total number of price increases that the ϵ-relaxation method can perform on any node. The proof is patterned after that for the linear cost case [Ber86a], [BeE88].

PROPOSITION 3. *Assume that for some integer $K \geq 1$, the initial price vector p^0 for the ϵ-relaxation method satisfies $K\epsilon$-CS together with some feasible flow vector x^0. Then, the ϵ-relaxation method performs at most $2(K+1)(N-1)$ price rises per node.*

Proof. Consider the pair (x,p) at the beginning of an ϵ-relaxation iteration. Since the surplus vector $g = (g_1, \ldots, g_N)$ is not zero and the flow vector x^0 is feasible, we conclude that for each node s with $g_s > 0$ there exists a node t with $g_t < 0$ and a path H from t to s that contains no cycles and is such that

$$x_{ij} > x_{ij}^0 \qquad \forall\ (i,j) \in H^+, \tag{7}$$

$$x_{ij} < x_{ij}^0 \qquad \forall\ (i,j) \in H^-, \tag{8}$$

where H^+ is the set of forward arcs of H and H^- is the set of backward arcs of H. (This can be seen from the conformal realization theorem ([Roc84] or [Ber91]) as follows. For the flow vector $x - x^0$, the net outflow from node t is $-g_t > 0$ and the net outflow from node s is $-g_s < 0$ (here we ignore the flow supplies), so, by the conformal realization theorem, there is a path H from t to s that contains no cycle and conforms to the flow $x - x^0$. That is, $x_{ij} - x_{ij}^0 > 0$ for all $(i,j) \in H^+$ and $x_{ij} - x_{ij}^0 < 0$ for all $(i,j) \in H^-$. Equations (7) and (8) then follow.)

From equations (7) and (8) and the convexity of the functions f_{ij} for all $(i,j) \in \mathcal{A}$, we have

$$f_{ij}^-(x_{ij}) \geq f_{ij}^+(x_{ij}^0) \qquad \forall\ (i,j) \in H^+, \tag{9}$$

$$f_{ij}^+(x_{ij}) \leq f_{ij}^-(x_{ij}^0) \qquad \forall\ (i,j) \in H^-. \tag{10}$$

Since the pair (x,p) satisfies ϵ-CS, we also have that

$$p_i - p_j \in [f_{ij}^-(x_{ij}) - \epsilon, f_{ij}^+(x_{ij}) + \epsilon] \qquad \forall\ (i,j) \in \mathcal{A}. \tag{11}$$

Similarly, since the pair (x^0, p^0) satisfies $K\epsilon$-CS, we have

$$p_i^0 - p_j^0 \in [f_{ij}^-(x_{ij}^0) - K\epsilon, f_{ij}^+(x_{ij}^0) + K\epsilon] \qquad \forall\ (i,j) \in \mathcal{A}. \tag{12}$$

Combining equations (9)–(12), we obtain

$$p_i - p_j \geq p_i^0 - p_j^0 - (K+1)\epsilon \qquad \forall\ (i,j) \in H^+,$$

$$p_i - p_j \le p_i^0 - p_j^0 + (K+1)\epsilon \qquad \forall\ (i,j) \in H^-.$$

Applying the above inequalities for all arcs of the path H, we get

$$p_t - p_s \ge p_t^0 - p_s^0 - (K+1)|H|\epsilon, \tag{13}$$

where $|H|$ denotes the number of arcs of the path H. We observed earlier that if a node has negative surplus at some time, then its price is unchanged from the beginning of the method until that time; thus, $p_t = p_t^0$. Since the path contains no cycles, we also have that $|H| \le N-1$. Therefore, equation (13) yields

$$p_s - p_s^0 \le (K+1)|H|\epsilon \le (K+1)(N-1)\epsilon. \tag{14}$$

Since only nodes with positive surplus can increase their prices and, by Proposition 2, each price rise increment is at least $\epsilon/2$, we conclude from equation (14) that the total number of price rises that can be performed for node s is at most $2(K+1)$ $(N–1)$. □

The result of the preceding proposition is remarkable in that the bound on the number of price changes is independent of the cost functions but depends only on

$$K^0 = \min\{K \in \{0,1,\ldots\} \mid (x^0,p^0) \text{ satisfies } K\epsilon\text{-CS for some feasible flow vector } x^0\ \},$$

which is the minimum multiplicity of ϵ by which CS is violated by the starting price together with some feasible flow vector. This result will be used later to prove a particularly favorable complexity bound for the method. Note that K^0 is well defined for any p^0 because, for all K sufficiently large, $K\epsilon$-CS is satisfied by p^0 and the feasible flow vector x satisfying equation (3).

In order to show that the number of flow pushes that can be performed between successive price increases is finite, we first prove that the method maintains the acyclicity of the admissible graph.

PROPOSITION 4. *The admissible graph remains acyclic throughout the ϵ-relaxation method.*

Proof. We use induction. Initially, the admissible graph G^* is empty, so it is trivially acyclic. Assume that G^* remains acyclic for all subsequent iterations up to the mth iteration for some m. We will prove that after the mth iteration G^* remains acyclic. Clearly, after a flow push the admissible graph remains acyclic, since it either remains unchanged or some arcs are deleted from it. Thus, we only have to prove that after a price rise at a node i, no cycle involving i is created. We note that after a price rise at node i, all incident arcs to i in the admissible graph at the start of the mth iteration are deleted and new arcs incident to i are added. We claim that i cannot have any incoming arcs which belong to the admissible graph. To see this, note that just before a price rise at node i, we have from (4) that

$$p_j - p_i - f_{ji}^-(x_{ji}) \le \epsilon \qquad \forall\ (j,i) \in \mathcal{A},$$

and since each price rise is at least $\epsilon/2$, we must have

$$p_j - p_i - f_{ji}^-(x_{ji}) \le \frac{\epsilon}{2} \qquad \forall\ (j,i) \in \mathcal{A}$$

after the price rise. Then, by equation (5), (j,i) cannot be in the push list of node j. By a similar argument, we have that (i,j) cannot be in the push list of j for all

$(i,j) \in \mathcal{A}$. Thus, after a price increase at i, node i cannot have any incoming incident arcs belonging to the admissible graph, so no cycle involving i can be created. □

We say that a node i is a *predecessor* of a node j in the admissible graph G^* if a directed path from i to j exists in G^*. Node j is then called a *successor* of i. Observe that flow is pushed towards the successors of a node, and since G^* is acyclic, flow cannot be pushed from a node to any of its predecessors. A δ-flow push along an arc in G^* is said to be *saturating* if δ is equal to the flow margin of the arc. By our choice of δ (see Step 2 of the method), a nonsaturating flow push always exhausts (i.e., sets to zero) the surplus of the starting node of the arc. Thus we have the following proposition.

PROPOSITION 5. *The number of flow pushes between two successive price increases (not necessarily at the same node) performed by the ϵ-relaxation method is finite.*

Proof. We observe that a saturating flow push along an arc removes the arc from the admissible graph, while a nonsaturating flow push does not add a new arc to the admissible graph. Thus, the number of saturating flow pushes that can be performed between successive price increases is at most A. It will thus suffice to show that the number of nonsaturating flow pushes that can be performed between saturating flow pushes is finite. Assume the contrary; that is, there is an infinite sequence of successive nonsaturating flow pushes with no intervening saturating flow push. Then, the surplus of some node i^0 must be exhausted infinitely often during this sequence. This can happen only if the surplus of some predecessor i^1 of i^0 is exhausted infinitely often during the sequence. Continuing in this manner, we construct an infinite succession of predecessor nodes $\{i^k\}$. Thus, some node in this sequence must be repeated, which is a contradiction since the admissible graph is acyclic. □

By refining the proof of Proposition 5, we can further show that the number of flow pushes between successive price increases is at most $(N+1)A$, from which a complexity result for the ϵ-relaxation method may be derived. However, we will defer the analysis of complexity to section 4, where an implementation of the method with sharper complexity bound will be presented.

Propositions 3 and 5 prove that the ϵ-relaxation method terminates. Upon termination, we have that the flow-price vector pair satisfies ϵ-CS and that the flow vector is feasible since the surplus of all nodes will be zero. The following proposition, due to [BHT87], shows that the flow vector and the price vector obtained upon termination are primal optimal and dual optimal within a factor that is essentially proportional to ϵ.

PROPOSITION 6. *For each $\epsilon > 0$, let $x(\epsilon)$ and $p(\epsilon)$ denote any flow and price vector pair satisfying ϵ-CS with $x(\epsilon)$ feasible, and let $\xi(\epsilon)$ denote any flow vector satisfying CS together with $p(\epsilon)$ (note that $\xi(\epsilon)$ need not be feasible). Then*

$$0 \le f\left(x(\epsilon)\right) + q\left(p(\epsilon)\right) \le \epsilon \sum_{(i,j)\in\mathcal{A}} \left|x_{ij}(\epsilon) - \xi_{ij}(\epsilon)\right|.$$

Furthermore, $f\left(x(\epsilon)\right) + q\left(p(\epsilon)\right) \to 0$ as $\epsilon \to 0$.

Proposition 6 does not give an estimate of how small ϵ has to be in order to achieve a certain degree of optimality. However, in the common case where finiteness of the arc cost functions f_{ij} imply lower and upper bounds on the arc flows, i.e.,

$$-\infty < b_{ij} = \inf_{\xi}\{\xi \mid f_{ij}(\xi) < \infty\} \le \sup_{\xi}\{\xi \mid f_{ij}(\xi) < \infty\} = c_{ij} < \infty,$$

Proposition 6 together with the fact that $q\left(p(\epsilon)\right) \geq -f^*$ yields the estimate

$$0 \leq f\left(x(\epsilon)\right) - f^* \leq \epsilon A \max_{(i,j)\in\mathcal{A}} |c_{ij} - b_{ij}|,$$

where f^* is the optimal cost of (P). Similarly, we obtain

$$0 \leq q\left(p(\epsilon)\right) - q^* \leq \epsilon A \max_{(i,j)\in\mathcal{A}} |c_{ij} - b_{ij}|,$$

where q^* is the optimal cost of (D).

4. Complexity analysis for the ϵ-relaxation method. We now derive a bound on the running time of the ϵ-relaxation method. Because the cost functions are convex, it is not possible to express the size of the problem in terms of the problem data. To deal with this difficulty, we introduce a set of simple operations performed by the method, and we estimate the number of these operations. In particular, in addition to the usual arithmetic operations with real numbers, we consider the following operations:

(a) Given the flow x_{ij} of an arc (i, j), calculate the cost $f_{ij}(x_{ij})$, the left derivative $f_{ij}^-(x_{ij})$, and the right derivative $f_{ij}^+(x_{ij})$.

(b) Given the price differential t_{ij} of an arc (i, j), calculate $\sup\{\xi \mid f_{ij}^+(\xi) \leq t_{ij}\}$ and $\inf\{\xi \mid f_{ij}^-(\xi) \geq t_{ij}\}$.

Operation (a) is needed to compute the push list of a node and a price increase increment; operation (b) is needed to compute the flow margin of an arc and the flow initialization of equation (6). We will thus estimate the total number of simple operations performed by the method (see Proposition 8).

To obtain a sharper complexity bound, we introduce an order in which the nodes are chosen in iterations. This rule is based on the *sweep implementation* of the ϵ-relaxation method, which was introduced in [Ber86a] and was analyzed in more detail in [BeE88], [BeT89], and [BeC93] for the linear cost network flow problem. All the nodes are kept in a linked list T, which is traversed from the first to the last element. The order of the nodes in the list is consistent with the successor order implied by the admissible graph; that is, if a node j is a successor of a node i, then j must appear after i in the list. If the initial admissible graph is empty, as is the case with the initialization of equation (6), the initial list is arbitrary. Otherwise, the initial list must be consistent with the successor order of the initial admissible graph. The list is updated in a way that maintains the consistency with the successor order. In particular, let i be a node on which we perform an ϵ-relaxation iteration, and let N_i be the subset of nodes of T that are after i in T. If the price of i changes, then node i is removed from its position in T and placed in the first position of T. The next node chosen for iteration, if N_i is nonempty, is the node $i' \in N_i$ with positive surplus, which ranks highest in T. Otherwise, the positive surplus node ranking highest in T is picked. It can be shown (see the references cited earlier) that with this rule of repositioning nodes following a price change, the list order is consistent with the successor order implied by the admissible graph throughout the method.

A *sweep cycle* is a set of iterations whereby all nodes are chosen once from the list T, and an ϵ-relaxation iteration is performed on those nodes that have positive surplus. The idea of the sweep implementation is that an ϵ-relaxation iteration at a node i that has predecessors with positive surplus may be wasteful, since the surplus of i will be set to zero and become positive again through a flow push at a predecessor node.

Our complexity analysis follows the line of the corresponding analysis for the linear cost problem. First we have a proposition that estimates the number of sweep cycles required for termination.

PROPOSITION 7. *Assume that for some integer $K \geq 1$, the initial price vector p^0 for the sweep implementation of the ϵ-relaxation method satisfies $K\epsilon$-CS together with some feasible flow vector x^0. Then, the number of sweep cycles up to termination is $O(KN^2)$.*

Proof. Consider the start of any sweep cycle. Let N^+ be the set of nodes with positive surplus that have no predecessor with positive surplus; let N^0 be the set of nodes with nonpositive surplus that have no predecessor with positive surplus. Then, as long as no price change takes place during the cycle, all nodes in N^0 remain in N^0, and an iteration on a node $i \in N^+$ moves i from N^+ to N^0. So if no node changed price during the cycle, then all nodes in N^+ will be moved to N^0 and the method terminates. Therefore, there is a price change in every cycle except possibly the last one. Since by Proposition 3 there are $O(KN^2)$ price changes, the result follows. □

By using Proposition 7, we now bound the running time for the sweep implementation of the ϵ-relaxation method. The dominant computational requirements are as follows:

(1) the computation required for price increases,

(2) the computation required for saturating δ-flow pushes,

(3) the computation required for nonsaturating δ-flow pushes.

PROPOSITION 8. *Assume that for some $K \geq 1$ the initial price vector p^0 for the sweep implementation of the ϵ-relaxation method satisfies $K\epsilon$-CS together with some feasible flow vector x^0. Then, the method requires $O(KN^3)$ operations up to termination.*

Proof. According to Proposition 3, there are $O(KN)$ price increases per node, so the requirements for (1) above are $O(KNA)$ operations. Furthermore, whenever a flow push is saturating, it takes at least one price increase at one of the end nodes before the flow on that arc can be changed again. Thus, the total requirement for (2) above is $O(KNA)$ operations also. Finally, for (3) above we note that for each sweep cycle there can be only one nonsaturating δ-flow push per node. Thus, a time bound for (3) is $O(N \cdot \text{total number of sweep cycles})$, which, by Proposition 7, is $O(KN^3)$ operations. Adding the computational requirements for (1), (2), and (3) and using the fact that $A \leq N^2$, the result follows. □

It is well known that the theoretical and the practical performance of the ϵ-relaxation method can be improved by scaling. One possibility is *cost scaling* (see [BlJ92], [EdK72], [Roc80]). An analysis of cost scaling applied to ϵ-relaxation for the linear network flow problem is given in [BeE87] and also in [BeE88]. In the convex cost case, however, cost scaling may be difficult to implement since the arc cost functions may be unbounded. A second scaling approach in connection with the ϵ-relaxation method for linear cost problems is *ϵ-scaling.* This approach was originally introduced in [Ber79] as a means of improving the performance of the auction algorithm for the assignment problem. Its complexity analysis was given in [Gol87] and [GoT90].

The key idea of ϵ-scaling is to apply the ϵ-relaxation method several times, starting with a large value of ϵ, and to successively reduce ϵ up to a final value that will give the desirable degree of accuracy to our solution. Furthermore, the price and flow information from one application of the method is transferred to the next.

The procedure is as follows: first, we choose a scalar $\theta \in (0, 1)$, a price vector p^0, and a desirable value $\bar{\epsilon}$ for ϵ on termination. Next, we choose a sufficiently large ϵ^0 so that p^0 satisfies ϵ^0-CS with some feasible flow vector x^0. Then, for $k = 1, 2, \ldots$, we set $\epsilon^k = \theta\epsilon^{k-1}$, and for $k = 1, 2, \ldots, \bar{k}$, we apply the ϵ-relaxation method with $\epsilon = \epsilon^{k-1}$, where $\bar{k}$ is the first positive integer k for which ϵ^{k-1} is below $\bar{\epsilon}$. Let (x^k, p^k) be the flow-price vector pair obtained at the kth application of the method for $k = 1, 2, ..., \bar{k}$. Then, x^k is feasible and satisfies ϵ^{k-1}-CS with p^k. Furthermore, the admissible graph after the kth application of the method is acyclic. The initial price vector for the $(k+1)$st application is p^k, and the initial flow is x_{ij}^k for the arcs (i, j) that satisfy ϵ^k-CS with p^k; otherwise,

$$\sup\left\{\xi \mid f_{ij}^+(\xi) \leq p_i^k - p_j^k - \epsilon^k/2\right\}.$$

This choice of initial flows does not introduce any new arcs to the admissible graph, so the initial admissible graph for the $(k+1)$st application of the method is acyclic. For the 1st application of the method, the initial price vector is p^0 and the initial flow vector is chosen so that the initial admissible graph is acyclic.

We observe that for the $(k+1)$st application of the method $(k = 0, 1, \ldots \bar{k} - 1)$, the initial price vector p^k satisfies ϵ^k/θ-CS with the feasible flow vector x^k. Thus, based on Proposition 8, we conclude that the $(k+1)$st application of the method has a running time of $O\left(\lceil 1/\theta \rceil N^3\right)$, which is $O(N^3)$ since θ is a fixed scalar. The method will be applied at most $\bar{k} = \lceil \log_\theta(\epsilon^0/\bar{\epsilon}) \rceil$ times. We have thus obtained the following proposition.

Proposition 9. *The running time of the ϵ-relaxation method using the sweep implementation and ϵ-scaling as described above is $O\left(N^3 \ln(\epsilon^0/\bar{\epsilon})\right)$ operations.*

We note that a complexity bound of $O\left(NA \ln(N) \ln(\epsilon^0/\bar{\epsilon})\right)$ operations was derived in [KaM93] for the tighten and cancel method. For relatively dense problems where $A = \Theta(N^2/\ln N)$, our complexity bound for the ϵ-relaxation method is more favorable, while for sparse problems, where $A = \Theta(N)$, the reverse is true.

5. The reverse and forward–reverse ϵ-relaxation methods. The ϵ-relaxation method we presented in the previous sections performed iterations only on nodes with positive surplus. We will refer to it as the *forward* method. We can also define a method (namely, the *reverse* method), which performs iterations on nodes of negative surplus. This involves a simple reformulation of the flow and price changing operations we introduced in previous sections for the forward method. The reverse ϵ-relaxation method is the "mirror image" of the forward method that we developed in the previous sections. Naturally, it has similar properties to the forward method and its validity follows from a similar analysis.

It is possible to combine the forward and the reverse methods so that the resulting method will operate on both positive and negative surplus nodes. Our intuition is that if we perform ϵ-relaxation iterations on both sources and sinks, we will be able to find the optimal solution faster for certain classes of problems. We refer to the resulting method as the *forward–reverse method.* We initialize the arc flows and node prices in the same way we initialized them for the forward and the reverse methods so that the initial admissible graph is acyclic. The forward–reverse method operates as follows.

Typical Iteration of the Forward–Reverse ϵ-Relaxation Method.

> Pick a node i with nonzero surplus; if no such node exists then terminate. If i has positive surplus then perform an iteration of the forward ϵ-relaxation method. If i has negative surplus then perform an iteration of the reverse ϵ-relaxation method.

The idea of the forward–reverse method is recurrent in many relaxation-like methods. Termination of the method can be proved with an analysis similar to the one in section 3, provided that we also make the following assumption.

Assumption. The number of times the surplus of a node changes sign is finite.

The above assumption can be enforced by various mechanisms, some of which are discussed in [Tse86] for the relaxation method and in [Pol94] for the auction shortest path algorithm.

6. Computational results. We have developed and tested two experimental Fortran codes implementing the methods of this paper for convex cost problems. The first code, named NE-RELAX-F, implements the forward ϵ-relaxation method with the sweep implementation and ϵ-scaling as described in section 4. The second code, named NE-RELAX-FV, implements the forward–reverse version of NE-RELAX-F as described in section 5. These codes are based on the ϵ-relaxation code for linear cost problems described in Appendix 7 of [Ber91], which has been shown to be quite efficient. Several changes and enhancements were introduced in the codes for convex cost problems: all computations are done in real rather than integer arithmetic, and ϵ-scaling, rather than arc cost scaling, is used. Also, the updating of the push lists and prices are changed to improve efficiency. Otherwise, the sweep implementation and the general structure of the codes for linear and convex cost problems are identical. Initial testing on linear cost problems showed that the codes for convex cost problems perform as well as, and often better than, their counterparts for linear cost problems, which indicates that these codes are written efficiently. (The superior performance of the codes for convex cost problems may be due to the latter's efficient management of the push lists and the speed of floating point computations of the machine on which the codes were run.)

The codes NE-RELAX-F and NE-RELAX-FV were compared to two existing Fortran codes, NRELAX and MNRELAX from [BHT87]. The latter implement the relaxation method for, respectively, strictly convex cost and convex cost problems, and they are believed to be quite efficient. All codes were compiled and run on a Sun Sparc-5 workstation with 24 megabytes of RAM under the Solaris operating system. We used the -O compiler option in order to take advantage of the floating point unit and the design characteristics of the Sparc-5 processor. Unless otherwise indicated, all codes terminated according to the same criterion; namely, the cost of the feasible flow vector and the cost of the price vector agree in their first 12 digits.

For our testing, we used convex linear/quadratic problems corresponding to the case of (P) where

$$f_{ij}(x_{ij}) = \begin{cases} a_{ij}x_{ij} + b_{ij}x_{ij}^2 & \text{if } 0 \leq x_{ij} \leq c_{ij}, \\ \infty & \text{otherwise} \end{cases}$$

for some a_{ij}, b_{ij}, and c_{ij} with $-\infty < a_{ij} < \infty$, $b_{ij} \geq 0$, and $c_{ij} \geq 0$. We call a_{ij}, b_{ij}, and c_{ij} the linear cost coefficient, the quadratic cost coefficient, and the capacity, respectively, of arc (i, j). We created the test problems using two Fortran problem generators. The first is the public-domain generator NETGEN, written by Klingman, Napier, and Stutz [KNS74], which generates linear cost assignment/transportation/transshipment problems having a certain random structure. The second is the generator CHAINGEN, written by the second author, which generates transshipment problems having a chain structure as follows: starting with a chain through all the nodes, a user-specified number of forward arcs are added to each node (for example, if the user specifies 3 additional arcs per node then the arcs

TABLE 1

All problems are generated by NETGEN with linear cost coefficients in the range [1–100]*, total supply of* 10000*, one pure source and one pure sink, and arc capacities in the range* [100–500] *(except for problems* 22–24 *whose capacities are in the range* [1000–2000]*). For all problems, all arcs have quadratic cost coefficient in the range* [5–10]*. The run times for the codes* (*in seconds*) *were obtained on a Sun Sparc* 5 *with* 24*MB memory.*

Problem	N	A	NRELAX	MNRELAX	NE-RELAX-F	NE-RELAX-FV
1	200	1300	7.9	6.0	1.9	1.7
2	200	1500	7.5	6.3	2.1	1.7
3	200	2000	2.8	5.6	2.1	1.7
4	200	2200	20.4	10.6	2.4	2.0
5	200	2900	2.3	24.8	1.4	1.2
6	300	3150	7.3	22.1	2.7	1.6
7	300	4500	7.5	21.1	3.9	2.6
8	300	5155	48.3	26.7	3.8	2.0
9	300	6075	7.2	22.7	3.2	2.4
10	300	6300	4.4	31.5	2.7	3.3
11	400	1500	69.2	15.0	8.7	7.8
12	400	2250	17.6	17.2	4.9	4.3
13	400	3000	22.0	20.4	7.3	6.0
14	400	3750	13.2	24.3	3.0	1.8
15	400	4500	10.0	35.9	7.4	6.2
16	400	1306	85.1	25.4	8.6	8.4
17	400	2443	31.6	21.5	7.3	6.2
18	400	1416	7.5	9.0	0.9	0.9
19	400	2836	45.4	26.7	8.6	7.8
20	400	1382	79.9	17.7	9.9	8.4
21	400	2676	33.4	23.9	6.8	5.8
22	1000	3000	64.4	50.9	8.4	4.1
23	1000	5000	26.7	49.0	4.0	3.5
24	1000	10000	26.3	323.2	5.5	5.5

$(i, i+2)$, $(i, i+3)$, $(i, i+4)$ are added for each node i) and, for a user-specified percentage of nodes i, a reverse arc $(i, i-1)$ is also added. The graphs thus created have long diameters, and earlier tests on linear cost problems showed that the created problems are particularly difficult for all methods. As the above two generators create only linear cost problems, we modified the created problems as in [BHT87] so that a user-specified percent of the arcs generated a nonzero quadratic cost coefficient in a user-specified range.

Our tests were designed to study two key issues:

(a) the performance of the ϵ-relaxation methods relative to the relaxation methods and the dependence of this performance on network topology and problem ill conditioning,

(b) the sensitivity of the ϵ-relaxation methods to problem ill conditioning.

Ill-conditioned problems were created by assigning to some of the arcs smaller (but nonzero) quadratic cost coefficients compared to other arcs. When the arc cost functions have this structure, ill conditioning in the traditional sense of unconstrained nonlinear programming tends to occur.

We experimented with three sets of test problems: the first set comprises well-conditioned strictly convex quadratic cost problems generated using NETGEN (see Table 1); the second set comprises well-conditioned strictly convex quadratic cost problems generated using CHAINGEN (see Table 2); the third set comprises ill-conditioned strictly convex quadratic cost problems and mixed linear/quadratic cost

Table 2

All problems are generated by CHAINGEN with linear cost coefficients in the range [1–100]*, a supply of* 1000 *at node* 1 *and a demand of* 1000 *at node* N*, and arc capacities in the range* [100–1000]*. For all problems, all arcs have quadratic cost coefficient in the range* [5–10] *and half of the nodes have an additional reverse arc. The run times for the codes (in seconds) were obtained on a Sun Sparc* 5 *with* 24*MB memory. For these problems, the running times for NRELAX were excessively long even for* 5 *digits of accuracy and hence are not reported. For problem* 10*, MNRELAX did not terminate after the time shown, and this is indicated by the* > *in front of the time.*

Problem	N	Added Arcs	A	MNRELAX	NE-RELAX-F	NE-RELAX-FV
1	50	4	269	1.1	0.1	0.2
2	100	4	544	14.9	0.6	0.8
3	150	4	819	15.6	1.2	1.0
4	200	4	1094	33.0	2.1	2.1
5	250	4	1369	41.0	2.4	2.7
6	300	6	2235	93.9	4.6	5.2
7	350	6	2610	266.9	5.9	6.3
8	400	8	3772	1102.6	10.4	10.3
9	450	8	4247	2152.5	10.8	11.3
10	500	10	5705	>1300	17.7	17.5

Table 3

All problems are generated by NETGEN with linear cost coefficients in the range [1–100]*, total supply of* 1000*, one pure source and one pure sink, and arc capacities in the range* [100–300]*. For all problems, half of the arcs have quadratic cost coefficient in the range* [5–10] *and the remaining half have the small quadratic coefficient shown. Note that problems* 6 *and* 12 *are mixed cost problems. The runs for the codes* (*in seconds*) *were obtained on a Sun Sparc* 5 *with* 24*MB memory. For NRELAX, the numbers in parentheses indicate the number of significant digits of solution accuracy obtained by NRELAX in the running time shown.*

Problem	N	A	Small Quad Cost	NRELAX	MNRELAX	NE-RELAX-F	NE-RELAX-FV
1	200	1300	1	3.6	3.6	0.5	0.5
2	200	1300	0.1	20.9	4.3	0.6	0.9
3	200	1300	0.01	56.1	3.6	0.6	1.1
4	200	1300	0.001	(5)791.2	3.2	0.7	0.7
5	200	1300	0.0001	(5)1866.6	2.7	0.7	0.7
6	200	1300	0	-	-	0.6	0.8
7	400	4500	1	52.2	14.1	1.7	1.8
8	400	4500	0.1	53.4	11.2	1.8	2.0
9	400	4500	0.01	(5)80.5	13.7	2.3	2.6
10	400	4500	0.001	(5)710.7	15.0	2.6	2.6
11	400	4500	0.0001	(4)5753.4	13.5	3.6	3.1
12	400	4500	0	-	-	2.7	2.6

problems generated using NETGEN (see Table 3). The running time of the codes on these problems are shown in the last three to four columns of Tables 1–3. In all problems, the ϵ-relaxation codes were run to the point where they yielded higher or comparable solution accuracy than the relaxation codes. From the running times we can draw the following conclusions: first, the ϵ-relaxation codes NE-RELAX-F and NE-RELAX-FV have similar performance and both consistently outperform, by a factor of at least 3 and often much more, the relaxation codes NRELAX and MNRELAX on all test problems, independent of network topology and problem ill conditioning. In fact, on the CHAINGEN problems, the ϵ-relaxation codes outperform the relaxation codes by an order of magnitude or more. Other than the favorable complexity results that we obtained in this paper, we have no clear explanation of this phenomenon.

REFERENCES

[Ber79] D. P. Bertsekas, *A Distributed Algorithm for the Assignment Problems*, Laboratory for Information and Decision Systems Working Paper, M.I.T., Cambridge, MA, 1979.

[Ber85] D. P. Bertsekas, *A unified framework for minimum cost network flow problems,* Math. Programming, 32 (1985), pp. 125–145.

[Ber86a] D. P. Bertsekas, *Distributed Relaxation Methods for Linear Network Flow Problems*, in Proc. 25th IEEE Conference on Decision and Control, Athens, Greece, 1986, pp. 2101–2106.

[Ber86b] D. P. Bertsekas, *Distributed Asynchronous Relaxation Methods for Linear Network Flow Problems*, Laboratory for Information and Decision Systems report P-1606, M.I.T., Cambridge, MA, 1986.

[Ber91] D. P. Bertsekas, *Linear Network Optimization: Algorithms and Codes*, M.I.T. Press, Cambridge, MA, 1991.

[Ber92] D. P. Bertsekas, *An Auction/Sequential Shortest Path Algorithm for the Min Cost Flow Problem*, Laboratory for Information and Decision Systems report P-2146, M.I.T., Cambridge, MA, 1992.

[BeC93] D. P. Bertsekas and D. A. Castanon, *A generic auction algorithm for the minimum cost network flow problem*, Comput. Optim. Appl., 2 (1993), pp. 229–260.

[BCE95] D. P. Bertsekas, D. Castanon, J. Eckstein, and S. A. Zenios, *Parallel network optimization survey*, Handbooks Oper. Res. Management Sci., 7 (1995), pp. 331–339.

[BeE87] D. P. Bertsekas and J. Eckstein, *Distributed Asynchronous Relaxation Methods for Linear Network Flow Problems*, in Proc. International Federation of Automatic Control '87, Munich, Germany, July 1987.

[BeE88] D. P. Bertsekas and J. Eckstein, *Dual coordinate step methods for linear network flow problems*, Math. Programming, 42 (1988), pp. 203–243.

[BeE87] D. P. Bertsekas and D. El Baz, *Distributed asynchronous relaxation methods for convex network flow problems*, SIAM J. Control Optim., 25 (1987), pp. 74–85.

[BHT87] D. P. Bertsekas, P. A. Hosein, and P. Tseng, *Relaxation methods for network flow problems with convex arc costs*, SIAM J. Control Optim., 25 (1987), pp. 1219–1243.

[BeT88] D. P. Bertsekas and P. Tseng, *Relaxation methods for minimum cost ordinary and generalized network flow problems*, Oper. Res., 36 (1988), pp. 93–114.

[BeT94] D. P. Bertsekas and P. Tseng, *RELAX-IV: A Faster Version of the RELAX Code for Solving Minimum Cost Flow Problems*, Laboratory for Information and Decision Systems report P-2276, M.I.T., Cambridge, MA, 1994.

[BeT89] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice–Hall, Englewood Cliffs, NJ, 1989.

[BlJ92] R. G. Bland and D. L. Jensen, *On the computational behavior of a polynomial-time network flow algorithm*, Math. Programming, 54 (1992), pp. 1–39.

[DMZ95] R. De Leone, R. R. Meyer, and A. Zakarian, *An ϵ-Relaxation Algorithm for Convex Network Flow Problems*, Computer Sciences Department Technical report, University of Wisconsin, Madison, WI, 1995.

[EdK72] J. Edmonds and R. M. Karp, *Theoretical improvements in algorithmic efficiency for network flow problems*, J. Assoc. Comput. Mach., 19 (1972), pp. 248–264.

[FoF62] L. R. Ford, Jr., and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, NJ, 1962.

[GoT90] A. V. Goldberg and R. E. Tarjan, *Solving minimum cost flow problems by successive approximation*, Math. Oper. Res., 15 (1990), pp. 430–466.

[Gol87] A. V. Goldberg, *Efficient Graph Algorithms for Sequential and Parallel Computers*, Laboratory for Computer Science Technical report TR-374, M.I.T., Cambridge, MA, 1987.

[Hag92] W. W. Hager, *The dual active set algorithm*, in Advances in Optimization and Parallel Computing, P. M. Pardalos, ed., North–Holland, Amsterdam, the Netherlands, 1992, pp. 137–142.

[HaH93] W. W. Hager and D. W. Hearn, *Application of the dual active set algorithm to quadratic network optimization*, Comput. Optim. Appl., 1 (1993), pp. 349–373.

[KaM84] P. V. Kamesam and R. R. Meyer, *Multipoint methods for separable nonlinear networks*, Math. Programming Study, 22 (1984), pp. 185–205.

[KaM93] A. V. Karzanov and S. T. McCormick, *Polynomial methods for separable convex optimization in unimodular linear spaces with applications to circulations and co-circulations in network*, SIAM J. Comput., 26 (1997), pp. 1248–1278.

[KNS74] D. Klingman, A. Napier, and J. Stutz, *NETGEN - A program for generating large scale (un) capacitated assignment, transportation, and minimum cost flow network problems*, Management Sci., 20 (1974), pp. 814–822.

[LiZ91] X. Li and S. A. Zenios, *Data Parallel Solutions of Min-Cost Network Flow Problems Using ϵ-Relaxations*, Department of Decision Sciences report 1991-05-20, University of Pennsylvania, Philadelphia, PA, 1991.

[Mey79] R. R. Meyer, *Two-segment separable programming*, Management Sci., 25 (1979), pp. 285–295.

[NiZ93] S. S. Nielsen and S. A. Zenios, *On the massively parallel solution of linear network flow problems*, in Network Flow and Matching: First DIMACS Implementation Challenge, D. Johnson and C. McGeoch, eds., American Mathematical Society, Providence, RI, 1993, pp. 349–369.

[Pol94] L. C. Polymenakos, *Parallel shortest path auction algorithms*, Parallel Comput., 20 (1994), pp. 1221–1247.

[Pol95] L. C. Polymenakos, *ϵ-Relaxation and Auction Algorithms for the Convex Cost Network Flow Problem*, Electrical Engineering and Computer Science Department Ph.D. thesis, M.I.T., Cambridge, MA, 1995.

[Roc80] H. Röck, *Scaling techniques for minimal cost network flows*, in Discrete Structures and Algorithms, U. Pape and Carl Hanser, eds., München, Germany, 1980, pp. 181–191.

[Roc70] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[Roc84] R. T. Rockafellar, *Network Flows and Monotropic Programming*, Wiley-Interscience, New York, NY, 1984.

[Tse86] P. Tseng, *Relaxation Methods for Monotropic Programming Problems*, Operations Research Center Ph.D. thesis, M.I.T., Cambridge, MA, 1986.

[TBT90] P. Tseng, D. P. Bertsekas, and J. N. Tsitsiklis, *Partially asynchronous, parallel algorithms for network flow and other problems*, SIAM J. Control Optim., 28 (1990), pp. 678–710.

[Ven91] J. A. Ventura, *Computational development of a lagrangian dual approach for quadratic networks*, Networks, 21 (1991), pp. 469–485.

[Wei74] A. Weintraub, *A primal algorithm to solve network flow problems with convex costs*, Management Sci., 21 (1974), pp. 87–97.

SIAM J. Optim.
Vol. 7, No. 3, pp. 871–887, August 1997

BOX CONSTRAINED QUADRATIC PROGRAMMING WITH PROPORTIONING AND PROJECTIONS*

ZDENĚK DOSTÁL†

Abstract. Two new closely related concepts are introduced that depend on a positive constant Γ. An iteration is proportional if the norm of violation of the Kuhn–Tucker conditions at active variables does not excessively exceed the norm of the part of the gradient that corresponds to free variables, while a progressive direction determines a descent direction that enables the released variables to move far enough from the boundary in a step called proportioning. An algorithm that uses the conjugate gradient method to explore the face of the region defined by the current iterate until a disproportional iteration is generated is proposed. It then changes the face by means of the progressive direction. It is proved that for strictly convex problems, the proportioning is a spacer iteration so that the algorithm converges to the solution. If the solution is nondegenerate then the algorithm finds the solution in a finite number of steps. Moreover, a simple lower bound on Γ is given to ensure finite termination even for problems with degenerate solutions. The theory covers a class of algorithms, allowing many constraints to be added or dropped at a time and accepting approximate solutions of auxiliary problems. Preliminary numerical results are promising.

Key words. quadratic programming, conjugate gradients, inexact subproblem solution, projected search

AMS subject classifications. 65K05, 90C20

PII. S1052623494266250

1. Introduction.

We shall be concerned with the problem to find

$$\min_{x\in\Omega} f(x) \tag{1.1}$$

with $\Omega = \{x \in \mathbb{R}^n \ : \ l \le x \le u\}$, $f(x) = \frac{1}{2}x^T Ax - x^T b$, l, u, and b given column n-vectors, and A an $n \times n$ symmetric positive definite matrix. We suppose $l < u$ but admit $l_i = -\infty$ or $u_i = \infty$.

Applications that lead to the problem (1.1) include contact problems in linear elasticity (e.g., Klarbring [14]), obstacle problems (Cimatti [4], Moré and Toraldo [16], Glowinski [7]), and problems of optimal design (in Haslinger and Neittaanmäki [11]). It may be advantageous to reduce some problems with more general linear constraints to the problem (1.1) by duality (Dostál [5]).

The algorithms for the solution of (1.1) may be classified in two groups. The algorithms that are known to terminate in the solution $\overline{x}$ of (1.1) in a finite number of steps due to arguments of combinatorial nature are called finite (Júdice and Pires [13]). They include the Polyak algorithm [18], which reduces the solution of (1.1) to the conjugate gradient minimization [12] of f on a finite sequence of auxiliary subspaces of $\mathbb{R}^n$ called faces, and its modifications. The first modification aims at more efficient minimization on faces by adapting a suitable preconditioning strategy; we refer to O'Leary [17] for both general idea and examples. The second modification concerns precision of the solution of the auxiliary minimization problems. Encouraged

* Received by the editors April 18, 1994; accepted for publication (in revised form) November 3, 1995. This research was supported by Czech Academy of Science grant 31656 and by Czech Grant Agency (GA ČR) grant 101/94/1853/.

http://www.siam.org/journals/siopt/7-3/26625.html

† Department of Applied Mathematics, Technical University Ostrava and Department of Mathematical Modelling, Institute of Geonics of Czech Academy of Sciences, Ostrava, Czech Republic (zdenek.dostal@vsb.cz).

by experiments of O'Leary, who reduced the number of iterations to about a half with an algorithm in which the accuracy of the conjugate gradient minimization was refined during the course of iterations, the author [6] has recently presented a modification of the Polyak algorithm that accepts approximate solutions of auxiliary subproblems and preserves the finite termination property of the original algorithm even for degenerate problems. Finally, the third modification concerns the definition of a new face. An obvious drawback of the original Polyak algorithm, which is typically unable to add more than one constraint to a current working face, has led to the development of the gradient projection methods (Yang and Tolle [19]).

The other class of algorithms arises from iterative procedures such as multigrid methods (Hackbusch and Mittelmann [10]), the Newton method (Zhang, Tapia, and Potra [20]), or a combination of the relaxation and conjugate gradients (Kočvara and Zowe [15]). On the basis of results of Calamai and Moré [3], Moré and Toraldo [16] proposed an algorithm that also exploits the conjugate gradients, but its convergence is driven by gradient projections with steplength satisfying the sufficient decrease condition [3]. For nondegenerate problems, their algorithm also terminates in the solution in a finite number of steps.

The algorithms that we propose here depend on a positive constant Γ in the way that they may be considered finite for large Γ and iterative for small Γ. The algorithms are described by means of two new concepts depending on Γ. An iteration is proportional if the norm of violation of the Kuhn–Tucker conditions at active variables does not excessively exceed the norm of the part of the gradient that corresponds to free variables, while a progressive direction determines the decrease direction that enables the released variables to move far enough from the boundary. We use the progressive directions to move from disproportional iterations to proportional ones in a process that we call proportioning.

Our new concepts arise from results of section 3, where we show that in certain stages of the computation it is possible to extract information about the binding set of the solution of the auxiliary problem from the gradient of the current iterate. These results are then used to develop algorithms with controlled precision for the solution of auxiliary problems.

The main result of section 4 is Theorem 4.2, which implies that the proportioning is a spacer iteration (Bertsekas [2]). Since the proportioning is easy to compute, it may be considered an attractive alternative to the gradient projections of Calamai and Moré [3] or Moré and Toraldo [16], which require a possibly expensive projected search. Our convergence result for algorithms driven by proportioning is analogous to the result of Calamai and Moré [3, Theorem 5.2] for the gradient projections.

The power of our new concepts is demonstrated in section 5, where a new class of algorithms is defined and its finite termination properties are studied. Apart from a result on the finite termination property for nondegenerate problems that is analogous to that of Moré and Toraldo [16, Theorem 5.2], we prove that the iterations end in the solution of (1.1) in a finite number of steps, even for degenerate problems, provided Γ is greater or equal to the critical release coefficient $\rho(A)$, and we give a simple upper bound for $\rho(A)$ in terms of the spectral condition number $\kappa(A)$ of A. Our class of algorithms includes also the algorithm presented at [6] so that the theoretical results of section 5 extend applicability of Algorithm 5.1 of [6] to all $\Gamma_E > 0$.

An implementation of our algorithm with conjugate gradients is proposed in section 6 and further specified in section 7. In particular, we describe how to incorporate projections into the algorithm so that it can drop and add many constraints each time the active set is changed.

We have implemented our algorithm in Matlab and carried out some experiments and comparisons that are reported in sections 8 and 9.

2. Notations and preliminaries. Throughout the whole paper, we shall use the notation of the introduction.

It is well known that a solution to the problem (1.1) always exists and is necessarily unique. The solution $\overline{x}$ is fully determined by the Kuhn–Tucker conditions [1]. Thus $x \in \Omega$ is the solution of (1.1) iff for $i = 1, \dots, n$

$$r_i(x) \geq 0 \text{ for } x_i = l_i,\ r_i(x) \leq 0 \text{ for } x_i = u_i, \tag{2.1}$$

$$r_i(x) = 0 \text{ for } l_i \ < \ x_i \ < \ u_i, \tag{2.2}$$

where $r(x) = Ax - b = \nabla f(x)$. The conditions (2.1) will be called the *Kuhn–Tucker contact conditions*.

Let $\mathcal{N}$ denote the set of all indices so that

$$\mathcal{N} = \{1, 2, \dots, n\}.$$

The set of all indices for which the variables x_i are at their bounds is called an *active set* of x. We shall denote it by $\mathcal{A}(x)$ so that

$$\mathcal{A}(x) = \{i \in \mathcal{N} \ : \ x_i = l_i \text{ or } x_i = u_i\}.$$

Its subset

$$\mathcal{B}(x) = \{i \in \mathcal{N} \ : \ x_i = l_i \text{ and } r_i(x) \geq 0 \text{ or } x_i = u_i \text{ and } r_i(x) \leq 0\}$$

and complement

$$\mathcal{F}(x) = \{i \in \mathcal{N} \ : \ l_i \ < \ x_i \ < \ u_i\}$$

are called a *binding set* of x and a *free set* of x, respectively.

To enable an alternative reference to the Kuhn–Tucker conditions, we shall introduce a notation for the parts of $r(x)$ that are defined by

$$\begin{aligned}
\varphi_i(x) &= r_i(x) && \text{for} && i \in \mathcal{F}(x),\\
\varphi_i(x) &= 0 && \text{for} && i \in \mathcal{A}(x),\\
\beta_i(x) &= r_i^-(x) && \text{for} && x_i = l_i,\\
\beta_i(x) &= r_i^+(x) && \text{for} && x_i = u_i,\\
\beta_i(x) &= 0 && \text{for} && i \in \mathcal{F}(x),
\end{aligned}$$

where we have used the notation

$$x_i^+ = \max\{x_i, 0\} \text{ and } x_i^- = \min\{x_i, 0\} \text{ for } x \in \mathbb{R}^m.$$

The vectors $\varphi(x)$ and $\beta(x)$ will be called *a free gradient* and an *unbalanced contact gradient*, respectively. Thus, the Kuhn–Tucker contact conditions (2.1) are satisfied at x iff $\beta(x) = o$, and the Kuhn–Tucker conditions (2.1) and (2.2) are satisfied iff the *projected gradient* $\nu(x) = \varphi(x) + \beta(x)$ is reduced to zero.

The Euclidean norm and the l_∞-norm of any $x \in \mathbb{R}^m$ will be denoted by $\|x\|$ and $\|x\|_\infty$, respectively. Analogous notation will be used for induced matrix norms.

For any decomposition I, J of the set of indices $\mathcal{N}$ and for any $x \in \mathbb{R}^n$, let us denote by x_I and x_J the parts of x whose indices belong to I and J, respectively. Corresponding to this decomposition of $\mathcal{N}$, we also partition and rearrange the vectors $r = r(x)$ and b and the matrix A. With this notation,

$$\begin{pmatrix} r_I \\ r_J \end{pmatrix} = \begin{pmatrix} A_{II} & A_{IJ} \\ A_{JI} & A_{JJ} \end{pmatrix} \begin{pmatrix} x_I \\ x_J \end{pmatrix} - \begin{pmatrix} b_I \\ b_J \end{pmatrix}, \tag{2.3}$$

and for any $y \in \mathbb{R}^n$, the minimization of $f(x)$ on the face

$$\mathcal{W}(I, y) = \{x \in \mathbb{R}^n \ : \ x_i = y_i \text{ for } i \in I\}$$

amounts to unconstrained minimization of

$$f_J(x) = \frac{1}{2} x_J^T A_{JJ} x_J - x_J^T (b_J - A_{JI} y_I).$$

To simplify manipulation with x_I in our algorithms, we shall use notation P_I for the diagonal matrix with diagonal entries equal to 1 or 0 for $i \in I$ and $i \notin I$, respectively.

Finally, we shall denote by P_Ω the projection from $\mathbb{R}^n$ to Ω.

3. Release criteria and release coefficients. The purpose of this section is to develop useful tests for the control of precision of auxiliary problems. To this end, it is important to recognize the indices that belong to the current active set but do not belong to the binding set of the solution of the auxiliary problem.

LEMMA 3.1. *Let I, J denote a decomposition of the set of indices $\mathcal{N}$ such that $J \neq \theta$. Let $x \in \mathbb{R}^n$ and let $\bar{x}$ minimize $f(y)$ on the face $\mathcal{W}(I, x)$.*

Then for any $i \in I$,

$$r_i > A_{iJ} A_{JJ}^{-1} r_J \textit{ implies } \bar{r}_i > 0, \tag{3.1a}$$

$$r_i < A_{iJ} A_{JJ}^{-1} r_J \textit{ implies } \bar{r}_i < 0, \tag{3.1b}$$

where $r = r(x)$ and $\bar{r} = r(\bar{x})$.

Proof. After rearranging the indices, we can write the formulas for r and $\bar{r}$ in the form (2.3). Observing that $\bar{r}_J = o$ and $x_I = \bar{x}_I$, we get

$$\begin{pmatrix} r_I - \bar{r}_I \\ r_J \end{pmatrix} = \begin{pmatrix} A_{II} & A_{IJ} \\ A_{JI} & A_{JJ} \end{pmatrix} \begin{pmatrix} o \\ x_J - \bar{x}_J \end{pmatrix}, \tag{3.2}$$

and after simple computations,

$$r_I - \bar{r}_I = A_{IJ} A_{JJ}^{-1} r_J.$$

We have used the assumption that A is positive definite so that A_{JJ}^{-1} exists.

Now let us decompose $\bar{r}_I = \bar{r}_I^+ + r_I^-$. We get

$$\bar{r}_I^+ = r_I - A_{IJ} A_{JJ}^{-1} r_J - r_I^- \geq r_I - A_{IJ} A_{JJ}^{-1} r_J, \tag{3.3a}$$

$$\bar{r}_I^- = r_I - A_{IJ}A_{JJ}^{-1}r_J - r_I^+ \le r_I - A_{IJ}A_{JJ}^{-1}r_J, \tag{3.3b}$$

where all the inequalities should be read by coordinates. The statement of Lemma 3.1 is just an interpretation of (3.3). □

Lemma 3.1 gives the condition in terms of $r_i(x)$ that fully determines the sign of $r_i(\bar{x})$ for $i \in \mathcal{A}(x)$ and indicates that the sign of r_i is that of $\bar{r}_i$ provided $\|r_J\|$ is small. Indeed, under the notation and assumptions of Lemma 3.1 and by using the well-known relations between the vector norms and the interlacing properties of the spectra of symmetric positive definite matrices, we get

$$|A_{iJ}A_{JJ}^{-1}r_J| \le \|A_{IJ}A_{JJ}^{-1}r_J\|_\infty \le \|A_{IJ}A_{JJ}^{-1}r_J\| \le \|A\|\|A_{JJ}^{-1}\|\|r_J\| \le \kappa(A)\|r_J\|, \tag{3.4}$$

so that for $i \in I$,

$$\begin{aligned} r_i - A_{iJ}A_{JJ}^{-1}r_J &\ge r_i - \kappa(A)\|r_J\|, \\ r_i - A_{iJ}A_{JJ}^{-1}r_J &\le r_i + \kappa(A)\|r_J\|, \end{aligned}$$

and by Lemma 3.1

$$r_i > \kappa(A)\|r_J\| \text{ implies } \bar{r}_i > 0, \tag{3.5a}$$

$$r_i < -\kappa(A)\|r_J\| \text{ implies } \bar{r}_i < 0. \tag{3.5b}$$

Motivated by (3.5), we may introduce *release criteria* in the form

$$\|\beta(x)\|_\infty > \Gamma\|\varphi(x)\|$$

with an arbitrary nonnegative *release coefficient* Γ. The release coefficients that are large enough to yield information on $\beta(\bar{x})$ will be called determining. More formally, for a given positive definite matrix A, the nonnegative Γ is a *determining release coefficient* for A iff for any $x \in \mathbb{R}^n$

$$\|\beta(x)\|_\infty > \Gamma\|\varphi(x)\| \text{ implies } \beta(\bar{x}) \ne o, \tag{3.6}$$

where $\bar{x}$ minimizes $f(y)$ subject to $y \in \mathcal{W}(\mathcal{A}(x), x)$.

Let use define the *critical release coefficient* $\rho(A)$ as the infimum of the set of all determining release coefficients for A. Using (3.1) and (3.5), it is easy to check that $\rho(A) = 0$ iff A is diagonal and that $\rho(A) \le \kappa(A)$ for any positive definite matrix. The following improvement of the latter estimate is based on [6].

THEOREM 3.2. *Let $x \in \mathbb{R}^n$ such that*

$$\|\beta(x)\|_\infty > \kappa(A)^{1/2}\|\varphi(x)\|, \tag{3.7}$$

and let $\bar{x}$ minimize $f(y)$ on the face $\mathcal{W}(\mathcal{A}(x), x)$. Then $\beta(\bar{x}) \ne o$.

Proof. Let use denote $\Omega = \mathcal{W}(\mathcal{A}(x), x)$. By Theorem 1 of [6] and under the assumptions of Theorem 3.2, the vector

$$y = x - \|A\|^{-1}\beta(x)$$

satisfies

$$f(y) < f(\bar{x}),$$

so that

$$0 > f(y) - f(\bar{x}) = (A\bar{x} - b)^T(y - \bar{x}) + \frac{1}{2}(y - \bar{x})A(y - \bar{x}) > r(\bar{x})^T(y - \bar{x}). \tag{3.8}$$

Observing that $\bar{x}_i = x_i$ for $i \in \mathcal{A}(x)$ and that $r_i(\bar{x}) = 0$ for $i \in \mathcal{F}(x)$, we get

$$r(\bar{x})^T(y - \bar{x}) = r(\bar{x})^T(y - x) = -\|A\|^{-1} r(\bar{x})^T \beta(x) \geq -\|A\|^{-1}\beta(\bar{x})^T\beta(x). \tag{3.9}$$

However, (3.8) and (3.9) imply $\beta(\bar{x}) \neq o$. □

COROLLARY 3.3. *For any positive definite matrix A, the critical release coefficient $\rho(A)$ for A satisfies*

$$\rho(A) \leq \kappa(A)^{1/2}. \tag{3.10}$$

4. Proportioning and convergence. To simplify our exposition, we shall start with two closely related definitions that are motivated by the discussion of the previous section. They use a release coefficient Γ and the gap g between l and u defined by

$$g = \min\{u_i - l_i \ : \ i \in \mathcal{N}\}. \tag{4.1}$$

We assume Γ to be fixed throughout the whole section.

A vector $x \in \mathbb{R}^n$ is *proportional* (with Γ) iff

$$\|\beta(x)\|_\infty \leq \Gamma\|\varphi(x)\| \tag{4.2}$$

and *disproportional* otherwise.

A nonzero vector d is called a *progressive direction* at x iff

$$\|\beta(x)\|_\infty = \|d_I\|_\infty \text{ for } I = \mathcal{A}(x), \tag{4.3a}$$

$$r^T d \geq \|d\|^2 \text{ and } x - (g/\|d\|_\infty)d \in \Omega. \tag{4.3b}$$

The conditions (4.3a) and $r^T d \geq \|d\|^2$ are obviously satisfied by the projected gradient of [3, 16] and ensure that $-d$ is a descent direction for f that may be used to generate an iteration with a reduced active set, while the last condition ensures that it is possible to move far enough. If x is disproportional then the most simple choice of a progressive direction is $d = \beta(x)$. We shall use progressive directions to move from disproportional iterations to proportional ones in a step called *proportioning.*

If $\Gamma > 0$ and a vector $x \in \Omega$ is disproportional, then it is easy to check that

$$\|\beta(x)\|_\infty \leq \|d\|_\infty \text{ and } \|\varphi(x)\| < \Gamma^{-1}\|d\|_\infty \tag{4.4}$$

for any progressive direction d at x.

LEMMA 4.1. *Let x and d denote given n-vectors, $d \neq o$, $r = r(x)$, and $\delta \geq 0$. If*

$$r^T d \geq \|d\|^2 \textit{ and } \min\{\alpha^{cg}, \delta/\|d\|_\infty\} \leq \alpha \leq \alpha^{cg}, \tag{4.5a}$$

$$\alpha^{cg} = r^T d / d^T A d, \tag{4.5b}$$

then

$$f(x) - f(x - \alpha d) \geq \frac{1}{2}\min\{\|A\|^{-1}\|d\|^2, \ \delta\|d\|\}. \tag{4.6}$$

Proof. Simple computations show that the function

$$\Delta(\xi) = f(x) - f(x - \xi d) = \xi r^T d - \frac{1}{2}\xi^2 d^T A d$$

is increasing for $\xi \in [0,\ \alpha^{cg}]$ and that

$$\Delta(\alpha^{cg}) = \frac{1}{2}(r^T d)^2 / d^T A d \geq \frac{1}{2}\|A\|^{-1}\|d\|^2$$

for any d which satisfies $r^T d \geq \|d\|^2$.

Let $\lambda_1, \dots, \lambda_n$ denote the eigenvalues of A. For $0 \leq \xi \leq \|A\|^{-1}$, the eigenvalues θ_i of $I - \frac{1}{2}\xi A$ satisfy

$$\theta_i = 1 - \frac{1}{2}\xi\lambda_i \geq 1 - \frac{1}{2}\|A\|^{-1}\lambda_i \geq \frac{1}{2},$$

so that

$$\Delta(\xi) \geq \xi d^T \left(I - \frac{1}{2}\xi A\right) d \geq \frac{1}{2}\xi\|d\|^2 \text{ for } \xi \leq \|A\|^{-1}.$$

Applying this inequality to

$$\mu = \min\{\|A\|^{-1},\ \delta/\|d\|\},$$

we get for $\delta/\|d\|_\infty \leq \alpha^{cg}$

$$\Delta(\delta/\|d\|_\infty) \geq \Delta(\delta/\|d\|) \geq \Delta(\mu) \geq \frac{1}{2}\mu\|d\|^2.$$

Substituting for μ, we get for $\delta/\|d\|_\infty \leq \alpha^{cg}$

$$\Delta(\delta/\|d\|_\infty) \geq \frac{1}{2}\min\{\|A\|^{-1}\|d\|^2,\ \delta\|d\|\}. \qquad \square$$

THEOREM 4.2. *Let $\Gamma > 0$, let $0 < \delta < g$ where g is the gap* (4.1) *between l and u, and let $\{x^k\}$ denote an infinite sequence of $x^k \in \Omega$ that satisfies*

$$f(x^{k+1}) \leq f(x^k). \tag{4.7}$$

Let K_p denote the set of all indices such that x^j is disproportional for each $j \in K_p$ and let there be a progressive direction d^j at x^j so that $x^{j+1} = x^j - \alpha_j d^j$ with

$$\min\{\alpha_j^{cg},\ \delta/\|d^j\|_\infty\} \leq \alpha_j \leq \alpha_j^{cg},\ \alpha_j^{cg} = r(x^j)^T d^j/(d^j)^T A d^j. \tag{4.8}$$

If K_p is infinite, then $\{x^k\}$ converges to the solution $\bar{x}$ of (1.1).

Proof. Since the set

$$S = \{x \in \Omega\ :\ f(x) \leq f(x^0)\}$$

is compact, there is a limit point $\bar{x}$ of the sequence $\{x^j\ :\ j \in K_p\} \subset S$ and a subset K_p^0 of K_p such that $\{x^j\ :\ j \in K_p^0\}$ converges to $\bar{x}$. The function f being continuous, it follows that $\{f(x^j)\ :\ j \in K_p^0\}$ converges to $f(\bar{x})$ and, using the definition of α_j and Lemma 4.1, we conclude that $\{\|d^j\|\ :\ j \in K_p^0\}$ converges to zero.

To show that $\bar{x}$ satisfies the Kuhn–Tucker conditions, let us first suppose that for some fixed $i \in \mathcal{N}$,

$$l_i < \bar{x}_i < u_i.$$

Then for sufficiently large $j \in K_p^0$,

$$l_i < x_i^j < u_i$$

and since by (4.4) for such i and j

$$|r_i(x^j)| \leq \|\varphi(x^j)\| \ < \Gamma^{-1}\|d^j\|_\infty \leq \Gamma^{-1}\|d^j\|,$$

$\{|r_i^j| \ : \ j \in K_p^0\}$ converges to zero and $r_i(\bar{x}) = 0$.

If $l_i = \bar{x}_i$, then we shall distinguish two cases. If there is an infinite subsequence of $\{x^j \ : \ j \in K_p^0\}$ such that $l_i < x_i^j$, we shall show as above that $r_i(x^j)$ converges to zero. If there is no such subsequence, then there is an infinite subset K_p^1 of K_p^0 such that

$$l_i = x_i^j \text{ for } j \in K_p^1.$$

Since by (4.4) in this case

$$|r_i(x^j)^-| \leq \|d^j\|_\infty,$$

we conclude that $r_i(\bar{x})^- = 0$.

In the same way, we can check that if $u_i = \bar{x}_i$, then $r_i(\bar{x})^+ = 0$. Summing up all three cases, we conclude that $\bar{x}$ is the solution of (1.1) that satisfies the Kuhn–Tucker conditions (2.1) and (2.2). In particular, it follows that

$$(A\bar{x} - b)^T(x - \bar{x}) \geq 0 \text{ for any } x \in \Omega. \tag{4.9}$$

Now for each integer k, let

$$M(k) = \min\{s \in K_p^0 \ : \ s \geq k\} \text{ and } m(k) = \max\{x \in K_p^0 \ : \ s \leq k\}.$$

With this notation,

$$f(x^{M(k)}) - f(\bar{x}) \geq f(x^k) - f(\bar{x}) \geq f(x^{m(k)}) - f(\bar{x}),$$

so that

$$f(\bar{x}) = \ \inf\ \{f(x^k)\}. \tag{4.10}$$

Using (4.9), we obtain

$$f(x^k) - f(\bar{x}) = (A\bar{x} - b)^T(x^k - \bar{x}) + \frac{1}{2}(x^k - \bar{x})^T A(x^k - \bar{x}) \geq \frac{1}{2}\lambda_{\min}\|x^k - \bar{x}\|^2,$$

where $\lambda_{\min}$ is the least eigenvalue of A. Since $\{f(x^k)\}$ converges to $f(\bar{x})$ by (4.10), it follows that $\{x^k\}$ converges to the solution $\bar{x}$ of (1.1). □

5. Proportioning and the finite termination property. Using observations of section 4, we shall now present a class of algorithms driven by proportioning that reach the solution $\bar{x}$ of (1.1) in a finite number of steps even for degenerate problems. To deal with the latter, let us decompose the active set $\mathcal{A}(\bar{x})$ of the solution $\bar{x}$ of (1.1) into

$$\mathcal{A}^0 = \{i \in \mathcal{A}(\bar{x}) \ : \ r_i(\bar{x}) = 0\}$$

and

$$\mathcal{A}^1 = \{i \in \mathcal{A}(\bar{x}) \ : \ r_i(\bar{x}) \neq 0\},$$

so that the degenerate problems are those with $\mathcal{A}^0 \neq \theta$.

ALGORITHM 5.1 (GENERAL PROPORTIONING SCHEME). *Let $x^0 \in \Omega$, $0 < \delta < g$, and $\Gamma > 0$ be given. For $k \geq 0$, choose x^{k+1} by the following rules*:

(a) *If x^k is disproportional (with respect to Γ), set*

$$x^{k+1} = x^k - \alpha_k d^k \tag{5.1}$$

with any progressive direction d^k and α_k defined by (4.8) *so that*

$$\mathcal{A}(x^k) \supsetneq \mathcal{A}(x^{k+1}). \tag{5.2}$$

(b) *If x^k is proportional (with respect to Γ), choose $x^{k+1} \in \Omega$ so that*

$$f(x^{k+1}) \leq f(x^k) \text{ and } \mathcal{A}(x^k) \subset \mathcal{A}(x^{k+1}) \tag{5.3}$$

and x^{k+1} satisfies at least one of the conditions

$$f(x^{k+1}) = \min\{f(x) \ : \ x \in \mathcal{W}(\mathcal{A}(x^k),\ x^k)\}, \tag{5.4}$$

$$\mathcal{A}(x^k) \subsetneq \mathcal{A}(x^{k+1}), \tag{5.5}$$

or x^{k+1} is disproportional.

LEMMA 5.2. *Let $\{x^k\}$ denote an infinite sequence generated by Algorithm* 5.1 *and $k \geq 0$.*

(i) *If x^{k+1} is generated by* (5.4), *then x^{k+1} is proportional iff the Kuhn–Tucker conditions* (2.1) *and* (2.2) *are satisfied at x^{k+1}.*

(ii) *If x^{k+1} is generated by* (5.4) *and x^{k+1} is proportional, then x^{k+1} is the solution $\bar{x}$ of* (1.1) *and*

$$\bar{x} = x^{k+1} = x^{k+2} = \cdots. \tag{5.6}$$

(iii) *If*

$$\mathcal{A}(x^k) = \mathcal{A}(x^{k+1}) = \mathcal{A}(x^{k+2}), \tag{5.7}$$

then $\bar{x} = x^{k+1} = x^{k+2} = \cdots$.

(iv) *The sequence $\{f(x^k)\}$ is nonincreasing.*

Proof. (i) If x^{k+1} is generated by (5.4), then

$$\|\varphi(x^{k+1})\| = 0. \tag{5.8}$$

However, (5.8) implies $\nu(x^{k+1}) = \beta(x^{k+1})$, so that

$$\|\beta(x^{k+1})\|_\infty \le \Gamma\|\varphi(x^{k+1})\| \text{ iff } \nu(x^{k+1}) = o.$$

(ii) If x^{k+1} is generated by (5.4) and x^{k+1} is proportional, then

$$0 \le \|\beta(x^{k+1})\|_\infty \le \Gamma\|\varphi(x^{k+1})\| = 0,$$

so that $\nu(x^{k+1}) = o$ and $x^{k+1} = \bar{x}$. Moreover, it follows that x^{k+2} also satisfies the assumptions of (ii), so that $\bar{x} = x^{k+2}$, etc.

(iii) Let us assume that $\mathcal{A}(x^k) = \mathcal{A}(x^{k+1}) = \mathcal{A}(x^{k+2})$. Comparing this assumption with (5.2), (5.3), and (5.5), we get that x^{k+1} is proportional and that x^{k+1} is generated by (5.4). Thus the assumptions of (ii) are satisfied and (5.6) follows.

(iv) The statement is obvious. □

THEOREM 5.3. *Let $\{x^k\}$ denote an infinite sequence generated by* Algorithm 5.1 *with given $x^0 \in \Omega$ and $\Gamma > 0$.*

(i) *$\{x^k\}$ converges to the solution $\bar{x}$ of* (1.1).

(ii) *If the problem* (1.1) *is not degenerate, then there is k such that $\bar{x} = x^k$.*

(iii) *If $\Gamma \ge \rho(A)$, then there is k such that $\bar{x} = x^k$.*

(iv) *If $\Gamma \ge \kappa(A)^{1/2}$, then there is k such that $\bar{x} = x^k$.*

Proof. (i) Since the number of elements of $\mathcal{A}(x^k)$ cannot exceed the dimension n of the problem (1.1), it follows that either there is k such that

$$\mathcal{A}(x^k) = \mathcal{A}(x^{k+1}) = \mathcal{A}(x^{k+2}) \tag{5.9}$$

or there is an infinite set of indices K_p such that

$$\mathcal{A}(x^k) \supsetneq \mathcal{A}(x^{k+1}). \tag{5.10}$$

In the first case, we can use Lemma 5.2 (iii) to get

$$\bar{x} = x^{k+1} = x^{k+2} = \cdots$$

so that $\{x^k\}$ trivially converges to $\bar{x}$. In the other case, it is enough to observe that if k satisfies (5.10), then x^{k+1} is generated by (5.1). Since $f(x^{k+1}) \le f(x^{k+1})$, the assumptions of Theorem 4.2 are satisfied and we conclude that $\{x^k\}$ converges to $\bar{x}$ for any $\Gamma > 0$.

(ii) Let us suppose that $\mathcal{A}^0 = \theta$. Since $\{x^k\}$ converges to $\bar{x}$, there is an n_0 such that for $k \ge n_0$,

$$l_i < x_i^k < u_i \text{ for } i \in \mathcal{F}(\bar{x}) \tag{5.11}$$

and

$$r_i(x^k) > \frac{\mu}{2} \text{ and } x_i^k < u_i \text{ or } r_i(x^k) < \frac{\mu}{2} \text{ and } l_i < x_i^k \text{ for } i \in \mathcal{A}^1, \tag{5.12}$$

where

$$\mu = \min\{|r_i(\bar{x})| \ : \ i \in \mathcal{A}^1\}.$$

Since $\mathcal{A}^0$ is empty, (5.11) and (5.12) imply that $\beta(x^k) = o$ for $k \ge n_0$ so that x^k is proportional for $k \ge n_0$. Hence, by the definition of Algorithm 5.1,

$$\mathcal{A}(x^k) \subset \mathcal{A}(x^{k+1}) \subset \cdots \text{ for } k \ge n_0.$$

We complete the proof by using the dimension argument and Lemma 5.2 (iii).

(iii) Now suppose that $\Gamma \geq \rho(A)$. Since $\{x^k\}$ converges to $\bar{x}$ and the mapping $x \mapsto r(x)$ is continuous, for any $\varepsilon > 0$ there is an n_0 such that for $k \geq n_0$,

$$|r_i(x^k)| < \varepsilon \text{ for } i \in \mathcal{A}^0 \cup \mathcal{F}(\bar{x}) \tag{5.13}$$

and both (5.11) and (5.12) are satisfied. For sufficiently small ε and $k \geq n_0$, it follows from (5.11), (5.12), and (5.13) that x^k is proportional whenever

$$\mathcal{F}(x^k) \cap \mathcal{A}^1 \neq \theta, \tag{5.14}$$

so that

$$\mathcal{A}(x^k) \subset \mathcal{A}(x^{k+1}) \text{ for } k \geq n_0 \text{ and } \mathcal{F}(x^k) \cap \mathcal{A}^1 \neq \theta. \tag{5.15}$$

Using the same arguments as above, we now deduce that either there is $k \geq n_0$ such that (5.7) is satisfied, which yields the desired result, or there is n_1 greater than n_0 such that

$$\mathcal{F}(x^k) \cap \mathcal{A}^1 = \theta \text{ for } k \geq n_1, \tag{5.16}$$

as $k \geq n_0$ and $\mathcal{F}(x^k) \cap \mathcal{A}^1 = \theta$ imply $\mathcal{F}(x^{k+1}) \cap \mathcal{A}^1 = \theta$ by (5.12).

To examine the latter case, let us assume that $k \geq n_1$ so that (5.11), (5.12), and (5.16) are satisfied. In particular, (5.11) and (5.16) imply that $\mathcal{A}(x^k) \supset \mathcal{A}^1$ and $\mathcal{F}(x^k) \supset \mathcal{F}(\bar{x})$ so that the solution $\bar{x}$ of (1.1) satisfies

$$f(\bar{x}) = \min\{f(x) \ : \ x \in \mathcal{W}(\mathcal{A}(x^k), \ x^k)\}. \tag{5.17}$$

Let us assume that x^k is disproportional, so that

$$\|\beta(x^k)\|_\infty > \Gamma \|\varphi(x^k)\|.$$

Since by assumptions $\Gamma \geq \rho(A)$, it follows that $\beta(\bar{x}) \neq o$ in contradiction with the assumption that $\bar{x}$ is the solution of (1.1). We conclude that x^k is proportional for $k \geq n_1$. Thus

$$\mathcal{A}(x^k) \subset \mathcal{A}(x^{k+1}) \subset \ \cdots \text{ for } k \geq n_1$$

and we complete the proof by the dimension argument and Lemma 5.2 (iii).

(iv) The statement is an immediate consequence of (iii) and Corollary 3.3. □

6. Proportioning with conjugate gradients.

In this section, we shall formulate a general framework for implementation of Algorithm 5.1 with the conjugate gradient method.

Inspired by [8], we shall describe our algorithms in an easily understandable variant of a Matlab-like language. To preserve readability, we do not distinguish generations of variables by indices unless it is convenient for further reference.

ALGORITHM 6.1 (PROPORTIONING WITH CONJUGATE GRADIENTS). *Given a starting vector $x \in \Omega$, $0 < \delta < g$, and $\Gamma \geq 0$, the algorithm generates a finite or infinite sequence $\{x^k\}$ in order to solve* (1.1).

```
    {Initialization.}
Step 0. $k = 0$; $x^0 = x$; $r = Ax - b$
    while $\|\nu(x^k)\| > 0$
```

```
            if $\|\beta(x^k)\|_\infty \le \Gamma\|\varphi(x^k)\|$
              {Proportional $x^k$. Initialization of the conjugate gradient loop.}
Step 1.       $y = x^k$; $J = \mathcal{F}(x^k)$; $p = P_J r$; $\alpha = 0$; $\alpha^{cg} = 0$
              while $\|\nu(y)\| > 0$ and $\alpha = \alpha^{cg}$ and $\|\beta(y)\|_\infty \le \Gamma\|\varphi(y)\|$
                {Set steplength.}
Step 2.         $\alpha^{cg} = r_J^T p_J / p_J^T A_{JJ} p_J$
                if $y - \alpha^{cg} p \in \Omega$
Step 2a.          $\alpha = \alpha^{cg}$
                else
Step 2b.          Choose $\alpha$ so that
                    $f(P_\Omega(y - \alpha p)) \le f(x^k)$ and $\mathcal{A}(x^k) \subsetneq \mathcal{A}(P_\Omega(y - \alpha p))$
                end if
                {Conjugate gradient update.}
Step 3.         $y = y - \alpha p$; $r = r - \alpha A p$
                $\beta = r_J^T A_{JJ} p_J / p_J^T A_{JJ} p_J$; $p_J = r_J - \beta p_J$
              end while
              {Set $x^{k+1}$ by Algorithm 5.1(b).}
Step 4.       $x^{k+1} = P_\Omega(y)$; $r = Ax^{k+1} - b$; $k = k + 1$
            end if
            if $\|\beta(x^k)\|_\infty > \Gamma\|\varphi(x^k)\|$
              {Disproportional $x^k$. Proportioning.}
Step 5.       Assign $d^k$ a progressive direction at $x^k$
              $\alpha_k = \min\{\delta/\|d^k\|_\infty,\ r^T d^k/(d^k)^T A d^k\}$
              $x^{k+1} = x^k - \alpha_k d^k$; $r = r - \alpha_k A d^k$; $k = k + 1$
            end if
          end while
```

Postponing the discussion on implementation of Step 2b to the next section, we shall complete this section by the following theorem.

THEOREM 6.2. *Let $\{x^k\}$ denote a finite or infinite sequence generated by Algorithm* 6.1 *with given $x^0 \in \Omega$, $0 < \delta < g$, and $\Gamma > 0$.*

(i) *If $\{x^k\}$ is finite, then Algorithm* 6.1 *ends at the solution $\bar{x}$ of* (1.1).

(ii) *If $\{x^k\}$ is infinite, then $\{x^k\}$ converges to $\bar{x}$.*

(iii) *If the problem* (1.1) *is nondegenerate, then $\{x^k\}$ is finite.*

(iv) *If $\Gamma \ge \rho(A)$, then $\{x^k\}$ is finite.*

(v) *If $\Gamma \ge \kappa(A)^{1/2}$, then $\{x^k\}$ is finite.*

Proof. The algorithm ends iff $\nu(x^k) = o$, so that x^k satisfies the Kuhn–Tucker conditions (2.1) and (2.2).

If x^k is proportional, then the inner loop beginning just after the initialization in Step 1 generates the conjugate gradient iterations until either the minimum on the face $\mathcal{W}(\mathcal{A}(x^k),\ x^k)$ is reached or some other condition is satisfied, which may happen earlier. In any case, it is easy to check that x^{k+1} is assigned in Step 4 in agreement with rule (b) of Algorithm 5.1.

If x^k is not proportional, then x^{k+1} is assigned in proportioning Step 5 in agreement with (5.1) of rule (a) of Algorithm 5.1.

To finish the proof, it is enough to apply Theorem 5.3. □

7. Steplength computation. We have implemented three variants of the choice of α in Step 2b of Algorithm 6.1 that may be conveniently described by means of the following two procedures. For convenience, we shall assume that min over the empty set returns ∞.

Algorithm 7.1 (Feasible steplength). *Given n-vectors y and p, the algorithm returns $\alpha = \max\{\mu \ : \ p - \mu p \in \Omega\}$ for $y \in \Omega$ and $\alpha = 0$ for $y \notin \Omega$.*

```
function: $\alpha = fs(y, p)$
  $\mu^0 = \min\{(y_i - l_i) * (u_i - y_i) \ : \ i \in \mathcal{N}\}$
  $\mu^l = \min\{(y_i - l_i)/p_i \ : \ i \in \mathcal{N}$ and $p_i > 0\}$
  $\mu^u = \min\{(y_i - u_i)/p_i \ : \ i \in \mathcal{N}$ and $p_i < 0\}$
  $\alpha = (\min\{\mu^l, \mu^u, \mu^0\})^+$
end
```

Algorithm 7.2 (Conjugate gradient or feasible steplength). *Given n-vectors x and y and $p \neq o$, the algorithm returns the conjugate gradient steplength if*

$$f(P_\Omega(y - \alpha^{cg} p)) \leq f(P_\Omega(x))$$

and $\alpha = fs(y, p)$ otherwise.

```
function: $\alpha = cgfs(x, y, p)$
  $\alpha^{cg} = r(y)^T p / p^T A p$
  if $f(P_\Omega(y - \alpha^{cg} p)) \leq f(P_\Omega(x))$
    $\alpha = \alpha^{cg}$
  else
    $\alpha = fs(x, p)$
  end
end
```

Probably the most simple choice of α in Step 2b of Algorithm 6.1 is

$$\alpha = fs(y, p). \tag{7.1}$$

We shall call it a *feasible strategy* as it ensures that $y - \alpha p \in \Omega$. The feasible strategy has been used in the Polyak algorithm and its variants [6, 18].

If we wish to generate the conjugate gradient iterations as long as the function value at the projection decreases or at least does not increase, we replace Step 2b by

$$\alpha = cgfs(y, y, p). \tag{7.2}$$

We shall call the choice (7.2) a *monotonic strategy* as it yields monotonic reduction of energy even in an inner conjugate gradient loop.

With regard to the well-known selfpreconditioning properties of the conjugate gradient method, we may wish to carry out the conjugate gradient iterations as long as possible. The choice

$$\alpha = cgfs(x^k, y, p) \tag{7.3}$$

does just this; we shall call it the *as long as possible strategy.*

The performance of the algorithm may be significantly improved by preconditioning. One suitable method is the SSOR preconditioning [9] since it requires neither additional storage space nor updating of the preconditioner. The drawback is that the cost of one conjugate gradient step amounts to two matrix vector multiplications.

Finally, to solve any problem with the proportioning algorithm, we have to specify Γ, which should be large enough to prevent frequent change of active sets without good reason. At the same time Γ should not be too large so that the solution of auxiliary problems is not too expensive. In all our experiments we obtained good results with

$\Gamma = 1$. Indeed, it seems reasonable to release the bounds whenever the violation of the Kuhn–Tucker conditions in active variables dominates that in free variables, since only the latter are reduced by the conjugate gradient method. We often observed better results with $\Gamma = 0.1$ for $x^0 = l$, especially for preconditioned algorithms.

8. Numerical experiments. We have implemented our algorithms with the progressive direction $d = \beta$ and tested them on two model problems. We used the stopping criterion $\|\nu(x)\| \le 10^{-5}\|b\|$.

The first problem arises from the discretization of the inner obstacle problem to find the minimum of

$$f(u) = \frac{1}{2}\int_0^{0.5} \|u'(x)\|^2 dx + \int_0^{0.5} bu\, dx$$

subject to $u \in K$, where

$$K = \{u \in H^1[0, 0.5]\ :\ l \le u \text{ on } (0, 0.5) \text{ and } u(0) = u'(0.5) = 0\}.$$

The problem was discretized by linear finite elements on a regular grid with $n + 1$ nodes $x_i = \frac{1}{2}i/n,\ i = 0, 1, \ldots, n$.

We have used two supports in our tests. The concave support was defined by the upper part of circle of radius $R = 2.03$ with center $S = (.5,\ -2.032)$. The convex support was defined by the lower part of circle of radius $R = 2.03$ with center $S = (.5,\ 1.943)$. The latter problem is such that if we start the solution with $x^0 = o$, the first constraints that are activated in the process of solution are released in a later stage of computation.

Three combinations of supports and x^0 have been used. The concave support with $x^0 = o$ and $x^0 = l$ was used to demonstrate the performance of our algorithms on problems with expanding and shrinking active sets, respectively, and the convex support with $x^0 = o$ was used to examine the performance of our algorithms on problems with chaotic change of the active sets.

Depending on implementation of the steplength computations, we have used three variants of Algorithm 6.2. The algorithms with feasible strategy (7.1), monotonic strategy (7.2), and as long as possible strategy (7.3) are identified by $QPPf$, $QPPm$, and $QPPalap$, respectively.

We have carried out our computations with $n = 50$. The results for various values of Γ are in Table 1. The computational cost has been measured by the number n^A of multiplications by the matrix A. We have also included a number n^i of iterations x^i that may serve as an upper bound for the number of faces examined. Using the Matlab function cond, we found that $\kappa(A)^{1/2} \doteq 64$.

The second problem is the elastic–plastic torsion problem that was used to assess the performance of algorithms in [15] and [16]. The problem on the domain $\mathcal{D} = (0, 1) \times (0, 1)$ is defined by an obstacle function

$$l(x) = -\text{ dist } (x, \partial\mathcal{D})$$

and a constant function b on $\mathcal{D}$. The problem is to find the minimum of

$$f(u) = \frac{1}{2}\int_{\mathcal{D}} \|\nabla u\|^2 d\mathcal{D} - \int_{\mathcal{D}} bu\, d\mathcal{D}$$

subject to $u \in K$, where

$$K = \{v \in H_0^1(\mathcal{D})\ :\ l \le u \text{ on } \mathcal{D}\}.$$

Table 1
Performance of variants of proportioning algorithm on 1D *problem.*

Support	x^0	Algorithm	$\Gamma = 64$		$\Gamma = 5$		$\Gamma = 1$		$\Gamma = 0.1$	
			n^i	n^A	n^i	n^A	n^i	n^A	n^i	n^A
convex	o	$QPPf$	66	567	66	300	16	169	50	192
		$QPPm$	42	547	42	277	16	150	33	132
		$QPPalap$	42	594	42	324	42	197	33	179
concave	o	$QPPf$	20	87	20	87	23	116	36	130
		$QPPm$	7	137	7	109	7	90	13	114
		$QPPalap$	18	231	18	134	18	89	19	156
concave	l	$QPPf$	55	417	55	242	52	120	47	103
		$QPPm$	55	417	55	242	52	120	47	103
		$QPPalap$	55	417	55	242	52	120	47	103

Table 2
Elastic–plastic torsion problem.

		$QPPm$				$QPPm - SSOR$			
b	x^0	$\Gamma = 0.1$		$\Gamma = 1$		$\Gamma = 0.1$		$\Gamma = 1$	
		n^i	n^A	n^i	n^A	n^i	n^A	n^i	n^A
-5	l	35	385	32	426	31	304	32	401
	o	23	554	19	549	23	368	19	427
-10	l	17	159	15	179	17	128	15	147
	o	13	195	13	235	13	168	13	180
-20	l	7	61	8	73	7	51	8	65
	o	8	87	8	99	6	60	8	86

The problem has been discretized by the triangular elements on a regular grid in the same way as in [16].

We have used the torsion problem to test the preconditioned and unpreconditioned variants of $QPPm$. To this end, we have implemented $QPPm$ with SSOR preconditioning [9]. Table 2 shows the performance of $QPPm$ and its preconditioned variant $QPPm - SSOR$ for mesh 102 × 102.

9. Comparison with other codes. We shall tentatively compare the performance of the proposed algorithm with two related codes ($GPCG$ of Moré and Toraldo [16] and $SSORP - PCG$ of Kočvara and Zowe [15]). In both cases, we shall use the elastic–plastic torsion problem of the previous section and the results for $GPCG$ and $SSORP - PCG$ deduced from Table 9 of [15] with $n_i = n_I$ and $n^A = n_I \times n_m$. To enable the comparison, we resolved the torsion problem with the stopping rule

$$\|\nu(x^k)\| \leq 10^{-5}\|r(x_0)\|$$

that was used in [15] and [16]. We use heuristics of section 7 for the choice of Γ.

The code $GPCG$ does not use any problem-dependent preconditioning, so that it seems fair to compare its performance with $QPPm$. As in [16], we consider three mesh sizes with 77×77, 102×102, and 127×127 nodes. Table 3 shows the performance of $QPPm$ for $\Gamma = 1$ and $GPCG$ for initial approximation $x^0 = l$.

The algorithm $SSORP - PCG$ uses an efficient strategy for the initial approximation I^* of $\mathcal{A}(\bar{x})$. Since we are more interested in comparing basic strategies, we

TABLE 3
Comparison with Moré and Toraldo.

		$QPPm$		$GPCG$	
b	Mesh	n^i	n^A	n^i	n^A
-5	77x77	24	268	9	268
	102x102	32	399	11	415
	127x127	39	471	12	522
-10	77x77	12	163	7	172
	102x102	15	173	7	204
	127x127	20	221	7	243
-20	77x77	5	44	5	90
	102x102	8	72	5	101
	127x127	9	86	5	121

TABLE 4
Comparison with Kočvara and Zowe.

		$QPPm-SSOR$		$SSORP-PCG$	
b	Mesh	n^i	n^A	n^i	n^A
-20	77x77	6	38	4	44
	102x102	7	49	6	60
	127x127	10	71	4	44

restrict our attention to the case when $\mathcal{A}(\bar{x})$ is close to our initial approximation $\mathcal{A}(l)$. Thus $b = -20$ seems suitable for our purpose. Since $SSORP - PCG$ uses preconditioning, we compare it with $QPPm - SSOR$. We use $\Gamma = 0.1$ according to heuristics of section 7. Table 4 shows the performance of $QPPm - SSOR$ and $SSORP - PCG$ for $x^0 = l$.

According to an explanation passed kindly by Kočvara, the result of $SSORP - PCG$ for 127×127 grid represents the rare case that after changing the active sets by thousands of elements in each outer iteration, the algorithm hits $\mathcal{A}(\bar{x})$ at an early stage of the computation.

10. Comments and conclusions. We have presented a class of algorithms whose performance depends on a release coefficient Γ that controls the precision of the solution of auxiliary problems. In particular, for $\Gamma = 0$ we get variants of feasible direction methods that may not converge to the solution [1], for positive Γ that is less than the critical release coefficient $\rho(A)$ we get convergent algorithms that reach the solution of nondegenerate problems in a finite number of steps, and for $\Gamma \geq \rho(A)$ we get algorithms with the finite termination property even for degenerate problems. The projections may be exploited so that the algorithms can drop and add many constraints in one step.

The idea of proportioning is quite general and may be incorporated into other algorithms including those of Moré and Toraldo or Kočvara and Zowe. Moreover, since proportioning guarantees the convergence, other variants of known algorithms may be considered in simplified form. For example, the algorithm of Moré and Toraldo with proportioning may not require sufficient decrease condition.

The first numerical results for algorithms based on proportioning seem to be quite interesting and deserve further investigation. We believe that the proportioning may be exploited to the development of efficient and reliable algorithms for the solution of realistic problems.

REFERENCES

[1] M. S. Bazaraa and C. M. Shetty, *Nonlinear Programming*, John Wiley, New York, 1979.

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

[3] P. H. Calamai and J. J. Moré, *Projected gradient methods for linearly constrained problems*, Math. Programming, 39 (1987), pp. 93–116.

[4] G. Cimatti, *On a problem of the theory of lubrication governed by a variational inequality*, Appl. Math. Optim., 3 (1977), pp. 227–242.

[5] Z. Dostál, *Duality based domain decomposition for the solution of free boundary problems*, J. Comput. Appl. Math., 63 (1995), pp. 203–208.

[6] Z. Dostál, *Directions of large decrease and quadratic programming*, in Software and Algorithms of Numerical Mathematics X, I. Marek, ed., Charles University, Prague, 1993, pp. 9–22.

[7] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, Berlin, New York, 1984.

[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*, John Hopkins, London, 1989.

[9] I. Gustafsson, *Modified incomplete cholesky methods*, in Preconditioning Methods, D. J. Evans, ed., Gordon and Breach, New York, 1983.

[10] W. Hackbusch and H. D. Mittelmann, *On multigrid methods for variational inequalities*, Numer. Math., 42 (1983), pp. 65–76.

[11] J. Haslinger and P. Neittaanmäki, *Finite Element Approximation for Optimal Shape Design*, John Wiley, Chichester, 1988.

[12] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Stand., 49 (1952), pp. 409–436.

[13] J. J. Júdice and F. M. Pires, *Direct methods for convex quadratic programs subject to box constraints*, Investigación Operacional, 9 (1989), pp. 23–56.

[14] A. Klarbring, *Quadratic programs in frictionless contact problems*, Internat. J. Engrg. Sci., 24 (1986), pp. 1207–1217.

[15] M. Kočvara and J. Zowe, *An iterative two-step algorithm for linear complementarity problems*, Numer. Math., 68 (1994), pp. 95–106.

[16] J. J. Moré and G. Toraldo, *On the solution of large quadratic programming problems with bound constraints*, SIAM J. Optim., 1 (1991), pp. 93–113.

[17] D. P. O'Leary, *A generalized conjugate gradient algorithm for solving a class of quadratic programming problems*, Linear Algebra Appl., 34 (1980), pp. 371–399.

[18] B. T. Polyak, *The conjugate gradient method in extremal problems*, U.S.S.R Comput. Math. and Math. Phys., 9 (1969), pp. 94–112.

[19] E. K. Yang and J. W. Tolle, *A class of methods for solving large convex quadratic programs subject to box constraints*, Math. Programming, 51 (1991), pp. 223–228.

[20] I. Zhang, R. Tapia, and F. Potra, *On the superlinear convergence of interior-point algorithms for a general class of problems*, SIAM J. Optim., 3 (1993), pp. 413–422.

SIAM J. OPTIM.
Vol. 7, No. 4, pp. 889–912, November 1997

A STUDY OF GENERAL DYNAMIC NETWORK PROGRAMS WITH ARC TIME-DELAYS[*]

MALCOLM C. PULLAN[†]

Abstract. Dynamic network flow problems in which there is a delay on the flows in the arcs have been in existence since the early days of modern optimization. However, most previous work in this area has only considered models which are discrete in the time variable. In this paper we present a continuous-time model for a very broad class of dynamic network problems with arc time-delays. The model is a direct extension of the separated continuous linear program (SCLP) to include time-delays and is called the separated continuous linear program with time-delays (SCLPTD). By suitable transformations we are able to rewrite SCLPTD in a manner which is very close to SCLP itself. This then allows us to use all the recent theory and algorithms for SCLP to derive similar results for SCLPTD. In particular, the theory we present includes a characterization of the extreme-point solutions, an existence theorem for piecewise analytic optimal extreme-point solutions, and a strong duality theorem. We also present a class of convergent algorithms for the solution of SCLPTD in certain instances.

Key words. dynamic network flows, duality, continuous linear programming, linear optimal control

AMS subject classifications. 90C35, 49J30, 49N15, 49K30, 49M99, 49N05, 90C45

PII. S1052623495288180

1. Introduction.

Problems in networks form a large area of optimization. It is generally accepted that the foundation of this subject is the book by Ford and Fulkerson [13]. One of the problems that the authors discussed in [13] is a maximum dynamic network flow problem in which the commodity being transported takes some fixed amount of time to traverse the arc. The authors proposed that this problem is solved by solving a time-expanded network problem. This time-expanded network comprises copies of the nodes of the original network, each representing the original node at a particular time. The nodes are then linked by arcs over time, with the amount of time the arc spans being the delay or traversal time of that arc.

Since then a large number of authors have considered dynamic network flow problems, both with and without arc delays. These problems are not just restricted to maximum flow problems, but also include general minimum cost flow problems. The interest in such problems is no doubt because of the wide number of possible applications, such as building evacuation (see Chalmet, Francis, and Saunders [10]) or the dynamic routing of messages in communications networks (Frank [14], Segall [31], and Moss and Segall [18]). We refer the reader to Lovetskii and Melamed [17] for a survey of previous work in this area, both general models and their various applications. In [17], the authors make it clear that along with arc delays, another desirable feature in dynamic network flow problems is the possibility of storage at the nodes. Again, the traditional way to solve such problems is to form a time-expanded network with additional links between the same nodes at different times to represent the storage. This discrete approach appears to be firmly established as the way of solving such problems. For instance, in the recent encyclopedic book on network flow problems,

[*] Received by the editors June 26, 1995; accepted for publication (in revised form) February 7, 1996.

http://www.siam.org/journals/siopt/7-4/28818.html

[†] Department of Mathematical Sciences, Loughborough University, Loughborough, Leicestershire LE11 3TU, UK (m.c.pullan@elboro.ac.uk). This work was done while at St. John's College, Cambridge CB2 1TP, UK.

Ahuja, Magnanti, and Orlin [1], dynamic network problems are solved by using this approach without any further discussion. However, this discrete approach has a serious drawback in that the times at which decisions are made are predetermined before the problem is solved. This is by no means a necessary feature of the problem, and it would be desirable in many instances to allow decisions to be made at any arbitrary time in the time interval. For this to be a possibility we must consider a continuous model in which the flows in the arcs are functions representing the rates of flow at any particular time.

Continuous-time models for network flow problems were first considered by Philpott [20] and further studied in Anderson, Nash, and Philpott [5], Philpott [21], Anderson and Philpott [6], and Ogier [19]. The problems considered by these authors are all continuous-time analogues of various single-commodity network problems, which include the possibility of storage at the nodes, but not arc delays. The work by Ogier gives an algorithm for a very specific type of continuous-time network problem. The culmination of the work of the other authors was an algorithm for solving a general single-commodity network problem, called the *continuous network program* (CNP), under certain restrictions on the problem data. Unfortunately, it was later revealed that this algorithm often failed to converge to an optimal solution.

The first attempt at solving any sort of continuous network problems which includes arc delays appears to be by Philpott [22]. Here the author considered a maximum dynamic flow problem and proved a max-flow, min-cut theorem. The first discussion of a general continuous-time minimum cost dynamic network flow problem with both arc delays and storage appears to be by Anderson [3]. Here the author characterized the extreme-point solutions for the problem given rational traversal times. The problem studied in [3] is called the *dynamic network flow problem* (DNFP) and can be written as follows:

$$\begin{array}{lll}
\text{DNFP:} & \text{minimize} & \displaystyle\int_0^T \sum_{(i,j)\in A} c_{ij}(t)x_{ij}(t)\,dt \\
& \text{subject to} & \displaystyle y_j(t) = y_j(0) + \int_0^t \left[r_j(s) + \sum_{i=1}^n (x_{ij}(s-\lambda_{ij}) - x_{ji}(s)) \right] ds, \\
& & 0 \le y_j(t) \le a_j(t), \quad j = 1,\dots,n, \\
& & 0 \le x_{ij}(t) \le b_{ij}(t), \quad (i,j) \in A, \qquad t \in [0,T].
\end{array}$$

The problem is only defined over the interval $[0, T]$, and so it is an implicit constraint that $x_{ij}(t) = 0$ for each i and j and $t < 0$. Here A represents the set of arcs in a network of n nodes, and the variables are $x_{ij}(t)$, a bounded measurable function representing the rate of flow in arc (i, j) at time t, and $y_j(t)$, an absolutely continuous function representing the storage in node j at time t. The costs $c_{ij}(t)$ and the upper bounds on the flows $b_{ij}(t)$ are bounded measurable functions, and the storage bound $a_j(t)$ is an absolutely continuous function. The quantity λ_{ij} is the traversal time for the arc (i, j) and is assumed to be nonnegative.

Further work on DNFP does not appear to have been done until quite recently, in Anderson and Philpott [8]. Here the authors survey results relating to similar problems as well as introducing a dual with a corresponding definition of complementary slackness and prove a weak duality result.

It would appear from the above that, although modelling dynamic network problems in continuous time is desirable, it is not practical, as the resulting problems cannot be solved. Indeed, Lovetskii and Melamed [17] make precisely this observation following the introduction of continuous-time models. However, recent work on a more general class of problems has changed this, at least for the case where the network does not include arc delays. This more general class of problems is called the *separated continuous linear program* (SCLP) and is defined as follows:

$$\begin{aligned} \text{SCLP:} \quad & \text{minimize} \quad && \int_0^T c(t)^T x(t)\,dt \\ & \text{subject to} \quad && \int_0^t Gx(s)\,ds + y(t) = a(t), \\ &&& Hx(t) + z(t) = b(t), \\ &&& x(t), y(t), z(t) \geq 0, \qquad t \in [0, T]. \end{aligned}$$

Here $x(t)$, $z(t)$, $b(t)$, and $c(t)$ are bounded measurable functions and $y(t)$ and $a(t)$ are absolutely continuous functions. By taking G to be a node-arc incidence matrix, H the identity matrix, and $a_j(t)$ the total supply in node j up to time t, we obtain the single-commodity continuous network program CNP (see Anderson and Philpott [6]). The variable $x_i(t)$ then represents the rate of flow in arc i, and the variable $y_j(t)$ represents the storage in node j at time t. However, SCLP is much more general than a continuous single-commodity network program. In fact, SCLP can easily be used to give continuous analogues of various network problems without arc time-delays, such as multicommodity network programs, generalized network programs, or any of these with side constraints.

The problem SCLP first appeared in Anderson [2] as a continuous model for job-shop scheduling problems. The study of SCLP was continued in Anderson, Nash, and Perold [4], where a characterization of extreme-point solutions was given as well as a result for the existence of optimal solutions with a finite number of breakpoints in certain cases. However, it is the more recent work on the problem that has made solving continuous network programs a possibility. This work may be found in Pullan [24, 25, 27, 28, 29, 30], Anderson and Pullan [9], and Anderson and Philpott [7]. Among other things, these papers give a class of convergent algorithms for solving the problem in certain instances and extensive theories of duality and for the existence of optimal solutions with a finite number of breakpoints. This work has also resulted in a further convergent algorithm for the single-commodity network program CNP (see Philpott and Craddock [23]).

These advances in SCLP, and hence in CNP, show that solving continuous models is viable and raises the question of whether this is true for CNPs with arc time-delays, such as DNFP. Indeed Anderson and Philpott [8] show that it is quite a simple matter to extend some of the less technical matters from CNP to DNFP, such as the weak duality result referred to above. The purpose of this paper is to go much further than [8] and to show that, at least from a theoretical point of view, arc time-delays present no new problems whatsoever when the delay times are rational. This is because it is possible to transform the problem into one which is very close to a special case of SCLP but with extra constraints on y. Although the results from the papers mentioned above on SCLP cannot be applied directly, the same ideas can be used without any difficulty to arrive at the desired results.

We now define the problem that we shall study in this paper. In order to achieve full generality so that any possible network problem is covered by the results, we shall study a direct extension of SCLP to include time-delays on $x(t)$. We give this problem the name *separated continuous linear program with time-delay* (SCLPTD). As with SCLP, the increased generality unfortunately means that the network structure is lost in the discussion. We define the separated continuous linear programs with time delay as follows:

$$\begin{aligned}
&\text{SCLPTD:} \quad \text{minimize} && \int_0^T c(t)^T x(t)\,dt \\
&(1) \qquad \text{subject to} && \int_0^t (Gx(s))_i\,ds + \sum_{j=1}^{n_1} \int_0^t f_{ij} x_j(s-\lambda_{ij})\,ds + y_i(t) = a_i(t), \\
& && \hspace{20em} i = 1, \dots, n_2, \\
& && Hx(t) + z(t) = b(t), \\
& && x(t), y(t), z(t) \ge 0, \qquad t \in [0,T].
\end{aligned}$$

Again the problem is only defined over the interval $[0,T]$, and so it is an implicit constraint that $x(t) = 0$ for $t < 0$. Here, as with SCLP, $x(t)$, $z(t)$, $b(t)$, and $c(t)$ are bounded measurable functions and $y(t)$ and $a(t)$ are absolutely continuous functions. The dimensions of $x(t)$, $y(t)$, and $z(t)$ are n_1, n_2, and n_3, respectively. We let $\omega(t)$ denote a complete set of variables for SCLPTD, i.e., $\omega(t)^T = (x(t)^T, y(t)^T, z(t)^T)$. As with DNFP, we refer to λ_{ij} as the *traversal times*, and these are assumed to be nonnegative. We can also assume without loss of generality that $\lambda_{ij} < T$ for each i and j, because if $\lambda_{ij} \ge T$ for some i and j, then the integral in (1) of $f_{ij}x(s-\lambda_{ij})$ is always zero, and so we could define $f_{ij} = 0$ instead. Clearly, SCLPTD includes DNFP as a special case. SCLPTD is also a special case of a linear optimal control problem with state positivity constraints. However, apart from the articles on network problems such as DNFP above, the only comparable problem in the literature appears to be that by Farr and Hanson [12]. Here the authors give some duality results for a nonlinear continuous-time programming problem with time-delays.

For the purposes of this paper we shall study SCLPTD under the following weak assumption.

ASSUMPTION 1.1. *The traversal times are all rational, as is the final time T.*

We will assume that this holds throughout the rest of this paper. From a practical point of view this is no restriction at all, because any measurement of a traversal time in a practical problem must give rational data or, at least, only be able to be represented as a rational number on a computer. In any case, in all the literature on discrete dynamic network problems with arc delays, the traversal times and time T are integers.

The plan of this paper is as follows. In section 2 we transform SCLPTD into a problem which is very close to a special case of SCLP, the difference being that there are extra constraints connecting $y(0)$ and $y(T)$. This transformed problem allows us to repeat many of the more important results on SCLP for the problem SCLPTD. This repetition of results takes up the rest of the paper. Most of the proofs are either identical or are very minor extensions to the previous proofs for SCLP. For this reason, and to allow as much theory to be covered as possible in a single paper, we either omit the proofs or just present an outline.

The results that we present are as follows. In section 3 we study the feasible region of SCLPTD and show that it is convex and closed in the $\sigma(L_\infty^{n_1}[0,T], L_1^{n_1}[0,T])$ topology (the weak topology on the dual pair of vector spaces $(L_\infty^{n_1}[0,T], L_1^{n_1}[0,T])$; see, for example, Holmes [15]). We then give a characterization of the extreme points of the feasible region which closely resembles the result for SCLP in Anderson, Nash, and Perold [4]. This result includes the result for DNFP given in Anderson [3]. Under the further assumption that the feasible region is nonempty and bounded, we then show that it is both compact and sequentially compact in the $\sigma(L_\infty^{n_1}[0,T], L_1^{n_1}[0,T])$ topology. Hence we conclude that, in this case, SCLPTD has an optimal extreme-point solution. In section 4 we then extend the results from Pullan [25] to show that a piecewise analytic optimal extreme-point solution exists for SCLPD given piecewise analytic problem data. In section 5 we introduce a dual problem for SCLPTD based on the dual problem SCLP* for SCLP. Again this includes the dual problem for DNFP given in Anderson and Philpott [8]. We then prove a weak duality result. In section 6 we consider the extension of the algorithms discussed in Pullan [24, 29] to SCLPTD. This involves introducing special discretizations of the problem. Once this is done it is fairly trivial to see that all the results from these two papers carry over verbatim, resulting in a class of convergent algorithms for SCLPTD and a strong duality theorem. Such a theorem was the starting point of the extensive duality theory for SCLP in Pullan [27]. We remark that this theorem for SCLPTD will probably lead to the same duality theory for SCLPTD, but as [27] is very long and technical we do not pursue this matter at this point.

Finally, in section 7, we comment on the results achieved for SCLPTD. We note that the results have all been proved quite readily from their SCLP counterparts. We then comment on those results from SCLP that we have not extended to SCLPTD in this paper, most notably the duality theory in Pullan [27], and conclude that it is probably not difficult to establish these results as well. We also comment on the algorithms for SCLPTD developed in this paper. In particular it is noted that, because the transformed problem could be very large in general, the algorithms may be difficult to use, although perhaps not impossible.

2. Transformation of SCLPTD.

In this section we transform the problem SCLPTD into one that closely resembles SCLP. This then allows us to study the problem in a similar light to SCLP. The transformation here of SCLPTD has several steps.

First, given Assumption 1.1 concerning the rational traversal times and time T, we can rewrite the problem so that T is an integer and λ_{ij} is an integer between 0 and $T-1$ inclusive for each i and j. Indeed, because λ_{ij}, for each i and j, and T form a finite set of rational numbers, we can write all these quantities as fractions with the same common denominator; i.e., for some integers μ_{ij}, N, and S, we have $\lambda_{ij} = \mu_{ij}/N$, for each i and j, and $T = S/N$. We can then make the substitution $\tau = Nt$ in SCLPTD to give an equivalent formulation of the problem with integer traversal times and final time.

Assume then that SCLPTD has T an integer and λ_{ij} an integer between 0 and $T-1$ inclusive for each i and j. We can now change the problem further so that no λ_{ij} is zero by absorbing the corresponding f_{ij} into the matrix G in SCLPTD. In particular, we can replace G by G', where

$$G'_{ij} = \begin{cases} G_{ij} + f_{ij}, & \lambda_{ij} = 0, \\ G_{ij}, & \text{otherwise.} \end{cases}$$

Given then that the traversal times are all nonzero, we now define matrices $F^{(k)}$, $k = 1, \ldots, T-1$, each of dimension $n_2 \times n_1$, by

$$F^{(k)}_{ij} = \begin{cases} f_{ij}, & \lambda_{ij} \le k, \\ 0, & \text{otherwise.} \end{cases}$$

Thus, for each k and i,

$$(F^{(k)}x(t))_i = \sum_{\{\,j:\lambda_{ij}\le k\,\}} f_{ij}x_j(t).$$

We can now write the problem SCLPTD in the following equivalent manner:

$$\begin{aligned}
\text{T1:}\quad \text{minimize}\quad & \int_0^T c(t)^T x(t)\,dt\\
\text{subject to}\quad & \int_0^t Gx(s)\,ds + \sum_{k=1}^{\lfloor t\rfloor}\int_0^t F^{(k)}x(s-k)\,ds + y(t) = a(t),\\
& Hx(t) + z(t) = b(t),\\
& x(t), y(t), z(t) \ge 0, \qquad t\in[0,T],
\end{aligned}$$

where $\lfloor t\rfloor$ denotes the greatest integer less than or equal to t.

We now perform the final, and more radical, transformation on the problem. Define the matrices $\mathcal{G}$ and $\mathcal{H}$ of dimensions $Tn_2 \times Tn_1$ and $Tn_3 \times Tn_1$, respectively, by

$$\mathcal{G} = \begin{bmatrix}
G & 0 & \cdots & & & & 0\\
F^{(1)} & G & 0 & \cdots & & & 0\\
F^{(2)} & F^{(1)} & G & 0 & \cdots & & 0\\
F^{(3)} & F^{(2)} & F^{(1)} & G & 0 & \cdots & 0\\
\vdots & \vdots & \vdots & \vdots & \ddots & & \vdots\\
F^{(T-2)} & F^{(T-3)} & F^{(T-4)} & F^{(T-5)} & \cdots & G & 0\\
F^{(T-1)} & F^{(T-2)} & F^{(T-3)} & F^{(T-4)} & \cdots & F^{(1)} & G
\end{bmatrix},$$

$$\mathcal{H} = \begin{bmatrix}
H & 0 & \cdots & & & & 0\\
0 & H & 0 & \cdots & & & 0\\
0 & 0 & H & 0 & \cdots & & 0\\
0 & 0 & 0 & H & 0 & \cdots & 0\\
\vdots & \vdots & \vdots & \vdots & \ddots & & \vdots\\
0 & 0 & 0 & 0 & \cdots & H & 0\\
0 & 0 & 0 & 0 & \cdots & 0 & H
\end{bmatrix}.$$

We now define functions $A(t)$, $B(t)$, and $C(t)$ over the interval $[0,1]$ as follows:

$$A(t) \;=\; \begin{bmatrix} a(t)-a(0)\\ a(t+1)-a(1)\\ \vdots\\ a(t+T-1)-a(T-1)\end{bmatrix},\quad B(t) \;=\; \begin{bmatrix} b(t)\\ b(t+1)\\ \vdots\\ b(t+T-1)\end{bmatrix},$$

$$C(t) \;=\; \begin{bmatrix} c(t)\\ c(t+1)\\ \vdots\\ c(t+T-1)\end{bmatrix}.$$

This defines the problem data for the transformed problem that we are working towards which have obvious counterparts in the normal SCLP model. We now define some more problem data which do not have direct counterparts in the SCLP model. In particular, we define a vector d of dimension Tn_2 by

$$d = \begin{bmatrix} a(0) \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

We also define a matrix $\mathcal{D}$, of dimension $Tn_2 \times Tn_2$, by

$$\mathcal{D} = \begin{bmatrix} 0 & 0 & \cdots & & & & 0 \\ I & 0 & 0 & \cdots & & & 0 \\ 0 & I & 0 & 0 & \cdots & & 0 \\ 0 & 0 & I & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & I & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & I & 0 \end{bmatrix}.$$

This completes the specification of the problem data for the transformed problem. The variables for the transformed problem are $X(t)$, a bounded measurable function of dimension Tn_1, $Y(t)$, an absolutely continuous function of dimension Tn_2, and $Z(t)$, a bounded measurable function of dimension Tn_3. We now define the transformed problem T2 over the time interval $[0,1]$ as follows:

$$\text{T2:} \quad \text{minimize} \quad \int_0^1 C(t)^T X(t)\,dt$$

$$\text{(2)} \qquad \text{subject to} \quad \int_0^t \mathcal{G}X(s)\,ds + Y(t) - Y(0) = A(t),$$

$$\text{(3)} \qquad Y(0) - \mathcal{D}Y(1) = d,$$

$$\text{(4)} \qquad \mathcal{H}X(t) + Z(t) = B(t),$$

$$X(t), Y(t), Z(t) \geq 0, \qquad t \in [0,1].$$

This is very similar to SCLP. In fact, if $\mathcal{D}$ were zero, this would be precisely an SCLP problem. As with SCLP, we let $\Omega(t)^T = (X(t)^T, Y(t)^T, Z(t)^T)$ denote a complete set of variables for T2.

It is not too difficult to see that T2 and T1 are equivalent problems, and hence so are T2 and SCLPTD. This can be seen by making the following connection between the variables:

$$\text{(5)} \qquad \Omega(t) = \begin{bmatrix} \omega(t) \\ \omega(t+1) \\ \vdots \\ \omega(t+T-1) \end{bmatrix}, \qquad t \in [0,1].$$

With this connection it is clear that a set of variables is feasible (optimal) for T1 if and only if the corresponding set of variables is feasible (optimal) for T2.

This completes our transformations of the problem SCLPTD. In the next sections we shall study SCLPTD in more depth by studying T2 in the light of recent work on SCLP. In particular, to prove something about SCLPTD, such as the existence of piecewise analytic optimal solutions, we shall prove the result for T2 and thus conclude that the result is true for SCLPTD. This will involve considerable switching between SCLPTD solutions and T2 solutions by constructing the appropriate T1 solution and then using (5), or vice versa. In order to avoid referring to this process every time, we shall adopt the notation that if $\omega(t)$ is defined as a feasible solution for SCLPTD, then $\Omega(t)$ shall mean the T2 solution given by (5) from the corresponding T1 solution. Similarly, given a feasible solution $\Omega(t)$ for T2, $\omega(t)$ shall mean the SCLPTD solution constructed from the T1 solution which is given by (5).

3. Structure of the feasible region. We now begin the study of SCLPTD with a study of the topological nature of the feasible region, that is, the set

$$\begin{aligned} F = \{\, x(t) \in L_\infty^{n_1}[0,T] : \; & \text{there exists an absolutely continuous function } y(t) \text{ and a} \\ & \text{bounded measurable function } z(t) \text{ such that } \omega(t)^T = (x(t)^T, y(t)^T, z(t)^T) \\ & \text{is feasible for SCLPTD}\,\}. \end{aligned}$$

We now prove the following result concerning F. The equivalent result for SCLP is split into various parts in the literature and may be found in Anderson, Nash, and Perold [4] and Pullan [27, 26].

THEOREM 3.1. *The feasible region F for* SCLPTD *is both convex and closed in the $\sigma(L_\infty^{n_1}[0,T], L_1^{n_1}[0,T])$ topology. Furthermore, if F is nonempty and bounded, then it is also both compact and sequentially compact in the $\sigma(L_\infty^{n_1}[0,T], L_1^{n_1}[0,T])$ topology. Hence, in this case, there exists an optimal extreme-point solution for* SCLPTD.

Proof. The proof is essentially the same as the proofs of the similar results for SCLP in the above-mentioned papers. We will thus keep the exposition brief. The convexity of F is trivial. We now prove that F is closed in the $\sigma(L_\infty^{n_1}[0,T], L_1^{n_1}[0,T])$ topology. This result was proved for SCLP in [27, Lem. 4.1].

Suppose $x \notin F$. There are three cases to consider depending on which constraint of SCLPTD is violated. If $x_i(t) < 0$ on some set S of nonzero measure for some i, then define $h \in L_1^{n_1}[0,T]$ by

$$h_j = \begin{cases} 0, & j \neq i, \\ \chi_S, & j = i, \end{cases}$$

where χ_S is the characteristic function of S. Then

$$\int_0^T h(t)^T x(t)\, dt < 0,$$

but for any $\alpha \in F$,

$$\int_0^T h(t)^T \alpha(t)\, dt \geq 0,$$

and so x is contained in some weakly open set that does not intersect with F. Now suppose that

$$\int_0^t (Gx(s))_i\, ds + \sum_{j=1}^{n_1} \int_0^t f_{ij} x_j(s - \lambda_{ij})\, ds > a_i(t)$$

for some index i and $t \in [0,T]$. By the continuity of the integrals and of a, this will be true for all t in some open interval $S = (t_1, t_2)$, with equality at the point t_1.

Define $S(s) = [0,T] \cap (t_1 - s, t_2 - s)$ for $s \in [0,T]$. Thus $S = S(0)$. We now define $h \in L_1^{n_2}[0,T]$ by

$$h_j = G_{ij}\chi_S + f_{ij}\chi_{S(\lambda_{ij})}.$$

Then

$$\int_0^T h(t)^T x(t)\,dt > \int_S a_i(t)\,dt,$$

and for any $\alpha \in F$,

$$\int_0^T h(t)^T \alpha(t)\,dt \le \int_S a_i(t)\,dt,$$

and so again, x is contained in some weakly open set that does not intersect with F. The remaining case, namely $(Hx(t))_i > b_i(t)$ on some set S of nonzero measure for some i, is similar. Thus F is closed in the $\sigma(L_\infty^{n_1}[0,T], L_1^{n_1}[0,T])$ topology as claimed.

The compactness of F in the $\sigma(L_\infty^{n_1}[0,T], L_1^{n_1}[0,T])$ topology given a nonempty and bounded feasible region now follows from Alaoglu's Theorem (see, for example, Holmes [15]) and the existence of an optimal extreme-point solution from another standard result (see again Holmes [15, p. 74]).

To show sequential compactness, we recall a result from functional analysis which states that if X is a separable normed linear space (that is, a normed linear space with a countable dense subset), then any norm bounded set in X^* (the dual of X) which is also closed in the $\sigma(X^*, X)$ topology is sequentially compact in the $\sigma(X^*, X)$ topology (see, for example, Kolmogorov and Fomin [16, Cor. 1, p. 203]). Now $L_1^{n_1}[0,T]$ is a separable space and $L_1^{n_1}[0,T]^* = L_\infty^{n_1}[0,T]$. Hence, as F is closed in the $\sigma(L_\infty^{n_1}[0,T], L_1^{n_1}[0,T])$ topology by the above, it is also sequentially compact in this topology. This establishes the result. $\square$

Our next result concerns the characterization of the extreme points of F. This result for SCLP may be found in Anderson, Nash, and Perold [4] and has proved to be very important in establishing most of the recent results on the problem. The result for SCLPTD that we present is very similar to that for SCLP in [4].

THEOREM 3.2. *Let $x \in F$, the feasible region of* SCLPTD, *and $\omega(t)$ the corresponding* SCLPTD *solution. Then x is an extreme point of F if and only if the columns of*

$$\mathcal{K} = \begin{bmatrix} \mathcal{G} & I & 0 \\ \mathcal{H} & 0 & I \end{bmatrix}$$

corresponding to the support of $\Omega(t)$ (that is, i such that $\Omega_i(t) > 0$) are linearly independent for almost all $t \in [0,T]$.

Proof. The proof is very similar to the proof of the result for SCLP in [4], so we will omit some of the details. Suppose x is not an extreme point of F. Then there exists $x^{(1)}, x^{(2)} \in F$, both distinct from x, such that $x(t) = (x^{(1)}(t) + x^{(2)}(t))/2$ a.e. on $[0,T]$. Let $\omega^{(1)}(t)$ and $\omega^{(2)}(t)$ be the corresponding feasible solutions for SCLPTD. If we now differentiate the constraint (2) in T2 we see that

$$\mathcal{K}\begin{bmatrix} X^{(1)}(t) \\ \dot{Y}^{(1)}(t) \\ Z^{(1)}(t) \end{bmatrix} = \mathcal{K}\begin{bmatrix} X^{(2)}(t) \\ \dot{Y}^{(2)}(t) \\ Z^{(2)}(t) \end{bmatrix} = \begin{bmatrix} \dot{A}(t) \\ B(t) \end{bmatrix},$$

a.e. on $[0, T]$. Hence the columns of $\mathcal{K}$ corresponding to the support of $\Omega(t)$ are linearly dependent on some set of nonzero measure.

Now suppose that the columns of $\mathcal{K}$ corresponding to the support of $\Omega(t)$ are linearly dependent on some set of nonzero measure. Since there are only a finite number of choices of basis for $\mathcal{K}$, and $Y(t)$ is continuous, we can choose an open interval I, a set $P \subseteq I$ of nonzero measure, and $\varepsilon > 0$ such that the support of $\Omega(t)$ is constant on P, and for all $t \in P$ we have $Y_i(t) > \varepsilon$ for all i such that $Y_i(t) > 0$ on P. Choose $q \neq 0$ with $\mathcal{K}q = 0$ and $q_i \neq 0$ only if $\Omega_i(t) > 0$ on P. We will define new feasible solutions $\Omega^{(1)}(t)$ and $\Omega^{(2)}(t)$ for T2 by

$$\tag{6} \left\{ \begin{array}{rcl} \begin{bmatrix} X^{(j)}(t) \\ \dot{Y}^{(j)}(t) \\ Z^{(j)}(t) \end{bmatrix} & = & \begin{bmatrix} X(t) \\ \dot{Y}(t) \\ Z(t) \end{bmatrix} + h_j(t)q, \quad t \in [0,1], \\ Y^{(j)}(t) & = & Y(0) + \displaystyle\int_0^t \dot{Y}^{(j)}(s)\,ds, \quad t \in [0,1], \end{array} \right.$$

for $j = 1, 2$. Define

$$f_1(t) = \min_{k \in I_1}\{\Omega_k(t)/q_k\}, \qquad I_1 \neq \emptyset,$$
$$f_2(t) = \min_{k \in I_2}\{-\Omega_k(t)/q_k\}, \qquad I_2 \neq \emptyset,$$

where

$$I_1 = \{\, k : q_k > 0,\ k \leq Tn_1,\ k > T(n_1 + n_2)\,\},$$
$$I_2 = \{\, k : q_k < 0,\ k \leq Tn_1,\ k > T(n_1 + n_2)\,\}.$$

If either I_1 or I_2 is empty, then we set the corresponding f_i to 1. Note that I_1 and I_2 cannot both be empty. Set $f(t) = \min\{f_1(t), f_2(t)\}$ for $t \in P$. We now choose disjoint subsets P_1 and P_2 of P, each of nonzero measure, such that

$$\int_{P_1} f(t)\,dt = \int_{P_2} f(t)\,dt,$$
$$\left| q_k \int_{P_j} f(t)\,dt \right| < \varepsilon, \qquad Tn_1 < k \leq T(n_1 + n_2), \quad j = 1, 2.$$

We now define

$$h_1(t) = \begin{cases} f(t), & t \in P_1, \\ -f(t), & t \in P_2, \\ 0, & \text{otherwise}, \end{cases}$$

and $h_2(t) = -h_1(t)$ for $t \in [0, T]$, and then $\Omega^{(1)}(t)$ and $\Omega^{(2)}(t)$ by (6). Then it is clear that $\Omega^{(1)}(t)$ and $\Omega^{(2)}(t)$ satisfy the constraints (2) and (4) of T2, as well as the positivity constraints. Also, for any $t \in [0, T]$ such that either $t \geq s$ or $t \leq s$ for all $s \in P$, we must have $Y^{(1)}(t) = Y^{(2)}(t) = Y(t)$. Hence the constraint (3) is satisfied as well. Thus $\Omega^{(1)}(t)$ and $\Omega^{(2)}(t)$ are feasible for T2. Moreover, $\Omega(t) = (\Omega^{(1)}(t) + \Omega^{(2)}(t))/2$ on $[0, 1]$. Hence the corresponding $x^{(1)}, x^{(2)} \in F$ satisfy $x(t) = (x^{(1)}(t) + x^{(2)}(t))/2$ on $[0, T]$, and so $x(t)$ is not an extreme point of F. □

It is not too difficult to see that when SCLPTD is specialized to DNFP, this result is equivalent to the result in Anderson [3], which, like other results about extreme points on networks, is given in terms of the absence of cycles.

4. Existence of piecewise optimal solutions. In this section we prove the analogues of the results in Pullan [25] for the problem SCLPTD. In particular, we prove that SCLPTD has a piecewise analytic optimal solution if all the problem data are piecewise analytic. The importance of such results was discussed in detail in [25]. Of most importance is that such results are necessary if we are to have any hope of solving the problem in practice, or of obtaining a solution which is practical to implement. These results for SCLP have also been instrumental in obtaining the detailed duality theory in Pullan [27], and we suspect that the same will be true here for SCLPTD. It is certainly true in some instances (see Theorem 6.13).

Unlike all the other results on SCLPTD in this paper, the result in this section may be proved by appealing to the equivalent result for SCLP, and thus it need not be proved from scratch.

THEOREM 4.1. *Suppose that the feasible region for* SCLPTD *is nonempty and bounded and that $a(t)$, $b(t)$, and $c(t)$ are piecewise analytic on* $[0, T]$ *(with $a(t)$ continuous). Then* SCLPTD *has a piecewise analytic optimal extreme-point solution. If $a(t)$ and $b(t)$ are also piecewise polynomials of degrees $n + 1$ and n, respectively, then* SCLPTD *has an optimal extreme-point solution with $x(t)$ piecewise polynomial of degree n.*

Proof. By Theorem 3.1 there exists an optimal (extreme-point) solution $\omega^*(t)$ to SCLPTD. Consider the problem T2 with the extra constraints

$$\begin{aligned} Y(0) &\equiv \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(T-1) \end{bmatrix} = \begin{bmatrix} y^*(0) \\ y^*(1) \\ \vdots \\ y^*(T-1) \end{bmatrix}, \\ Y(1) &\equiv \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(T) \end{bmatrix} = \begin{bmatrix} y^*(1) \\ y^*(2) \\ \vdots \\ y^*(T) \end{bmatrix}. \end{aligned}$$

Call this problem T3. Now $\Omega^*(t)$ is optimal for T2 and hence, as it is feasible for T3, it is also optimal for T3. But T3 is precisely an SCLP problem with equality constraints on the final value of Y, $Y(1)$. Such problems were studied in Pullan [30], where they were shown to be equivalent to SCLP itself. In particular, the existence of piecewise analytic optimal extreme-point solutions for such problems was proved in [30, Thm. 4.3]. Hence T3 has a piecewise analytic optimal extreme-point solution given piecewise analytic problem data. Since this solution must be feasible for T2, it is also optimal for T2. This can now be used to construct a piecewise analytic optimal solution for the original SCLPTD. Moreover, this solution is an extreme-point solution for SCLPTD because the characterization of extreme-point solutions for SCLPTD in Theorem 3.2 is identical to that for SCLP with equality constraints on the final value of Y (see Theorem 4.1 in [30]).

The existence of piecewise polynomial optimal extreme-point solutions when the problem data are piecewise polynomial follows from Theorem 5.1 in Pullan [25] by extending it to include SCLP with restrictions on the final value of Y (see section 4.1 in Pullan [30]). □

It is worth mentioning that Pullan [25] also gives two other important results concerning the piecewise nature of optimal solutions. Both of these can be carried across to SCLPTD without any difficulty. The first is an implicit bound on the number of breakpoints in the optimal solution. The second is a necessary condition at a breakpoint in an optimal solution (see Theorem 4.4 in [25]). We do not repeat either of these results here because the first is cumbersome to state and the second requires several extra definitions.

5. Duality. We now turn to duality for SCLPTD. Duality in optimization, and especially linear optimization, has been key to the development of efficient algorithms for their solution. This has also been observed for SCLP. In this section we state a dual problem for SCLPTD based on that for SCLP given in Pullan [24]. This dual problem includes the one for DNFP in Anderson and Philpott [8] as a special case. We then wish to study the dual in a similar manner to studying the dual for SCLP in [24]. To do this we need to transform the dual into an equivalent dual for T2, as this is the problem that closely resembles SCLP. The details of the transformation of the dual are identical to that of the primal SCLPTD in section 2, with one exception which we explain below. We do not repeat the working from section 2 here, but just state the resulting duals at the various stages of the transformation; i.e., we just state the duals of T1 and T2.

The dual problem we present for SCLPTD is:

$$\begin{aligned}
\text{SCLPTD*:}\quad & \text{maximize} \quad \int_0^T \eta(t)^T b(t)\,dt - \int_0^T d\pi(t)^T a(t) \\
(7)\quad & \text{subject to} \quad (c(t) - G^T\pi(t) - H^T\eta(t))_i - \sum_{j=1}^{n_2} f_{ji}\pi_j(t+\lambda_{ji}) \geq 0, \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad i = 1,\ldots,n_1, \\
& \eta(t) \leq 0, \text{ a.e. on } [0,T], \\
& \pi(t) \text{ monotonic increasing and right continuous} \\
& \text{on } [0,T] \text{ with } \pi(T) = 0.
\end{aligned}$$

As with SCLPTD, the problem is only defined over the interval $[0,T]$, and so it is an implicit constraint that $\pi(t) = 0$ for $t > T$. We let $\theta(t)$ denote a complete set of variables for SCLPTD* and $\psi(t)$ denote the vector of left-hand sides of (7) for $i = 1,\ldots,n_1$. Thus $\theta(t)^T = (\pi(t)^T, \eta(t)^T)$ and $\psi_i(t) = (c(t) - G^T\pi(t) - H^T\eta(t))_i - \sum_{j=1}^{n_2} f_{ji}\pi_j(t+\lambda_{ji})$. As with the primal problems, it is not difficult to see that this dual includes the dual of DNFP in Anderson and Philpott [8] as a special case.

If we now repeat the transformations of section 2 we may derive the following dual for T1.

$$\begin{aligned}
\text{T1*:}\quad & \text{maximize} \quad \int_0^T \eta(t)^T b(t)\,dt - \int_0^T d\pi(t)^T a(t) \\
& \text{subject to} \quad c(t) - G^T\pi(t) - \sum_{k=1}^{\lfloor T-t \rfloor} {F^{(k)}}^T \pi(t+k) - H^T\eta(t) \geq 0, \\
& \eta(t) \leq 0, \text{ a.e. on } [0,T], \\
& \pi(t) \text{ monotonic increasing and right continuous} \\
& \text{on } [0,T] \text{ with } \pi(T) = 0.
\end{aligned}$$

This problem is easily seen to be equivalent to SCLPTD* by a simple scaling of the variables to make the traversal times and time T integers.

Continuing with these transformations from section 2 we arrive at the following dual for T2.

$$
\begin{array}{llll}
& \text{T2*:} & \text{maximize} & \displaystyle\int_0^1 \Upsilon(t)^T B(t)\,dt - \int_0^1 d\Pi(t)^T A(t) + \Pi(0)^T d \\
(8) & & \text{subject to} & C(t) - \mathcal{G}^T\Pi(t) - \mathcal{H}^T\Upsilon(t) \geq 0, \\
& & & \Upsilon(t) \leq 0, \text{ a.e. on } [0,1], \\
& & & \Pi(t) \text{ monotonic increasing and right continuous} \\
& & & \text{on } [0,1] \text{ with } \Pi(1) = 0 \text{ and } \mathcal{D}^T\Pi(0) \geq \Pi(1-).
\end{array}
$$

We let $\Theta(t)$ denote a complete set of variables for T2* and $\Psi(t)$ denote the left-hand side of (8). Thus $\Theta(t)^T = (\Pi(t)^T, \Upsilon(t)^T)$ and $\Psi(t) = C(t) - \mathcal{G}^T\Pi(t) - \mathcal{H}^T\Upsilon(t)$.

It is clear that the dual problems T1* and T2* are equivalent by making the following connection between the variables:

$$
(9) \qquad \Theta(t) = \begin{bmatrix} \theta(t) \\ \theta(t+1) \\ \vdots \\ \theta(t+T-1) \end{bmatrix}, \qquad t \in [0,1),
$$

with $\Theta(1)^T = (\Pi(1)^T, \Upsilon(1)^T)$ given by $\Pi(1) = 0$ and $\Upsilon(1)^T = (\eta(1)^T, \eta(2)^T, \ldots, \eta(T)^T)$. Note that $\Pi(1)^T \neq (\pi(1)^T, \pi(2)^T, \ldots, \pi(T)^T)$ in general. This accounts for the added term $\Pi^T(0)^T d$ in the objective function. Indeed, using this connection between the variables we have

$$
\begin{aligned}
&\int_0^1 d\Pi(t)A(t) - \Pi(0)^T d \\
&= \sum_{i=0}^{T-1} \int_0^1 d\pi(t+i)^T(a(t+i) - a(i)) - \sum_{i=0}^{T-1} \pi(i+1)^T(a(i+1) - a(i)) \\
&\quad - \pi(0)^T a(0) \\
&= \int_0^T d\pi(t)^T a(t) - \sum_{i=0}^{T-1} \int_0^1 d\pi(t+i)^T a(i) - \sum_{i=0}^{T-1} \pi(i+1)^T(a(i+1) - a(i)) \\
&\quad - \pi(0)^T a(0) \\
&= \int_0^T d\pi(t)^T a(t) - \sum_{i=0}^{T-1} (\pi(i+1) - \pi(i))^T a(i) - \sum_{i=0}^{T-1} \pi(i+1)^T(a(i+1) - a(i)) \\
&\quad - \pi(0)^T a(0) \\
&= \int_0^T d\pi(t)^T a(t).
\end{aligned}
$$

With this connection between the variables we have a set of variables feasible (optimal) for T1* if and only if the corresponding set of variables is feasible (optimal) for T2*. We have thus derived a correspondence between variables of SCLPTD* and T2* via (9). Similar to the notation $\omega(t)$ and $\Omega(t)$, which refer to corresponding respective solutions of SCLPTD and T2, we shall now use the notation $\theta(t)$ and $\Theta(t)$ to denote the corresponding respective solutions of SCLPTD* and T2*.

We now show that the problems are true "dual" problems by showing that weak duality holds. This is based on a corresponding result for SCLP in Pullan [24, Lem. 2.1]. The connection between the variables shows that it is sufficient to prove weak duality between only one of the pairs of primal and dual problems. Here, and throughout the remainder of this paper, we use the notation $V[\text{LP}]$ to denote the optimal value of a linear program LP, with the value taken to be ∞ if LP is an infeasible minimization problem, and $-\infty$ if LP is an infeasible maximization problem.

THEOREM 5.1 (weak duality). $V[\text{SCLPTD*}] \leq V[\text{SCLPTD}]$.

Proof. We shall prove the weak duality result for T2 and argue as in [24]. Suppose that $\Omega(t)$ is feasible for T2 and $\Theta(t)$ is feasible for T2*. Then

$$\begin{aligned}
&\int_0^1 \Upsilon(t)^T B(t)\,dt - \int_0^1 d\Pi(t)^T A(t) + \Pi(0)^T d \\
&= \int_0^1 \Upsilon(t)^T (\mathcal{H}X(t) + Z(t))\,dt - \int_0^1 d\Pi(t)^T \left(\int_0^t \mathcal{G}X(s)\,ds + Y(t) - Y(0)\right) \\
&\quad + \Pi(0)^T(Y(0) - \mathcal{D}Y(1)) \\
&= \int_0^1 (\mathcal{G}^T\Pi(t) + \mathcal{H}^T\Upsilon(t))^T X(t)\,dt + \int_0^1 \Upsilon(t)^T Z(t)\,dt - \int_{[0,1)} d\Pi(t)^T Y(t) \\
&\quad - \int_{\{1\}} d\Pi(t)^T Y(t) + \int_0^1 d\Pi(t)^T Y(0) + \Pi(0)^T Y(0) - (\mathcal{D}^T\Pi(0))^T Y(1),
\end{aligned}$$

by integrating by parts (see Dunford and Schwartz [11, p. 154]). Now

$$\begin{aligned}
\int_{\{1\}} d\Pi(t)^T Y(t) &= (\Pi(1) - \Pi(1-))^T Y(1) \\
&= -\Pi(1-)^T Y(1), \\
\int_0^1 d\Pi(t)^T Y(0) &= (\Pi(1) - \Pi(0))^T Y(0) \\
&= -\Pi(0)^T Y(0).
\end{aligned}$$

Hence,

$$\begin{aligned}
&\int_0^1 \Upsilon(t)^T B(t)\,dt - \int_0^1 d\Pi(t)^T A(t) + \Pi(0)^T d \\
&= \int_0^1 (\mathcal{G}^T\Pi(t) + \mathcal{H}^T\Upsilon(t))^T X(t)\,dt + \int_0^1 \Upsilon(t)^T Z(t)\,dt - \int_{[0,1)} d\Pi(t)^T Y(t) \\
&\quad + (\Pi(1-) - \mathcal{D}^T\Pi(0))^T Y(1).
\end{aligned}$$

We now have

$$\begin{aligned}
&\int_0^1 C(t)^T X(t)\,dt - \int_0^1 \Upsilon(t)^T B(t)\,dt + \int_0^1 d\Pi(t)^T A(t) - \Pi(0)^T d \\
&= \int_0^1 \Psi(t)^T X(t)\,dt - \int_0^1 \Upsilon(t)^T Z(t)\,dt + \int_{[0,1)} d\Pi(t)^T Y(t) \\
&\quad + (\mathcal{D}^T\Pi(0) - \Pi(1-))^T Y(1) \\
&\geq 0,
\end{aligned}$$

by the feasibility of $\Omega(t)$ and $\Theta(t)$. This establishes the result. □

As with SCLP, further study of duality for SCLPTD requires an algorithm. In the next section we give such an algorithm in that it gives a strong duality result as a corollary from which a more detailed study of duality could proceed.

6. A class of convergent algorithms and a strong duality result. In this section we mimic the main results from Pullan [24, 29] to produce a class of convergent algorithms for T2, and hence for SCLPTD. As in [24], we have a strong duality theorem as a corollary. We work under assumptions on the problem data similar to those in [24, 29]. In particular, we assume that the following holds throughout the remainder of this section.

Assumption 6.1. *The costs, $c(t)$, are piecewise linear, $a(t)$ is piecewise linear and continuous, $b(t)$ is piecewise constant, and the feasible region for* SCLPTD *is nonempty and bounded.*

Before beginning the discussion we give a few definitions based on concepts used in the literature on SCLP.

Definition 6.1.

1. *The* breakpoints *of a piecewise linear or piecewise constant function are the discontinuities in either the function or its derivative.*
2. *We define the* initial breakpoint partition *to be the smallest partition of* $[0, 1]$ *consisting of all the breakpoints of $A(t)$, $B(t)$, and $C(t)$.*
3. *Let $\omega(t)$ be a feasible solution for* SCLPTD *such that $x(t)$ is piecewise constant on $[0, T]$. We define the* breakpoint partition for $\Omega(t)$ *to be the partition of* $[0, 1]$ *consisting of all the breakpoints of $\Omega(t)$ and the points in the initial breakpoint partition.*
4. *Let f be any real valued function. We use the notation $f(t-)$ to denote* $\lim_{s\uparrow t} f(s)$ *and $f(t+)$ to denote* $\lim_{s\downarrow t} f(s)$ *when these limits exist.*

In [24], two discretizations DP and AP were introduced for SCLP. We introduce counterparts for these discretizations here, called DPTD and APTD, which we write in a form resembling DP and AP in the later work on SCLP (e.g., Pullan [27, 28]). It will be seen that all the results between DP, AP, and SCLP carry over to DPTD, APTD, and SCLPTD (or, more correctly, T2) without any difficulty. Let $P = \{t_0, t_1, \dots, t_m\}$ be any partition of $[0, 1]$ which is a refinement of the initial breakpoint partition. We now define

$$\begin{array}{lll}
\mathrm{DPTD}(P): & \text{minimize} & \displaystyle\sum_{i=1}^{m} C(u_i)^T \hat{X}(t_{i-1}+) \\
& \text{subject to} & \mathcal{G}\hat{X}(t_{i-1}+) + \hat{Y}(t_i) - \hat{Y}(t_{i-1}) = A(t_i) - A(t_{i-1}), \\
& & \qquad\qquad\qquad\qquad\qquad\qquad i = 1, \dots, m, \\
& & \hat{Y}(t_0) - \mathcal{D}\hat{Y}(t_m) = d, \\
& & \mathcal{H}\hat{X}(t_{i-1}+) + \hat{Z}(t_{i-1}+) = (t_i - t_{i-1})B(t_{i-1}+), \\
& & \qquad\qquad\qquad\qquad\qquad\qquad i = 1, \dots, m, \\
& & \hat{X}(t_{i-1}+), \hat{Z}(t_{i-1}+) \geq 0,\ i = 1, \dots, m, \\
& & \hat{Y}(t_i) \geq 0,\ i = 0, \dots, m,
\end{array}$$

and

$$\mathrm{APTD}(P): \quad \text{minimize} \quad \sum_{i=1}^{m} \Big[C(t_{i-1}+)^T \hat{X}(t_{i-1}+) + C(t_i-)^T \hat{X}(t_i-) \Big]$$

$$\begin{aligned}\text{subject to}\quad & \mathcal{G}\hat{X}(t_{i-1}+)+\hat{Y}(u_i)-\hat{Y}(t_{i-1})=A(u_i)-A(t_{i-1}),\\ &\qquad\qquad\qquad\qquad\qquad\qquad i=1,\dots,m,\\ & \mathcal{G}\hat{X}(t_i-)+\hat{Y}(t_i)-\hat{Y}(u_i)=A(t_i)-A(u_i),\\ &\qquad\qquad\qquad\qquad\qquad\qquad i=1,\dots,m-1,\\ & \hat{Y}(t_0)-\mathcal{D}\hat{Y}(t_m)=d,\\ & \mathcal{H}\hat{X}(t_{i-1}+)+\hat{Z}(t_{i-1}+)=\tau_i B(t_{i-1}+),\quad i=1,\dots,m,\\ & \mathcal{H}\hat{X}(t_i-)+\hat{Z}(t_i-)=\tau_i B(t_i-),\quad i=1,\dots,m,\\ & \hat{X}(t_{i-1}+),\hat{X}(t_i-),\hat{Y}(u_i),\hat{Z}(t_{i-1}+),\hat{Z}(t_i-)\ge 0,\\ &\qquad\qquad\qquad\qquad\qquad\qquad i=1,\dots,m,\\ & \hat{Y}(t_i)\ge 0,\quad i=0,\dots,m,\end{aligned}$$

where

$$\begin{aligned}u_i&=\frac{t_{i-1}+t_i}{2},\\ \tau_i&=\frac{t_i-t_{i-1}}{2}.\end{aligned}$$

We let $\hat{\Omega}_D$ and $\hat{\Omega}$ denote a complete set of variables for DPTP(P) and APTD(P), respectively.

Strictly speaking, these are really discretizations of T2, rather than the original SCLPTD. However, it is these discretizations that we shall study, just as it was the dual problem T2* rather than SCLPTD* that was studied in the previous section. The reason we do not bother with the "true" discretizations for SCLPTD is that there is little to be gained from this, and that they are also very cumbersome to state, especially if the traversal times are nonintegers. In the case where SCLPTD has integer traversal times and final time T, it is possible to write down the discretizations for T1 fairly easily. For instance, the "true" DPTD for T1 is

$$\begin{aligned}\text{minimize}\quad & \sum_{k=0}^{T-1}\sum_{i=1}^{m} c(u_i+k)^T\hat{x}((t_{i-1}+k)+)\\ \text{subject to}\quad & G\hat{x}(t_0+)+\hat{y}(t_1)=a(t_1),\\ & G\hat{x}((t_{i-1}+k)+)+\sum_{l=1}^{k}F^{(l)}\hat{x}((t_{i-1}+k-l)+)+\hat{y}(t_i+k)-\hat{y}(t_{i-1}+k)\\ &\qquad =a(t_i+k)-a(t_{i-1}+k),\\ &\qquad\qquad\qquad i=1,\dots,m,\ k=0,\dots,T-1,\ (i,k)\ne(0,0),\\ & H\hat{x}((t_{i-1}+k)+)+\hat{z}((t_{i-1}+k)+)=(t_i-t_{i-1})b(t_{i-1}+k)+),\\ &\qquad\qquad\qquad\qquad i=1,\dots,m,\ k=0,\dots,T-1,\\ & \hat{x}((t_{i-1}+k)+),\hat{z}((t_{i-1}+k)+)\ge 0,\quad i=1,\dots,m,\ k=0,\dots,T-1,\\ & \hat{y}(t_i+k),\quad i=0,\dots,m,\ k=0,\dots,T-1,\ (i,k)\ne(0,0).\end{aligned}$$

This is no more than a rewriting of a finite-dimensional linear program by giving different names to some of the variables, and would therefore involve no real difference when formulating on a computer. In contrast, our transformations of SCLPTD and SCLPTD* involved rewriting continuous problems which would not be directly represented on a computer. For these reasons, we treat the discretizations above as the appropriate discretizations of the original problem.

It is not difficult to see that DPTP and APTD have the same properties in relation to SCLPTD (or, more correctly, T2) as DP and AP, respectively, have for SCLP. In particular, there is a natural correspondence between feasible solutions of SCLPTD, DPTD, and APTD. This is made precise by the following definition.

DEFINITION 6.2. *Let $P = \{t_0, t_1, \ldots, t_m\}$ be any refinement of the initial breakpoint partition. Suppose that $\omega(t)$ is feasible for* SCLPTD *with $X(t)$ piecewise constant with breakpoints in P. We say that $\hat{\Omega}_D$ defined by*

$$\begin{aligned} \hat{X}(t_{i-1}+) &= (t_i - t_{i-1})X(t_{i-1}+), \quad i = 1, \ldots, m, \\ \hat{Y}(t_i) &= Y(t_i), \quad i = 0, \ldots, m, \\ \hat{Z}(t_{i-1}+) &= (t_i - t_{i-1})Z(t_{i-1}+), \quad i = 1, \ldots, m, \end{aligned}$$

is the natural solution *for* DPTD(P) (*constructed from* $\omega(t)$). *Similarly, we say that $\hat{\Omega}$ defined by*

$$\begin{aligned} \hat{X}(t_{i-1}+) &= \tau_i X(t_{i-1}+), \quad i = 1, \ldots, m, \\ \hat{X}(t_i-) &= \tau_i X(t_i-), \quad i = 1, \ldots, m, \\ \hat{Y}(t_i) &= Y(t_i), \quad i = 0, \ldots, m, \\ \hat{Y}(u_i) &= Y(u_i), \quad i = 1, \ldots, m, \\ \hat{Z}(t_{i-1}+) &= \tau_i Z(t_{i-1}+), \quad i = 1, \ldots, m, \\ \hat{Z}(t_i-) &= \tau_i Z(t_i-), \quad i = 1, \ldots, m, \end{aligned}$$

is the natural solution *for* APTD(P) (*constructed from* $\omega(t)$).

Conversely, suppose now that $\hat{\Omega}_D$ is any feasible solution for DPTD(P); *then we say that $\omega(t)^T = (x(t)^T, y(t)^T, z(t)^T)$, defined by*

$$\begin{aligned} X(t) &= \begin{cases} \dfrac{1}{t_i - t_{i-1}} \hat{X}(t_{i-1}+), & t \in [t_{i-1}, t_{i-1}), \quad i = 1, \ldots, m, \\ \dfrac{1}{t_i - t_{i-1}} \hat{X}(t_m-), & t = 1, \end{cases} \\ Y(0) &= \hat{Y}(0), \end{aligned}$$

and with $Y(t)$ and $Z(t)$ given by the constraints of T2 (*i.e., satisfying* (2) *and* (4)), *is the* natural solution *for* SCLPTD (*constructed from* $\hat{\Omega}_D$). *Similarly, suppose that $\hat{\Omega}$ is any feasible solution for* APTD(P); *then we say that $\omega(t)$, defined by*

$$\begin{aligned} X(t) &= \begin{cases} \dfrac{1}{\tau_i} \hat{X}(t_{i-1}+), & t \in [t_{i-1}, u_i), \quad i = 1, \ldots, m, \\ \dfrac{1}{\tau_i} \hat{X}(t_i-), & t \in [u_i, t_i), \quad i = 1, \ldots, m, \\ \dfrac{1}{\tau_m} \hat{X}(t_m-), & t = T, \end{cases} \\ Y(0) &= \hat{Y}(0), \end{aligned}$$

and with $Y(t)$ and $Z(t)$ given by the constraints of T2*, is the* natural solution *for* SCLPTD (*constructed from* $\hat{\Omega}$).

We now have the following relationships. We omit the proofs, as they are quite trivial, and anyway, they are identical to those in [24].

THEOREM 6.3. *Suppose that $\omega(t)$ is feasible for* SCLPTD *with $x(t)$ piecewise constant. Let P be any refinement of the breakpoint partition for $\Omega(t)$. Then the* *natural solution* $\hat{\Omega}_D$ *for* DPTD(P) *is feasible for* DPTD(P), *and the objective function values of the two solutions are the same in their respective linear programs.*

Conversely, let P *be any refinement of the initial breakpoint partition and suppose that* $\hat{\Omega}_D$ *is feasible for* DPTD(P). *Then the natural solution* $\omega(t)$ *for* SCLPTD *is feasible for* SCLPTD, *and the objective function values of the two solutions are the same in their respective linear programs.*

THEOREM 6.4. *Suppose that* $\omega(t)$ *is feasible for* SCLPTD *with* $x(t)$ *piecewise constant. Let* P *be any refinement of the breakpoint partition for* $\Omega(t)$. *Then the natural solution* $\hat{\Omega}$ *for* APTD(P) *is feasible for* APTD(P), *and the objective function values of the two solutions are the same in their respective linear programs.*

Conversely, let $P = \{t_0, t_1, \ldots, t_m\}$ *be any refinement of the initial breakpoint partition and suppose that* $\hat{\Omega}$ *is feasible for* APTD(P). *Then the natural solution* $\omega(t)$ *for* SCLPTD *is feasible for* SCLPTD, *and the difference in the values of the objective function is given by*

$$\alpha(\hat{\Omega}) \equiv \hat{C}^T\hat{\Omega} - \int_0^T c(t)x(t)\,dt = \sum_{i=1}^m \frac{\tau_i}{4}(\hat{X}(t_i-) - \hat{X}(t_{i-1}+))^T\dot{C}(t_i-). \tag{10}$$

We now turn to the relationships between the dual problems. Now the standard linear programming dual of APTD(P) is

$$\begin{aligned}
\text{maximize}\quad & \sum_{i=1}^m (\hat{\Pi}(t_{i-1}+) + \hat{\Pi}(t_i-))^T(A(t_i) - A(u_i)) \\
& + \sum_{i=1}^m \tau_i(\hat{\Upsilon}(t_{i-1}+) + \hat{\Upsilon}(t_i-))^T B(t_i-) + \sigma^T d \\
\text{subject to}\quad & C(t_i-) - \mathcal{G}^T\hat{\Pi}(t_i-) - \mathcal{H}^T\hat{\Upsilon}(t_i-) \geq 0, \quad i = 1, \ldots, m, \\
& C(t_{i-1}+) - \mathcal{G}^T\hat{\Pi}(t_{i-1}+) - \mathcal{H}^T\hat{\Upsilon}(t_{i-1}+) \geq 0, \quad i = 1, \ldots, m, \\
& \hat{\Upsilon}(t_i-), \hat{\Upsilon}(t_{i-1}+) \leq 0, \qquad i = 1, \ldots, m, \\
& \hat{\Pi}(t_i-) - \hat{\Pi}(t_{i-1}+) \geq 0, \qquad i = 1, \ldots, m, \\
& \hat{\Pi}(t_i+) - \hat{\Pi}(t_i-) \geq 0, \qquad i = 1, \ldots, m-1, \\
(11)\quad & \hat{\Pi}(t_0+) - \sigma \geq 0, \\
(12)\quad & \hat{\Pi}(t_m-) - \mathcal{D}^T\sigma \leq 0.
\end{aligned}$$

We eliminate the variable σ from the above dual. First, to eliminate it from the constraints we recall the definition of $\mathcal{D}$, in particular that $\mathcal{D}$ has nonnegative elements, to combine (11) and (12). This gives $\mathcal{D}^T\hat{\Pi}(t_0+) \geq \hat{\Pi}(t_m-)$. Second, to eliminate σ from the objective function, we note that $d^T = (a(0)^T, 0^T, \ldots, 0^T) \geq 0$ if SCLPTD is to have a feasible solution. Now if $d_i > 0$, then any optimal solution to the above dual must have $\sigma_i = \hat{\Pi}(t_0+)$. If $d_i = 0$, then the objective function of the above dual is independent of σ_i. Hence, in this case as well, given an optimal solution to the above dual exists, there is an optimal solution to the above problem with $\sigma_i = \hat{\Pi}(t_0+)$. We can thus replace the term $\sigma^T d$ in the objective function by $\hat{\Pi}(t_0+)^T d$. Hence we can rewrite the above dual in the following equivalent form:

$$\text{APTD*}(P):\quad \text{maximize}\quad \sum_{i=1}^m (\hat{\Pi}(t_{i-1}+) + \hat{\Pi}(t_i-))^T(A(t_i) - A(u_i))$$

$$
\begin{aligned}
&\quad + \sum_{i=1}^{m} \tau_i (\hat{\Upsilon}(t_{i-1}+) + \hat{\Upsilon}(t_i-))^T B(t_i-) + \hat{\Pi}(t_0+)^T d \\
\text{subject to} \quad & C(t_i-) - \mathcal{G}^T \hat{\Pi}(t_i-) - \mathcal{H}^T \hat{\Upsilon}(t_i-) \geq 0, \quad i = 1, \dots, m, \\
& C(t_{i-1}+) - \mathcal{G}^T \hat{\Pi}(t_{i-1}+) - \mathcal{H}^T \hat{\Upsilon}(t_{i-1}+) \geq 0, \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad i = 1, \dots, m, \\
& \hat{\Upsilon}(t_i-), \hat{\Upsilon}(t_{i-1}+) \leq 0, \qquad i = 1, \dots, m, \\
& \hat{\Pi}(t_i-) - \hat{\Pi}(t_{i-1}+) \geq 0, \qquad i = 1, \dots, m, \\
& \hat{\Pi}(t_i+) - \hat{\Pi}(t_i-) \geq 0, \qquad i = 1, \dots, m-1, \\
& \mathcal{D}^T \hat{\Pi}(t_0+) \geq \hat{\Pi}(t_m-).
\end{aligned}
$$

We use the notation $\hat{\Theta}$ to denote a complete set of variables for APTD*(P).

Now in [24] it was observed that AP*(P), the dual of the discretization AP(P) for SCLP, was a discretization of SCLP*, the dual of SCLP. The only differences between the duals of SCLP* and T2* are the extra constraints $\mathcal{D}^T \Pi(0) \geq \Pi(1-)$ and the added term $\Pi(0)^T d$ in the objective function. Similarly, the only differences between AP*(P) and APTD*(P) are the extra constraints $\mathcal{D}^T \hat{\Pi}(t_0+) \geq \hat{\Pi}(t_m-)$ and the added term $\hat{\Pi}(t_0+)^T d$ in the objective function (there is also an additional term $\Pi(t_0+)^T A(t_0)$ missing in the objective function of APTD*(P), however, $A(t_0) = A(0) = 0$ by definition). Thus it is not difficult to see that APTD*(P) is a discretization of T2* and any result that is true between AP*(P) and SCLP* is also true between APTD*(P) and T2*. We now recall these results [24] and restate them in terms of SCLPTD. Before doing this we define the natural connection between solutions to the dual problems T2* and APTD*(P).

Definition 6.5. *Let $P = \{t_0, t_1, \dots, t_m\}$ be any refinement of the initial breakpoint partition. Suppose that $\theta(t)$ is feasible for* SCLPTD* *and $\Theta(t)$ is piecewise linear with breakpoints in P. We say that $\hat{\Theta}$, defined by*

$$
\begin{aligned}
\hat{\Theta}(t_{i-1}+) &= \Theta(t_{i-1}+), \quad i = 1, \dots, m, \\
\hat{\Theta}(t_i-) &= \Theta(t_{i-1}-), \quad i = 1, \dots, m, \\
\hat{\Theta}(1) &= \Theta(1),
\end{aligned}
$$

is the natural solution *for* APTD*(P) (*constructed from $\theta(t)$*).

Conversely, suppose now that $\hat{\Theta}$ is any feasible solution for APTD*(P). *Then we say that $\theta(t)$, defined by*

$$
\Theta(t) = \left(\frac{t_i - t}{t_i - t_{i-1}} \right) \hat{\Theta}(t_{i-1}+) + \left(\frac{t - t_{i-1}}{t_i - t_{i-1}} \right) \hat{\Theta}(t_i-)
$$

for $t \in [t_{i-1}, t_i)$ and $i = 1, \dots, m$, and $\Theta(1)^T = (\Pi(1)^T, \Upsilon(1)^T)$ given by $\Pi(1) = 0$ and $\Upsilon(1) = \hat{\Upsilon}(t_m-)$, is the natural solution *for* SCLPTD* (*constructed from $\hat{\Theta}$*).

We now have the following results.

Theorem 6.6. *Let P be any refinement of the initial breakpoint partition. Suppose that $\theta(t)$ is feasible for* SCLPTD* *and is piecewise linear with breakpoints in P. Then the natural solution $\hat{\Theta}$ for* APTD*(P) *is feasible for* APTD*(P), *and the objective function values of the two solutions are the same in their respective linear programs.*

Conversely, suppose that $\hat{\Theta}$ is feasible for APTD*(P). *Then the natural solution $\theta(t)$ for* SCLPTD* *is feasible for* SCLPTD*, *and the objective function values of the two solutions are the same in their respective linear programs.*

Theorem 6.7. *Let P be any partition of $[0,T]$ which contains the breakpoints of the problem data. Then $V[\text{APTD}(P)] \leq V[\text{SCLPTD*}] \leq V[\text{SCLPTD}]$.*

Theorem 6.8. *Suppose that $\omega(t)$ is feasible for* SCLPTD *and that the corresponding $\hat{\Omega}$ is optimal for* APTD(P). *Then $\omega(t)$ is optimal for* SCLPTD.

Having discussed the properties of the discretizations DPTD(P) and APTD(P) we now extend the improvement step given in Pullan [24] to SCLPTD. Again, the extension is quite trivial, so in most cases we merely state the results. As with the work on SCLP, this step will form the basis of a class of algorithms for solving SCLPTD.

Let $\omega(t)$ be a feasible solution for SCLPTD such that $x(t)$ is piecewise constant and P any refinement of the breakpoint partition for $\Omega(t)$. Let $\hat{\Omega}$ be the natural solution for APTD(P). If $\hat{\Omega}$ is optimal for APTD(P), then by Theorem 6.8, $\omega(t)$ is optimal for SCLPTD. Otherwise we may construct an improved feasible solution $\hat{\hat{\Omega}}$ for APTD(P). Let

$$\delta = \hat{C}^T \hat{\hat{\Omega}} - \hat{C}^T \hat{\Omega};$$

then $\delta < 0$. Let $\tilde{\omega}(t)$ be the natural solution for SCLPTD constructed from $\hat{\hat{\Omega}}$. Choose $\varepsilon \in [0,1]$ and set $\varepsilon_i = \tau_i \varepsilon$. We now define a new feasible solution $\bar{\omega}_\varepsilon(t)$ for SCLPTD by

$$\begin{aligned} \bar{X}_\varepsilon(t) &= \begin{cases} \tilde{X}(t), & t \in [t_{i-1}, t_{i-1}+\varepsilon_i) \cup [t_i - \varepsilon_i, t_i), \quad i = 1, \dots, m, \\ X(t), & \text{otherwise}, \end{cases} \\ \bar{Y}_\varepsilon(0) &= (1-\varepsilon)Y(0) + \varepsilon \tilde{Y}(0), \end{aligned}$$

with $\bar{Y}_\varepsilon(t)$ and $\bar{Z}_\varepsilon(t)$ derived from the constraints of T2. We refer to this as *patching $\omega(t)$ and $\tilde{\omega}(t)$ together.* The feasibility of $\bar{\omega}_\varepsilon(t)$ follows from the following theorem, the proof of which is identical to the equivalent result in [24] (Lemma 4.1).

Theorem 6.9.

$$\begin{aligned} \bar{Y}_\varepsilon(t_i) &= (1-\varepsilon)Y(t_i) + \varepsilon \tilde{Y}(t_i), \quad i = 0, \dots, m, \\ \bar{Y}_\varepsilon(t_{i-1}+\varepsilon_i) &= (1-\varepsilon)Y(t_{i-1}) + \varepsilon \tilde{Y}(u_i), \quad i = 1, \dots, m, \\ \bar{Y}_\varepsilon(t_i-\varepsilon_i) &= (1-\varepsilon)Y(t_i) + \varepsilon \tilde{Y}(u_i), \quad i = 1, \dots, m. \end{aligned}$$

Corollary 6.10. *$\bar{\omega}_\varepsilon(t)$ is feasible for* SCLPTD *for all $\varepsilon \in [0,1]$.*

Proof. By definition, $\bar{\Omega}_\varepsilon(t)$ satisfies all the constraints of T2 except possibly (3) and the positivity constraints. However, the former is satisfied because it is satisfied for both $\Omega(t)$ and $\tilde{\Omega}(t)$, and the above theorem shows that $\bar{Y}_\varepsilon(0) = (1-\varepsilon)Y(0) + \varepsilon\tilde{Y}(0)$ and $\bar{Y}_\varepsilon(1) = (1-\varepsilon)Y(1) + \varepsilon\tilde{Y}(1)$. The positivity of $\bar{\Omega}_\varepsilon(t)$ also follows from the above theorem and the positivity of $\Omega(t)$ and $\tilde{\Omega}(t)$. □

Not only do we obtain a new feasible solution, but this solution also gives an improvement over $\omega(t)$ in objective function value for appropriately chosen ε. Again the proof is identical to corresponding ones in [24] (Lemma 4.3 and Corollary 4.5).

Theorem 6.11. *For ε sufficiently small, $\int_0^T c(t)^T \bar{x}_\varepsilon(t)\,dt < \int_0^T x(t)^T x(t)\,dt$ and*

$$\min_\varepsilon \int_0^T c(t)^T \bar{x}_\varepsilon(t)\,dt - \int_0^T c(t)^T x(t)\,dt = \begin{cases} \dfrac{\delta^2}{4\alpha}, & \alpha < 0 \textit{ and } \dfrac{\delta}{2\alpha} < 1, \\ \delta - \alpha, & \textit{otherwise}, \end{cases}$$

where $\alpha = \alpha(\hat{\tilde{\Omega}})$ *is given by* (10), *and occurs at*

$$\varepsilon^* = \begin{cases} \dfrac{\delta}{2\alpha}, & \alpha < 0 \textit{ and } \dfrac{\delta}{2\alpha} < 1, \\ 1, & \textit{otherwise.} \end{cases}$$

We refer to patching $\omega(t)$ and $\tilde{\omega}(t)$ together with $\varepsilon = \varepsilon^*$ above as *patching* $\omega(t)$ *and* $\tilde{\omega}(t)$ *together optimally.*

We now have the following result.

THEOREM 6.12. *Let* $\omega(t)$ *be an optimal solution for* SCLPTD *and* P *be any refinement of the breakpoint partition for* $\Omega(t)$. *Then the natural solution* $\hat{\Omega}$ *is optimal for* APTD(P).

Proof. If not, then the algorithm above constructs an improved feasible solution for T2 which gives an improved feasible solution for SCLPTD. □

As with [24], strong duality under Assumption 6.1 is now immediate.

THEOREM 6.13 (strong duality). *Under Assumption* 6.1, V[SCLPTD*] $=$ V[SCLPTD] *and there exist a piecewise linear optimal solution for* SCLPTD* *and an optimal extreme-point solution for* SCLPTD *in which* $x(t)$ *is piecewise constant.*

Proof. The existence of the appropriate optimal solution $\omega(t)$ for SCLPTD is given by Theorem 4.1. Let P be the breakpoint partition for $\Omega(t)$. By Theorem 6.12 above, the natural solution is optimal for APTD(P). The result now follows by the strong duality theorem for ordinary finite-dimensional linear programming and Theorems 6.6 and 6.7. □

In Pullan [27] the equivalent result for SCLP was used as a starting point for the development of an extensive duality theory for SCLP. In particular, strong duality was proved under the assumption of piecewise analytic problem data. Such a result is probably true for SCLPTD, however we do not pursue this matter here, as the proof of the strong duality result in [27] is rather long and technical, and so a study of duality for SCLPTD is best left as a topic for future research.

Instead, we now give a class of convergent algorithms for SCLPTD under Assumption 6.1 based on the patching-together process above. The algorithms are similar to those for SCLP in Pullan [29]. In particular, we propose the following class of algorithms for SCLPTD.

0. Let P_1 be the initial breakpoint partition and $\omega^{(0)}(t)$ be any feasible solution for SCLPTD such that $\Omega^{(0)}(t)$ has breakpoints in P_1. Let $\hat{\Omega}^{(0)}$ be the natural solution for APTD(P_1). Set $n = 1$.
1. If $\hat{\Omega}^{(n-1)}$ is optimal for APTD(P_n) then stop as $\omega^{(n-1)}(t)$ is optimal for SCLPTD (Theorem 6.8).
2. Optimize APTD(P_n) to produce $\hat{\tilde{\Omega}}^{(n)}$. Let $\tilde{\omega}^{(n)}(t)$ be the natural solution for SCLPTD.
3. Patch $\Omega^{(n-1)}(t)$ and $\tilde{\Omega}^{(n)}(t)$ together optimally to produce the improved solution $\bar{\omega}^{(n)}(t)$ for SCLPTD.
4. Perform any other step to produce a feasible solution $\omega^{(n)}(t)$ for SCLPTD whose objective function value is at least as good as that of $\bar{\omega}^{(n)}(t)$.
5. Let P_{n+1} be any refinement of the breakpoint partition for $\Omega^{(n)}(t)$. Let $\hat{\Omega}^{(n)}$ be the natural solution for APTD(P_{n+1}). Set $n = n + 1$ and return to step 1.

We refer the reader to Pullan [29] for some suggestions for the general step 4.

The convergence of this general class of algorithms for SCLP was proved in [29]. The proof used the results from Pullan [24], most significantly, the formulae for the difference in objective function values of various solutions. In this section we have given direct equivalents of all these results for SCLPTD. In particular, the formulae in Theorems 6.4 and 6.11 are identical to those in [24] (that is, when the discretization APTD(P) is rewritten in the form of AP(P) in Pullan [24], rather than in the form of AP(P) in Pullan [27] or Pullan [28]). Hence, convergence of the algorithm above for SCLPTD follows by exactly the same proof as in [29]. We thus have the following result.

THEOREM 6.14. *The algorithm above for* SCLPTD *converges for any implementation of step* 4, *i.e., either the algorithm converges in a finite number of steps with an optimal solution, or*

$$\lim_{n\to\infty}\int_0^T c(t)x^{(n)}(t)\,dt = V[\text{SCLPTD}].$$

7. Remarks.

We now comment on the results obtained in this paper. From a theoretical point of view they have been very satisfactory. It has been observed that any theorem for SCLP can be extended with ease to give a similar theorem for SCLPTD. While we have not extended all the results from SCLP, most notably the extensive duality results in Pullan [27], we have no reason to suppose that they cannot be extended. The main reason for not pursuing such matters here is to keep the length of this paper to a manageable size.

There have also been several omissions in the extension of the work of SCLP to SCLPTD from a more algorithmic point of view. Again, we envisage no problems with such extensions. The omissions to which we are referring appear in Pullan [28], Anderson and Pullan [9], Philpott and Craddock [23], and Pullan [30].

The first paper, Pullan [28], is concerned with an extended algorithm for SCLP under the weaker assumption that the costs are general piecewise analytic functions, rather than just piecewise linear. The algorithm in [28] is simplex-like in nature and relies heavily on the duality theory in [27], which thus prevents any discussion of such an algorithm for SCLPTD here.

The second paper, Anderson and Pullan [9], contains a purification algorithm for SCLP, that is, an algorithm whereby a nonextreme-point solution is converted into an extreme-point solution for SCLP without increasing the value of the objective function. It has been observed that the use of a purification algorithm may be desirable in solving SCLP efficiently under the standard assumption, Assumption 6.1, and, in any case, it is necessary in the extended algorithm with piecewise analytic costs. Thus purification for SCLPTD deserves attention.

The third paper, Philpott and Craddock [23], is concerned with a different class of algorithms for solving SCLP than those in Pullan [24, 29] (although, strictly speaking, [23] is only concerned with CNP). The authors call this class of algorithms adaptive discretization algorithms. This is because they use the properties of the discretization AP for SCLP to insert breakpoints at favorable times, thereby adapting the discretization solved at each stage. The methods appear to give a fairly efficient solution procedure. Similar adaptive discretization algorithms for SCLPTD, using the properties of the discretization APTD, should be possible without much difficulty.

The final paper, Pullan [30], discusses a possible improved algorithm for SCLP based on the previous work in Pullan [24, 29]. The initial study of this improved algorithm suggests that it is very efficient. Again, a similar algorithm for SCLPTD should be possible. The paper Pullan [30] also studies several SCLP-like problems which are converted into special cases of SCLP. They include SCLP with linear costs on $y(t)$, and SCLP with linear constraints on either $y(t)$ or just $y(T)$. Equivalent results in this area should also be possible for SCLPTD.

The results that we have presented in this paper of an algorithmic nature also deserve further comment. In particular, we have extended the results in Pullan [24, 29] without difficulty, and it may appear that we can solve SCLPTD with the same ease as SCLP. However, we have to be a bit more cautious here. This is because the algorithm is basically an algorithm for solving T2. A quick review of how this problem was constructed will reveal that this problem could be very large in general. The first stage of the transformation is to convert rational traversal times into integers by scaling the time variable by some common denominator. This could make T very large. The second stage then scales the dimensions of the variables by T, and so the dimensions of the variables could also become very large. However, things might not be as bad as they first seem.

First, in all the literature on discrete models of such problems, the traversal times are integers. Thus no scaling is required in a continuous model of such a problem. Second, the discrete model is always solved by solving the time-expanded network problem, which is essentially DPTD(P) with P set to the initial breakpoint partition. The initial stages of our algorithm above will involve solving discretizations of roughly the same size and, therefore, complexity. It is in the latter stages of the algorithm, as more breakpoints are introduced, that we could have problems. However, if T is fairly large to begin with (that is, in comparison with the traversal times), then we would not expect the algorithm to introduce too many new breakpoints in order to produce a near optimal solution, other than the points $1, 2, 3, \dots, T$ contained implicitly in T2. Thus, in this case, we would expect that the time required to solve the continuous problem is not excessively more than that required to solve the equivalent static one. In any case, any algorithm for SCLPTD along the lines of those for SCLP developed in previous work needs to be very careful in maintaining as few breakpoints as possible in the current partition. We suspect that the extension of the algorithm in Pullan [30] to SCLPTD offers the best chance of this. Any further discussion along these lines obviously requires some numerical work.

Finally, we return to the starting point of this paper, namely network problems. As was mentioned in the introduction, the general nature of SCLPTD meant that the network structure of the problem was lost in the discussion. In order to regain this, we could consider the results of this paper when SCLPTD is restricted to various network problems, such as DNFP. Without going through several network problems in detail, the main general comment that we can make concerns the algorithms of the previous sections. This comment is that, in general, the discretizations DPTD and APTD have the same structure as the continuous problem. Thus these discretizations are single-commodity static network programs for the continuous problem DNFP. Similarly, if the continuous problem is of a multicommodity or generalized network type, then the discretizations will be static multicommodity or generalized network programs, respectively. Thus they can be solved efficiently using the specialized network algorithms available for such problems.

REFERENCES

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, NJ, 1993.

[2] E. J. Anderson, *A Continuous Model For Job-Shop Scheduling*, Ph.D. thesis, University of Cambridge, UK, 1978.

[3] E. J. Anderson, *Extreme-points for continuous network programs with arc delays*, J. Inform. Optim. Sci., 10 (1989), pp. 45–52.

[4] E. J. Anderson, P. Nash, and A. F. Perold, *Some properties of a class of continuous linear programs*, SIAM J. Control Optim., 21 (1983), pp. 758–765.

[5] E. J. Anderson, P. Nash, and A. B. Philpott, *A class of continuous network flow problems*, Math. Oper. Res., 7 (1982), pp. 501–514.

[6] E. J. Anderson and A. B. Philpott, *A continuous-time network simplex algorithm*, Networks, 19 (1989), pp. 395–425.

[7] E. J. Anderson and A. B. Philpott, *On the solutions of a class of continuous linear programs*, SIAM J. Control Optim., 32 (1994), pp. 1289–1296.

[8] E. J. Anderson and A. B. Philpott, *Optimisation of flows in networks over time*, in Probability, Statistics and Optimisation, F. P. Kelly, ed., J. Wiley and Sons, Chichester, UK, 1994, pp. 369–382.

[9] E. J. Anderson and M. C. Pullan, *Purification for separated continuous linear programs*, Z. Oper. Res., 43 (1996), pp. 9–33.

[10] L. G. Chalmet, R. L. Francis, and P. B. Saunders, *Network models for building evacuation*, Management Sci., 28 (1982), pp. 86–105.

[11] N. Dunford and J. T. Schwartz, *Linear Operators Part I: General Theory*, Wiley-Interscience, New York, 1988.

[12] W. H. Farr and M. A. Hanson, *Continuous time programming with nonlinear time-delayed constraints*, J. Math. Anal. Appl., 46 (1974), pp. 41–61.

[13] L. R. Ford and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, NJ, 1962.

[14] H. Frank, *Data communication networks*, IEEE Trans. Comm. Tech., 15 (1967), pp. 156–163.

[15] R. B. Holmes, *Geometric Functional Analysis and Its Applications*, Springer-Verlag, New York, 1975.

[16] A. N. Kolmogorov and S. V. Fomin, *Introductory Real Analysis*, Dover, New York, 1970.

[17] S. E. Lovetskii and I. I. Melamed, *Dynamic flows in networks*, Automat. Remote Control, 48 (1987), pp. 1417–1434.

[18] F. H. Moss and A. Segall, *An optimal control approach to dynamic routing in networks*, IEEE Trans. Automat. Control, 27 (1982), pp. 329–339.

[19] R. G. Ogier, *Minimum-delay routing in continuous-time dynamic networks with piecewise-constant capacities*, Networks, 18 (1988), pp. 303–318.

[20] A. B. Philpott, *Algorithms For Continuous Network Flow Problems*, Ph.D. thesis, University of Cambridge, UK, 1982.

[21] A. B. Philpott, *Network programming in continuous time with node storage*, in Infinite Programming: Proceedings of an International Symposium on Infinite Dimensional Linear Programming, E. J. Anderson and A. B. Philpott, eds., Springer-Verlag, Berlin, 1985, pp. 136–153.

[22] A. B. Philpott, *Continuous-time flows in networks*, Math. Oper. Res., 15 (1990), pp. 640–661.

[23] A. B. Philpott and M. Craddock, *An adaptive discretization algorithm for a class of continuous network programs*, Networks, 26 (1995), pp. 1–11.

[24] M. C. Pullan, *An algorithm for a class of continuous linear programs*, SIAM J. Control Optim., 31 (1993), pp. 1558–1577.

[25] M. C. Pullan, *Forms of optimal solutions for separated continuous linear programs*, SIAM J. Control Optim., 33 (1995), pp. 1952–1977.

[26] M. C. Pullan, *Linear optimal control problems with piecewise analytic solutions*, J. Math. Anal. Appl., 197 (1996), pp. 207–226.

[27] M. C. Pullan, *A duality theory for separated continuous linear programs*, SIAM J. Control Optim., 34 (1996), pp. 931–965.

[28] M. C. Pullan, *An extended algorithm for separated continuous linear programs*, Math. Programming, submitted.

[29] M. C. Pullan, *Convergence of a general class of algorithms for separated continuous linear programs*, SIAM J. Optim., submitted.

[30] M. C. Pullan, *A study of special separated continuous linear programs and their use in developing an improved algorithm*, J. Optim. Theory Appl., submitted.

[31] A. Segall, *The modeling of adaptive routing in data-communications networks*, IEEE Trans. Comm., 25 (1977), pp. 85–95.

SIAM J. Optim.
Vol. 7, No. 4, pp. 913–926, November 1997

A NEW CLASS OF INCREMENTAL GRADIENT METHODS FOR LEAST SQUARES PROBLEMS*

DIMITRI P. BERTSEKAS†

Abstract. The least mean squares (LMS) method for linear least squares problems differs from the steepest descent method in that it processes data blocks one-by-one, with intermediate adjustment of the parameter vector under optimization. This mode of operation often leads to faster convergence when far from the eventual limit and to slower (sublinear) convergence when close to the optimal solution. We embed both LMS and steepest descent, as well as other intermediate methods, within a one-parameter class of algorithms, and we propose a hybrid class of methods that combine the faster early convergence rate of LMS with the faster ultimate linear convergence rate of steepest descent. These methods are well suited for neural network training problems with large data sets. Furthermore, these methods allow the effective use of scaling based, for example, on diagonal or other approximations of the Hessian matrix.

Key words. gradient methods, nonlinear programming, least squares, neural networks

AMS subject classifications. 49N10, 65K05, 65K10, 65F20

PII. S1052623495287022

1. Introduction.

We consider least squares problems of the form

$$\text{(1)} \qquad \begin{array}{ll} \text{minimize} & f(x) = \sum_{i=1}^{m} f_i(x) \\ \text{subject to} & x \in \Re^n, \end{array}$$

where $\Re^n$ denotes the n-dimensional Euclidean space and $f_i : \Re^n \to \Re$ are continuously differentiable scalar functions on $\Re^n$. A special case of particular interest to us is the least squares problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} \sum_{i=1}^{m} \|g_i(x)\|^2 \\ \text{subject to} & x \in \Re^n, \end{array}$$

where $g_i : \Re^n \to \Re^{r_i}$, $i = 1, \ldots, m$, are continuously differentiable functions. Here we write $\|z\|$ for the usual Euclidean norm of a vector z; that is, $\|z\| = \sqrt{z'z}$, where prime denotes transposition. We also write ∇f and ∇f_i for the gradients of the functions f and f_i, respectively. Least squares problems often arise in contexts where the functions g_i correspond to data that we are trying to fit with a model parameterized by x. Motivated by this context, we refer to each component f_i as a *data block*, and we refer to the entire collection $(f_1, \ldots, f_m)$ as the *data set*.

In problems where there are many data blocks, and particularly in neural network training problems, gradient-like incremental methods are frequently used. In such methods, one does not wait to process the entire data set before updating x; instead, one cycles through the data blocks in sequence and updates the estimate of x after

[*] Received by the editors June 2, 1995; accepted for publication (in revised form) October 7, 1996. This research was supported by NSF grant 9300494-DMI.

http://www.siam.org/journals/siopt/7-4/28702.html

[†] Deptartment of Electrical Engineering and Computer Science, M.I.T., Cambridge, MA 02139 (dimitrib@mit.edu).

each data block is processed. Such methods include the Widrow–Hoff LMS algorithm [WiH60], [WiS85], for the case of a linear least squares problem, and its extension to nonlinear least squares problems. A cycle through the data set of this method starts with a vector x^k and generates x^{k+1} according to

$$x^{k+1} = \psi_m,$$

where ψ_m is obtained at the last step of the recursion

$$\psi_0 = x^k, \qquad \psi_i = \psi_{i-1} - \alpha^k \nabla f_i(\psi_{i-1}), \quad i = 1, \ldots, m, \tag{2}$$

and α^k is a positive stepsize. Thus the method has the form

$$x^{k+1} = x^k - \alpha^k \sum_{i=1}^{m} \nabla f_i(\psi_{i-1}). \tag{3}$$

We refer to this method, which is just the nonlinear version of the LMS algorithm, as the *incremental gradient method*.

The above method should be contrasted with the steepest descent method, where the data blocks f_i and their gradients are evaluated at the same vector x^k, that is,

$$\psi_0 = x^k, \qquad \psi_i = \psi_{i-1} - \alpha^k \nabla f_i(x^k), \quad i = 1, \ldots, m, \tag{4}$$

so that the iteration consisting of a cycle over the entire data set starting from x^k has the form

$$x^{k+1} = x^k - \alpha^k \sum_{i=1}^{m} \nabla f_i(x^k) = x^k - \alpha^k \nabla f(x^k). \tag{5}$$

Incremental methods are supported by stochastic convergence analyses [PoT73], [Lju77], [KuC78], [TBA86], [Pol87], [BeT89], [Whi89], [Gai94], [BeT96] as well as deterministic convergence analyses [Luo91], [Gri94], [LuT94], [MaS94], [Man93], [Ber95a], [BeT96]. It has been experimentally observed that the incremental gradient method (2)–(3) often converges much faster than the steepest descent method (5) when far from the eventual limit. However, near convergence, the incremental gradient method typically converges slowly because it requires a diminishing stepsize $\alpha^k = O(1/k)$ for convergence. If α^k is instead taken to be a small constant, an oscillation within each data cycle arises, as shown by [Luo91]. By contrast, for convergence of the steepest descent method, it is sufficient that the stepsize α^k is a small constant (this requires that ∇f be Lipschitz continuous; see, e.g., [Pol87]). The asymptotic convergence rate of steepest descent with a constant stepsize is typically linear and much faster than that of the incremental gradient method.

The behavior described above is most vividly illustrated in the case of a linear least squares problem where the vector x is one dimensional, as shown in the following example.

Example 1. Consider the least squares problem

$$\begin{aligned} &\text{minimize} \quad f(x) = \frac{1}{2} \sum_{i=1}^{m} (a_i x - b_i)^2 \\ &\text{subject to} \quad x \in \Re, \end{aligned} \tag{6}$$

where a_i and b_i are given scalars with $a_i \neq 0$ for all i. The minimum of each of the data blocks

$$f_i(x) = \frac{1}{2}(a_i x - b_i)^2 \tag{7}$$

is

$$x_i^* = \frac{b_i}{a_i},$$

while the minimum of the least squares cost function f is

$$x^* = \frac{\sum_{i=1}^m a_i b_i}{\sum_{i=1}^m a_i^2}.$$

It can be seen that x^* lies within the range of the data block minima

$$R = \left[\min_i x_i^*,\, \max_i x_i^*\right] \tag{8}$$

and that for all x *outside* the range R the gradient

$$\nabla f_i(x) = a_i(a_i x - b_i)$$

has the same sign as $\nabla f(x)$. As a result, the incremental gradient method given by

$$\psi_i = \psi_{i-1} - \alpha^k \nabla f_i(\psi_{i-1}) \tag{9}$$

(cf. (2)) approaches x^* at each step provided the stepsize α^k is small enough. In fact it is sufficient that

$$\alpha^k \leq \min_i \frac{1}{a_i^2}. \tag{10}$$

However, for x *inside* the region R, the ith step of a cycle of the incremental gradient method, given by (9), need not make progress because it aims to approach x_i^* but not necessarily x^*. It will approach x^* (for small enough stepsize α^k) only if the current point ψ_{i-1} does not lie in the interval connecting x_i^* and x^*. This induces an oscillatory behavior within the region R, and as a result the incremental gradient method will typically not converge to x^* unless $\alpha^k \to 0$. By contrast, it can be shown that the steepest descent method, which takes the form

$$x^{k+1} = x^k - \alpha^k \sum_{i=1}^m a_i(a_i x^k - b_i),$$

converges to x^* for any constant stepsize satisfying

$$\alpha^k \leq \frac{2}{\sum_{i=1}^m a_i^2}. \tag{11}$$

However, unless the stepsize choice is particularly favorable, for x outside the region R, a full iteration of steepest descent need not make more progress toward the solution than a single step of the incremental gradient method. In other words, *far from the solution (outside R), a single pass through the entire data set by the incremental* *gradient method is roughly as effective as m passes through the data set by the steepest descent method.*

The analysis of the preceding example relies on x being one dimensional, but in many multidimensional problems the same qualitative behavior can be observed. In particular, a pass through the ith data block f_i by the incremental gradient method can make progress toward the solution in the region where the data block gradient $\nabla f_i(\psi_{i-1})$ makes an angle less than 90 degrees with the cost function gradient $\nabla f(\psi_{i-1})$. If the data blocks f_i are not "too dissimilar," this is likely to happen in a region that is not too close to the optimal solution set. For example, consider the case of a linear least squares problem

$$f_i(x) = \frac{1}{2}\|A_i x - b_i\|^2, \tag{12}$$

where the vectors b_i and the matrices A_i are given. Then, it can be shown that sufficiently far from the optimal solution, the direction $\nabla f_i(x)$ used at the ith step of a data cycle of the incremental gradient method will be a descent direction for the entire cost function f if the matrix $A_i'A_i\sum_{j=1}^m A_j'A_j$ is positive definite in the sense that

$$x'A_i'A_i\left(\sum_{j=1}^{m} A_j'A_j\right)x > 0 \qquad \forall\, x \neq 0. \tag{13}$$

This will be true if the matrices A_i are sufficiently close to each other with respect to some matrix norm. One may also similarly argue on a heuristic basis that the incremental gradient method will be substantially more effective than the steepest descent method far from the solution if the above relation holds for a substantial majority of the indices i.

It is also worth mentioning that a similar argument can be made in favor of incremental versions of the Gauss–Newton method for least squares problems. These methods are closely related to the extended Kalman filter algorithm that is used extensively in control and estimation contexts; see, e.g., [Ber95b], [Bel94], [Dav76], [WaT90]. However, like the incremental gradient method, incremental Gauss–Newton methods also suffer from slow ultimate convergence because for convergence they require a diminishing stepsize [Ber95b]. Furthermore, for difficult least squares problems, such as many neural network training problems, it is unclear whether Gauss–Newton methods hold any advantage over gradient methods.

In this paper we introduce a class of gradient-like methods parameterized by a single nonnegative constant μ. For the two extreme values $\mu = 0$ and $\mu = \infty$, we obtain as special cases the incremental gradient and steepest descent methods, respectively. Positive values of μ yield hybrid methods with varying degrees of incrementalism in processing the data blocks. We also propose a time-varying hybrid method, where μ is gradually increased from $\mu = 0$ toward $\mu = \infty$. This method aims to combine the typically faster initial convergence rate of incremental gradient with the faster ultimate convergence rate of steepest descent. It starts out as the incremental gradient method (2)–(3), but gradually (based on algorithmic progress) it becomes less and less incremental, and asymptotically it approaches the steepest descent method (5). In contrast to the incremental gradient method, it uses a constant stepsize without resulting in an asymptotic oscillation. We prove convergence and a linear rate of convergence for this method in the case where the data blocks are positive semidefinite quadratic functions. Similar results can be shown for the case of nonquadratic data blocks and a parallel asynchronous computing environment.

In addition to a linear convergence rate, the use of a constant stepsize offers another important practical advantage: it allows a more effective use of scaling based, for example, on approximations of the Hessian matrix. Our experience shows that our method performs better than both the incremental gradient and the steepest descent method, particularly when scaling is used.

2. The new incremental gradient method.

We embed the incremental gradient method (2)–(3) and the steepest descent method (5) within a one-parameter family of methods for the least squares problem. Let us fix a scalar $\mu \geq 0$. Consider the method which given x^k generates x^{k+1} according to

$$x^{k+1} = \psi_m, \tag{14}$$

where ψ_m is generated at the last step of the algorithm

$$\psi_i = x^k - \alpha^k h_i, \qquad i = 1, \ldots, m, \tag{15}$$

and the vectors h_i are defined as follows:

$$h_i = \sum_{j=1}^{i} w_{ij}(\mu) \nabla f_j(\psi_{j-1}), \qquad i = 1, \ldots, m, \tag{16}$$

where

$$\psi_0 = x^k, \tag{17}$$

and

$$w_{ij}(\mu) = \frac{1 + \mu + \cdots + \mu^{i-j}}{1 + \mu + \cdots + \mu^{m-j}}, \qquad i = 1, \ldots, m,\ 1 \leq j \leq i. \tag{18}$$

It can be verified using induction that the vectors h_i can be generated recursively using the formulas

$$h_i = \mu h_{i-1} + \sum_{j=1}^{i} \xi_j(\mu) \nabla f_j(\psi_{j-1}), \qquad i = 1, \ldots, m, \tag{19}$$

where $h_0 = 0$ and

$$\xi_i(\mu) = \frac{1}{1 + \mu + \cdots + \mu^{m-i}}, \qquad i = 1, \ldots, m. \tag{20}$$

Thus the computation of h_i using (19) requires (essentially) no more storage or overhead per iteration than either the steepest descent method (5) or the incremental gradient method (2)–(3).

Note that since

$$w_{mj}(\mu) = 1, \qquad j = 1, \ldots, m,$$

it follows using (15)–(16) that the vector ψ_m obtained at the end of a pass through all the data blocks is

$$\psi_m = x^{k+1} = x^k - \alpha^k h_m = x^k - \alpha^k \sum_{j=1}^{m} \nabla f_j(\psi_{j-1}). \tag{21}$$

In the special case where $\mu = 0$, we have $w_{ij}(\mu) = 1$ for all i and j, and by comparing (15), (18), (2), and (3) we see that the method coincides with the incremental gradient method (2)–(3). In the case where $\mu \to \infty$, we have from (15), (18), and (19) $w_{ij}(\mu) \to 0$, $h_i \to 0$, and $\psi_i \to x^k$ for $i = 0, 1, \ldots, m-1$, so by comparing (21) and (5) we see that the method approaches the steepest descent method (5). Generally, it can be seen that as μ increases the method becomes "less incremental."

We first prove a convergence result for the method (13)–(17) for the case where μ is fixed and each data block f_i is positive semidefinite quadratic. This covers the case of a linear least squares problem. In particular, we show that if the stepsize α^k is a sufficiently small constant, the algorithm asymptotically oscillates around the optimal solution. However, the "size" of the oscillation diminishes as either $\alpha \to 0$ and μ is constant or as α is constant and $\mu \to \infty$. If the stepsize is diminishing of the form $\alpha^k = O(1/k)$, the method converges to the minimum for all values of μ.

PROPOSITION 2.1. *Suppose that the functions f_i have the form*

$$f_i(x) = \frac{1}{2}x'Q_i x - c_i'x, \qquad i = 1, \ldots, m,$$

where Q_i are given positive semidefinite symmetric matrices and c_i are given vectors. Consider the algorithm (*cf.* (13)–(17))

$$x^{k+1} = \psi_m, \tag{22}$$

where

$$\psi_0 = x^k, \qquad \psi_i = x^k - \alpha^k h_i, \quad i = 1, \ldots, m, \tag{23}$$

$$h_0 = 0, \qquad h_i = \mu h_{i-1} + \sum_{j=1}^{i} \xi_j(\mu)(Q_j\psi_{j-1} - c_j), \quad i = 1, \ldots, m. \tag{24}$$

Assume that $\sum_{j=1}^m Q_j$ is a positive definite matrix, and let x^* be the optimal solution of (1). *Then the following hold:*

(a) *For each $\mu \geq 0$, there exists $\overline{\alpha}(\mu) > 0$ such that if α^k is equal to some constant $\alpha \in (0, \overline{\alpha}(\mu)]$ for all k, $\{x^k\}$ converges to some vector $x(\alpha, \mu)$, and we have $\lim_{\alpha \to 0^+} x(\alpha, \mu) = x^*$. Furthermore, there exists $\overline{\alpha} > 0$ such that $\overline{\alpha} \leq \overline{\alpha}(\mu)$ for all $\mu \geq 0$, and for all $\alpha \in (0, \overline{\alpha}]$ we have $\lim_{\mu \to \infty} x(\alpha, \mu) = x^*$.*

(b) *For each $\mu \geq 0$, if $\alpha^k > 0$ for all k and*

$$\alpha^k \to 0, \qquad \sum_{k=0}^{\infty} \alpha^k = \infty, \tag{25}$$

then $\{x^k\}$ converges to x^*.

Proof. (a) We first note that from (21) we have

$$x^{k+1} = x^k - \alpha \sum_{j=1}^{m} (Q_j \psi_{j-1} - c_j),$$

so by using the definition $\psi_{j-1} = x^k - \alpha h_{j-1}$ we obtain

$$x^{k+1} = x^k - \alpha \sum_{j=1}^{m} (Q_j x^k - c_j) + \alpha^2 \sum_{j=1}^{m} Q_j h_{j-1}. \tag{26}$$

We next observe that from (18) and the definition $\psi_{j-1} = x^k - \alpha h_{j-1}$ we have for all i

$$h_i = \sum_{j=1}^{i} w_{ij}(\mu)(Q_j\psi_{j-1} - c_j) \tag{27}$$

$$= \sum_{j=1}^{i} w_{ij}(\mu)Q_j x^k - \alpha \sum_{j=1}^{i} w_{ij}(\mu)Q_j h_{j-1} - \sum_{j=1}^{i} w_{ij}(\mu)c_j.$$

From this relation it can be seen inductively that for all i, h_i can be written as

$$h_i = \sum_{j=1}^{i} w_{ij}(\mu)Q_j x^k - \sum_{j=1}^{i} w_{ij}(\mu)c_j + \alpha R_i(\alpha,\mu)x^k + \alpha r_i(\alpha,\mu), \tag{28}$$

where $R_i(\alpha,\mu)$ and $r_i(\alpha,\mu)$ are some matrices and vectors, respectively, depending on α and μ. Furthermore, using (27) and the fact that $w_{ij}(\mu) \in (0,1]$ for all i, j, and $\mu \geq 0$, we have that for any bounded interval T of stepsizes α there exist positive uniform bounds $\overline{R}$ and $\overline{r}$ for $\|R_i(\alpha,\mu)\|$ and $\|r_i(\alpha,\mu)\|$; that is,

$$\|R_i(\alpha,\mu)\| \leq \overline{R}, \qquad \|r_i(\alpha,\mu)\| \leq \overline{r} \qquad \forall\ i,\ \mu \geq 0,\ \alpha \in T. \tag{29}$$

From (26), (28), and (29) we obtain

$$x^{k+1} = A(\alpha,\mu)x^k + b(\alpha,\mu), \tag{30}$$

where

$$A(\alpha,\mu) = I - \alpha \sum_{j=1}^{m} Q_j + \alpha^2 S(\alpha,\mu), \tag{31}$$

$$b(\alpha,\mu) = \alpha \sum_{j=1}^{m} c_j + \alpha^2 s(\alpha,\mu), \tag{32}$$

I is the identity matrix, and the matrix $S(\alpha,\mu)$ and the vector $s(\alpha,\mu)$ are uniformly bounded over $\mu \geq 0$ and any bounded interval T of stepsizes α; that is, for some scalars $\overline{S}$ and $\overline{s}$,

$$\|S(\alpha,\mu)\| \leq \overline{S}, \qquad \|s(\alpha,\mu)\| \leq \overline{s} \qquad \forall\ \mu \geq 0,\ \alpha \in T. \tag{33}$$

Let us choose the interval T to contain small enough stepsizes so that for all $\mu \geq 0$ and $\alpha \in T$, the eigenvalues of $A(\alpha,\mu)$ are all strictly within the unit circle; this is possible since $\sum_{j=1}^{m} Q_j$ is assumed positive definite and (31) and (33) hold. Define

$$x(\alpha,\mu) = \big(I - A(\alpha,\mu)\big)^{-1} b(\alpha,\mu). \tag{34}$$

Then $b(\alpha,\mu) = \big(I - A(\alpha,\mu)\big)x(\alpha,\mu)$, and by substituting this expression in (30) it can be seen that

$$x^{k+1} - x(\alpha,\mu) = A(\alpha,\mu)\big(x^k - x(\alpha,\mu)\big),$$

from which

$$x^{k+1} - x(\alpha,\mu) = A(\alpha,\mu)^k \big(x^0 - x(\alpha,\mu)\big) \qquad \forall\ k.$$

Since all the eigenvalues of $A(\alpha,\mu)$ are strictly within the unit circle, we have $A(\alpha,\mu)^k \to 0$, so $x^k \to x(\alpha,\mu)$.

To prove that $\lim_{\alpha\to 0} x(\alpha,\mu) = x^*$, we first calculate x^*. We set the gradient of f to 0 to obtain

$$\sum_{j=1}^{m} (Q_j x^* - c_j) = 0,$$

so that

$$x^* = \left(\sum_{j=1}^{m} Q_j\right)^{-1} \sum_{i=1}^{m} c_j. \tag{35}$$

Then we use (34) to write $x(\alpha,\mu) = \big(I/\alpha - A(\alpha,\mu)/\alpha\big)^{-1}\big(b(\alpha,\mu)/\alpha\big)$, and we see from (31) and (32) that

$$\lim_{\alpha\to 0} x(\alpha,\mu) = \left(\sum_{j=1}^{m} Q_j\right)^{-1} \sum_{i=1}^{m} c_j = x^*.$$

To prove that $\lim_{\mu\to\infty} x(\alpha,\mu) = x^*$, we note that since $\lim_{\mu\to\infty} w_{ij}(\mu) = 0$ for $i = 1,\ldots,m-1$, it follows from (16) that h_{j-1} tends to 0 as $\mu \to \infty$ for $j = 1,\ldots,m-1$. Using this fact in conjunction with (26) and (30)–(32) it follows that

$$\lim_{\mu\to\infty} S(\alpha,\mu) = 0, \qquad \lim_{\mu\to\infty} s(\alpha,\mu) = 0.$$

From (31), (32), and (34) we then obtain

$$\lim_{\mu\to\infty} x(\alpha,\mu) = \left(\alpha \sum_{j=1}^{m} Q_j\right)^{-1} \left(\alpha \sum_{j=1}^{m} c_j\right) = x^*.$$

(b) We need the following well-known lemma (for a proof, see [Luo91], [Ber95a], [BeT96]).

Lemma 2.1. *Suppose that $\{e^k\}$ and $\{\gamma^k\}$ are nonnegative sequences and c is a positive constant such that*

$$e^{k+1} \le (1-\gamma^k)e^k + c(\gamma^k)^2, \quad \gamma^k \le 1, \qquad k = 0, 1 \ldots,$$

and

$$\gamma^k \to 0, \qquad \sum_{k=0}^{\infty} \gamma^k = \infty.$$

Then $e^k \to 0$.

Returning to the proof of Proposition 2.1, from (21) and (30)–(32) we have

$$x^{k+1} = x^k - \alpha^k \sum_{j=1}^{m} (Q_j x^k - c_j) + (\alpha^k)^2 S(\alpha^k, \mu)(x^k - x^*) + (\alpha^k)^2 e^k, \tag{36}$$

where

$$e^k = S(\alpha^k, \mu) x^* + s(\alpha^k, \mu). \tag{37}$$

Using also the expression (35) for x^*, we can write (36) as

$$x^{k+1} - x^* = \left(I - \alpha^k \sum_{j=1}^{m} Q_j + (\alpha^k)^2 S(\alpha^k, \mu) \right) (x^k - x^*) + (\alpha^k)^2 e^k. \tag{38}$$

For large enough k, the eigenvalues of $\alpha^k \sum_{j=1}^m Q_j$ are bounded from above by 1, and hence the matrix $I - \alpha^k \sum_{j=1}^m Q_j$ is positive definite. Without loss of generality, we assume that this is so for all k. Then we have

$$\left\| \left(I - \alpha^k \sum_{j=1}^{m} Q_j \right) (x^k - x^*) \right\| \le (1 - \alpha^k A) \|x^k - x^*\|, \tag{39}$$

where A is the smallest eigenvalue of $\sum_{j=1}^m Q_j$. Let also B and δ be positive scalars such that for all k we have

$$\big\| S(\alpha^k, \mu)(x^k - x^*) \big\| \le B \|x^k - x^*\|, \qquad \|e^k\| \le \delta. \tag{40}$$

Combining (38)–(40), we have

$$\begin{aligned} \|x^{k+1} - x^*\| &\le \left\| \left(I - \alpha^k \sum_{j=1}^{m} Q_j \right) (x^k - x^*) \right\| + (\alpha^k)^2 \big\| S(\alpha^k, \mu)(x^k - x^*) \big\| + (\alpha^k)^2 \|e^k\| \\ &\le (1 - \alpha^k A + (\alpha^k)^2 B) \|x^k - x^*\| + (\alpha^k)^2 \delta. \end{aligned} \tag{41}$$

Let $\overline{k}$ be such that $\alpha^k B \le A/2$ for all $k \ge \overline{k}$. Then from (41) we obtain

$$\|x^{k+1} - x^*\| \le (1 - \alpha^k A/2) \|x^k - x^*\| + (\alpha^k)^2 \delta \qquad \forall\ k \ge \overline{k},$$

and Lemma 2.1 can be used to show that $\|x^k - x^*\| \to 0$. □

The following proposition shows that if μ is increased toward ∞ at a sufficiently fast rate, the sequence $\{x^k\}$ generated by the method with a constant stepsize converges at a linear rate.

PROPOSITION 2.2. *Suppose that in the kth iteration of the method* (14)–(18)*, a k-dependent value of μ, say $\mu(k)$, and a constant stepsize $\alpha^k = \alpha$ are used. Under the assumptions of Proposition* 2.1*, if for some $q > 1$ and all k greater than some index $\overline{k}$, we have $\mu(k) \ge q^k$, then there exists $\overline{\alpha} > 0$ such that for all $\alpha \in (0, \overline{\alpha}]$ and k we have $\|x^k - x^*\| \le p(\alpha)\beta(\alpha)^k$, where $p(\alpha) > 0$ and $\beta(\alpha) \in (0, 1)$ are some scalars depending on α.*

Proof. We first note that the proof of Proposition 2.1(a) can be modified to show that there exists $\overline{\alpha} > 0$ such that for all $\alpha \in (0, \overline{\alpha}]$ we have $x^k \to x^*$. We also note that if for some $q > 1$, we have $\mu(k) \geq q^k$ for k after some index $\overline{k}$, then for all $i < m$ and $j \leq i$ we have

$$w_{ij}\big(\mu(k)\big) = O(\gamma^k), \tag{42}$$

where γ is some scalar with $\gamma \in (0, 1)$.

We next observe that similar to the derivation of (38) we have

$$x^{k+1} - x^* = \left(I - \alpha \sum_{j=1}^{m} Q_j + \alpha^2 S\big(\alpha, \mu(k)\big) \right) (x^k - x^*) + \alpha^2 e^k, \tag{43}$$

where

$$e^k = S\big(\alpha, \mu(k)\big) x^* + s\big(\alpha, \mu(k)\big). \tag{44}$$

From (27), we see that h_i can be written as a finite number of terms of bounded norm, which are multiplied by some term $w_{ij}(\mu(k))$. Thus, in view of (42), for $i < m$ we have $\|h_i\| = O(\gamma^k)$, which by comparing (27) and (28) implies that for all i

$$\|R_i\big(\alpha, \mu(k)\big)\| = O(\gamma^k), \qquad \|r_i\big(\alpha, \mu(k)\big)\| = O(\gamma^k).$$

It follows that

$$\|S\big(\alpha, \mu(k)\big)\| = O(\gamma^k), \qquad \|s\big(\alpha, \mu(k)\big)\| = O(\gamma^k). \tag{45}$$

From (44) we then obtain

$$\|e^k\| = O(\gamma^k). \tag{46}$$

From (43), (45), and (46), we obtain

$$\|x^{k+1} - x^*\| \leq \big(|1 - \alpha\delta| + O(\gamma^k)\big) \|x^k - x^*\| + \alpha^2 O(\gamma^k),$$

where δ is the minimum eigenvalue of $\sum_{j=1}^{m} Q_j$. This relation implies the desired rate of convergence result. □

There are a number of fairly straightforward extensions of the methods and the results just presented.

(1) When the data blocks are nonquadratic, stationarity of the limit points of sequences $\{x^k\}$ generated by the method (13)–(17) can be shown under certain assumptions (including Lipschitz continuity of the data block gradients) for the case of a fixed μ and the stepsize $\alpha^k = \gamma/(k + \delta)$, where γ and δ are positive scalars. Contrary to the case of quadratic data blocks, γ may have to be chosen sufficiently small to guarantee boundedness of $\{x^k\}$. The convergence proof is similar to the one of the preceding proposition, but it is technically more involved. In the case where the stepsize is constant, $\mu \to \infty$, and the data blocks are nonquadratic, it is also possible to show a result analogous to Proposition 2.2, but again the proof is technically complex and will not be given.

(2) Convergence results for parallel asynchronous versions of our method can be given, in the spirit of those in [TBA86], [BeT89, Chap. 7], and [MaS94]. These results follow well-established methods of analysis that rely on the stepsize being sufficiently small.

(3) Variations of our method involving a quadratic momentum term are possible. The use of such terms dates to the heavy ball method of Poljak (see [Pol64], [Pol87], [Ber95a]) in connection with the steepest descent method and has become popular in the context of the incremental gradient method, particularly for neural network training problems (see [MaS94] for an analysis).

(4) Diagonal scaling of the iterations generating ψ_i is possible by replacing the equation $\psi_i = x^k - \alpha^k h_i$ (cf. (15)) with the equation

$$\psi_i = x^k - \alpha^k D h_i, \qquad i = 1, \dots, m,$$

where D is a positive-definite symmetric matrix. A common approach is to use a diagonal matrix D whose diagonal elements are the inverses of the corresponding diagonal elements of the Hessian of the cost function

$$\sum_{j=1}^{m} \nabla^2 f_j(\psi_{j-1}).$$

An important advantage of this type of diagonal scaling is that it simplifies the choice of a constant stepsize; a value of stepsize equal to 1 or a little smaller typically works well. Diagonal scaling is often beneficial for steepest descent-like methods that use a constant stepsize but is not as helpful for the incremental gradient method because the latter uses a variable (diminishing) stepsize. For this reason diagonal scaling should be typically more effective for the constant stepsize methods proposed here than for the incremental gradient method. This was confirmed in our computational experiments; see also the discussion of the next section. For this reason, we believe that for problems where diagonal scaling is important for good performance our constant stepsize methods have a significant advantage over the LMS and the incremental gradient methods.

3. Implementation and experimentation. Let us consider algorithms where μ is iteration dependent and is increased with k toward ∞. While Proposition 2.2 suggests that a linear convergence rate can be obtained by keeping α constant, we have found in our experimentation that it may be important to change α simultaneously with μ when μ is still relatively small. In particular, as the problem of Example 1 suggests, when μ is near 0 and the method is similar to the incremental gradient method, the stepsize should be larger, while when μ is large, the stepsize should be of comparable magnitude to the corresponding stepsize of steepest descent.

The formula for $\xi_i(\mu)$ suggests that for $\mu \le 1$ the incremental character of the method is strong, so we have experimented with a μ-dependent stepsize formula of the form

$$\alpha(\mu) = \begin{cases} \gamma & \text{if } \mu > 1, \\ \big(1 + \phi(\mu)\big)\gamma & \text{if } \mu \in [0, 1]. \end{cases} \tag{47}$$

Here γ is the stepsize that works well with the steepest descent method and should be determined to some extent by trial and error (if diagonal scaling is used, then a choice of γ close to 1 often works well). The function $\phi(\mu)$ is a monotonically decreasing function with

$$\phi(0) = \zeta, \qquad \phi(1) = 0, \tag{48}$$

where ζ is a scalar in the range $[0, m-1]$. Examples are

$$\phi(\mu) = \zeta(1-\mu), \qquad \phi(\mu) = \zeta(1-\mu^2), \qquad \phi(\mu) = \zeta(1-\sqrt{\mu}). \tag{49}$$

In some of the variations of the method that we experimented with, the scalar ζ was decreased by a certain factor each time μ was increased. Generally, with μ-dependent stepsize selection of the form (49) and diagonal scaling, we have found the constant stepsize methods proposed here far more effective than the incremental gradient method that uses the same diagonal scaling and a diminishing stepsize.

Regarding the rule for increasing μ, we have experimented with schemes that start with $\mu = 0$ and update μ according to a formula of the form

$$\mu := \beta\mu + \delta,$$

where β and δ are fixed positive scalars with $\beta > 1$. The update of μ takes place at the start of a data cycle following the computation of x^{k+1} if either

$$\|x^{k+1} - x^k\| \le \epsilon, \tag{50}$$

where ϵ is a fixed tolerance, or $\hat{n}$ data cycles have been performed since the last update of μ, where $\hat{n}$ is an integer chosen by trial and error. This criterion tries to update μ when the method appears to be making little further progress at the current level of μ but also updates μ after a maximum specified number $\hat{n}$ of data cycles have been performed with the current μ.

We noted one difficulty with the method. When the number of data blocks m is large, the calculation of $\xi_i(\mu)$ using (20) involves high powers of μ. This tends to introduce substantial numerical error when μ is substantially larger than 1. To get around this difficulty, we modified the method by lumping together an increasing number of data blocks (the minimum number of terms in a data block was incremented by 1) each time μ was increased to a value above 1. This device effectively reduces the number of data blocks m and keeps the power μ^m bounded. In our computational experiments, it has eliminated the difficulty with numerical errors without substantially affecting the performance of the method.

Finally, let us try to compare the diagonally scaled version of our method with the diagonally scaled incremental gradient method given by

$$x^{k+1} = x^k - \alpha^k D \sum_{j=1}^{m} \nabla f_j(\psi_{j-1}), \tag{51}$$

where ψ_i is generated by

$$\psi_i = x^k - \alpha^k D \sum_{j=1}^{i} \nabla f_j(\psi_{j-1}). \tag{52}$$

We assume that D is a diagonal approximation of the inverse Hessian of f. It is difficult to draw definitive conclusions regarding the two methods because their performance depends a lot on various tuning parameters. In particular, it is very difficult to compare the methods using computational results with only a few test problems, and this will not be attempted. On the other hand, it is helpful to consider some extreme problem cases.

(1) Problems where diagonal scaling is effective because the Hessian matrix of f is nearly diagonal. For such problems, both methods can be very fast with proper tuning of the stepsize parameters. On the other hand the incremental gradient method after a few iterations slows down because of the diminishing stepsize. By contrast, our method maintains its rate of convergence, and, indeed, once μ reaches high values and when $\alpha^k \approx 1$, it may become even faster than in the early iterations where μ is small, because for large μ it effectively approximates Newton's method.

(2) Problems where diagonal scaling is ineffective because the Hessian matrix of f is not nearly diagonal and is ill conditioned. Then both methods will likely be slow regardless of how they are tuned. On the other hand the convergence rate of the incremental gradient method will continually deteriorate because of the diminishing stepsize, while our method will at least maintain a (slow) linear convergence rate.

(3) Problems that do not fall in the preceding categories but which have "homogeneous" data blocks, that is, problems where the Hessian matrices $\nabla^2 f_i$ of the data blocks are not too dissimilar. Then incrementalism is likely to be very beneficial (think of the extreme case where all the data blocks are identical). For such problems the incremental gradient method may have an edge in the early iterations because of its greater degree of incrementalism, although asymptotically our method maintains the advantage of the linear convergence rate.

(4) Problems that do not fall in the preceding categories, but which have "inhomogeneous" data blocks, where the Hessian matrices $\nabla^2 f_i$ of the data blocks are quite dissimilar. Then our method is likely to have an advantage over the incremental gradient method, because it gradually becomes nonincremental, while maintaining a nondiminishing stepsize and the attendant linear convergence rate.

The preceding arguments, while speculative, are consistent with the results of the author's experimentation. However, a far more comprehensive experimentation as well as experience with real-world problems is needed to support the preceding conclusions and to assess more reliably the merits of the method proposed.

REFERENCES

[BeT89] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice–Hall, Englewood Cliffs, NJ, 1989.

[BeT96] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, Belmont, MA, 1996.

[Bel94] B. M. Bell, *The iterated Kalman smoother as a Gauss–Newton method*, SIAM J. Optim., 4 (1994), pp. 626–636.

[Ber95a] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995.

[Ber95b] D. P. Bertsekas, *Incremental least squares methods and the extended Kalman filter*, SIAM J. Optim., 6 (1996), pp. 807–822.

[Dav76] W. C. Davidon, *New least squares algorithms*, J. Optim. Theory Appl., 18 (1976), pp. 187–197.

[Gai94] A. A. Gaivoronski, *Convergence analysis of parallel backpropagation algorithm for neural networks*, Optimization Methods and Software, 4 (1994), pp. 117–134.

[Gri94] L. Grippo, *A class of unconstrained minimization methods for neural network training*, Optimization Methods and Software, 4 (1994), pp. 135–150.

[KuC78] H. J. Kushner and D. S. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, Springer-Verlag, New York, 1978.

[Lju77] L. Ljung, *Analysis of recursive stochastic algorithms*, IEEE Trans. Automat. Control, 22 (1977), pp. 551–575.

[LuT94] Z. Q. Luo and P. Tseng, *Analysis of an approximate gradient projection method with applications to the backpropagation algorithm*, Optimization Methods and Software, 4 (1994), pp. 85–101.

[Luo91] Z. Q. Luo, *On the convergence of the LMS algorithm with adaptive learning rate for linear feedforward networks*, Neural Computation, 3 (1991), pp. 226–245.

[MaS94] O. L. Mangasarian and M. V. Solodov, *Serial and parallel backpropagation convergence via nonmonotone perturbed minimization*, Optimization Methods and Software, 4 (1994), pp. 103–116.

[Man93] O. L. Mangasarian, *Mathematical programming in neural networks*, ORSA J. Comput., 5 (1993), pp. 349–360.

[PoT73] B. T. Poljak and Y. Z. Tsypkin, *Pseudogradient adaptation and training algorithms*, Automat. Remote Control, 12 (1973), pp. 83–94.

[Pol87] B. T. Poljak, *Introduction to Optimization*, Optimization Software Inc., New York, 1987.

[TBA86] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, *Distributed asynchronous deterministic and stochastic gradient optimization algorithms*, IEEE Trans. Automat. Control, AC-31, (1986), pp. 803–812.

[WaT90] K. Watanabe and S. G. Tzafestas, *Learning algorithms for neural networks with the Kalman filters*, J. Intelligent and Robotic Systems, 3 (1990), pp. 305–319.

[Whi89] H. White, *Some asymptotic results for learning in single hidden-layer feedforward network models*, J. Amer. Statist. Assoc., 84 (1989), pp. 1003–1013.

[WiH60] B. Widrow and M. E. Hoff, *Adaptive Switching Circuits*, Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record, part 4, 1960, pp. 96–104.

[WiS85] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice–Hall, Englewood Cliffs, NJ, 1985.

SIAM J. Optim.
Vol. 7, No. 4, pp. 927–950, November 1997

ON THE CONVERGENCE THEORY OF TRUST-REGION-BASED ALGORITHMS FOR EQUALITY-CONSTRAINED OPTIMIZATION[*]

J. E. DENNIS[†] AND LUÍS N. VICENTE[‡]

Abstract. In a recent paper, Dennis, El-Alem, and Maciel proved global convergence to a stationary point for a general trust-region-based algorithm for equality-constrained optimization. This general algorithm is based on appropriate choices of trust-region subproblems and seems particularly suitable for large problems.

This paper shows global convergence to a point satisfying the second-order necessary optimality conditions for the same general trust-region-based algorithm. The results given here can be seen as a generalization of the convergence results for trust-regions methods for unconstrained optimization obtained by Moré and Sorensen. The behavior of the trust radius and the local rate of convergence are analyzed. Some interesting facts concerning the trust-region subproblem for the linearized constraints, the quasi-normal component of the step, and the hard case are presented.

It is shown how these results can be applied to a class of discretized optimal control problems.

Key words. equality-constrained optimization, trust regions, SQP methods, second-order necessary optimality conditions, local rate of convergence, hard case

AMS subject classifications. 49M37, 90C30

PII. S1052623494276026

1. Introduction.

Trust-region algorithms have been proved to be efficient and robust techniques to solve unconstrained optimization problems. An excellent survey in this area is Moré [22]. Other classical references for convergence results are Carter [3], Moré and Sorensen [23], Powell [26], and Shultz, Schnabel, and Byrd [29]. The standard techniques to handle the trust-region subproblems are the dogleg algorithm (Powell [25]), conjugate gradients (Steihaug [32] and Toint [33]), and Newton-like methods for the computation of locally constrained optimal steps (Gay [15], Moré and Sorensen [23], and Sorensen [30]). See also the book of Dennis and Schnabel [9]. Recent new algorithms to compute a locally constrained optimal step (in other words a step that satisfies a fraction of optimal decrease on the trust-region subproblem) that are very promising for large problems have been proposed by Rendl and Wolkowicz [28] and Sorensen [31].

Since the mid eighties a significant effort has been made to address the equality-constrained optimization problem. References are Celis, Dennis, and Tapia [4], Vardi [34] (see also El-Hallabi [14]), Byrd, Schnabel, and Shultz [2], Powell and Yuan [27], and El-Alem [13]. The fundamental questions associated with the application of trust-region algorithms to equality-constrained optimization are the decomposition of the step, the choice of the trust-region subproblems, and the choice of the merit function. During the first stages of the research conducted in this area it was not clear how to

[*] Received by the editors October 24, 1994; accepted for publication (in revised form) July 16, 1996.

http://www.siam.org/journals/siopt/7-4/27602.html

[†] Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005–1892 (dennis@rice.edu). Support of this author has been provided by DOE contract DOE–FG03–93ER25178, NSF cooperative agreement CCR–9120008, and AFOSR contract F49620–9310212.

[‡] Departamento de Matemática, Universidade de Coimbra, 3000 Coimbra, Portugal (lvicente@mat.uc.pt). This work was developed while the author was a graduate student at the Department of Computational and Applied Mathematics of Rice University. Support of this author has been provided by INVOTAN (NATO scholarship), CCLA (Fulbright scholarship), FLAD, and NSF cooperative agreement CCR–9120008.

answer these questions properly. However, if we examine carefully the most recent references (Byrd and Omojokon [24], Dennis, El-Alem, and Maciel [7], El-Alem [12], [13], and Lalee, Nocedal, and Plantenga [21]) we can observe the same decomposition of the step (in its normal, or quasi-normal, and tangential components) and the same trust-region subproblems (the trust-region subproblem for the linearized constraints and the trust-region subproblem for the Lagrangian reduced to the tangent subspace). This is explained in great detail in section 2 of this paper. As in the unconstrained case, the conditions that each component has to satisfy and the way they are computed might of course differ from algorithm to algorithm. We can see also in these most recent references that the merit function used is either the ℓ_2 penalty function without constraint term squared (cases of [21], [24]) or the augmented Lagrangian function (in [7], [12], [13]).

Consider now the equality-constrained optimization (ECO) problem

$$\begin{array}{rl} \text{minimize} & f(x) \\ \text{subject to} & C(x) = 0, \end{array} \tag{1.1}$$

where $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, $c_i : \mathbb{R}^n \longrightarrow \mathbb{R}$, $i = 1, \ldots, m$, $C(x) = (c_1(x) \cdots c_m(x))^T$, and $m < n$. The functions f and c_i, $i = 1, \ldots, m$, are assumed to be at least twice continuously differentiable in the domain of interest.

In [7], Dennis, El-Alem, and Maciel have considered a general trust-region-based algorithm for the solution of the ECO problem (1.1). This general algorithm is very much like the algorithm proposed by Byrd and Omojokon [24].[1] As mentioned before, each step s is decomposed as $s^{\mathsf{n}} + s^{\mathsf{t}}$, where s^{n} is the quasi-normal component of the step, associated with the trust-region subproblem for the linearized constraints and s^{t} is the tangential component, associated with the trust-region subproblem for the Lagrangian reduced to the tangent subspace. Each component of the step is required to satisfy only a fraction of Cauchy decrease on the corresponding trust-region subproblem. Another key feature of this general algorithm is the choice of the augmented Lagrangian as a merit function and the use of the El-Alem's scheme [11] to update the penalty parameter. Under appropriate assumptions, it can be shown that there exists a subsequence of iterates driving to zero the norm of the residual of the constraints and the norm of the gradient of the Lagrangian reduced to the tangent subspace (see [7, section 8]). It is important to remark that this global convergence result is obtained under very mild conditions on the components of the step, on the multipliers estimates, and on the Hessian approximations. Thus, the Dennis, El-Alem, and Maciel [7] result is similar to the global result given by Powell [26] for unconstrained optimization.

One of the purposes of this paper is to show global convergence to a point satisfying the second-order necessary optimality conditions for this class of algorithms. Our result is similar to the results established by Moré and Sorensen [23], [30] for trust-region algorithms for unconstrained optimization. We accomplish this here by imposing a fraction of optimal decrease on the tangential component s^{t} of the step, by using exact second-order information, and by imposing conditions on the quasi-normal component s^{n} and on the Lagrange multipliers.

In [2], Byrd, Schnabel, and Shultz have proposed a trust-region algorithm for equality-constrained optimization and established global convergence to a point satisfying the second-order necessary optimality conditions. However, this algorithm does

[1] The thesis [24] was directed by Professor R. H. Byrd. The trust-region algorithm proposed here is usually referred as the Byrd and Omojokon algorithm.

not belong to the class of trust-region algorithms considered here, and their result is obtained with the use of the (exact) normal component and the least-squares multipliers update which we do not require in this paper. Other differences are that they use the ℓ_1 penalty function as the merit function and the analysis is carried out by using an orthogonal null-space basis. In recent papers, Coleman and Yuan [6] and El-Alem [12] have proposed trust-region algorithms for which they prove global convergence to points satisfying first-order and second-order necessary optimality conditions. Their algorithms use the (exact) normal component, an orthogonal null-space basis, and the least-squares multipliers update.

The conditions we need to impose to assure that a limit point of the sequence of iterates satisfies the second-order necessary optimality conditions are

$$\nabla_x \ell(x_k, \lambda_k)^T s_k^{\mathsf{n}} = \mathcal{O}(\delta_k \|C(x_k)\|) \text{ and } \|\Delta\lambda_k\| = \|\lambda_{k+1} - \lambda_k\| = \mathcal{O}(\delta_k),$$

where $\ell(x, \lambda) = f(x) + \lambda^T C(x)$, s_k^{n} is the quasi-normal component of the step s_k, and δ_k is the trust-region radius. In the case where $\|C(x_k)\|$ is small compared with δ_k, the first condition implies that any increase of the quadratic model of the Lagrangian from x_k to $x_k + s_k^{\mathsf{n}}$ is $\mathcal{O}(\delta_k^2)$. To see why this is relevant recall that a fraction of optimal decrease is being imposed on the tangential component s_k^{t} yielding a decrease of $\mathcal{O}(\delta_k^2)$ on the quadratic model. The second condition is needed for the same reasons because $\Delta\lambda_k$ also appears in the definition of predicted decrease. We show that both conditions are satisfied when either (i) the (exact) normal component and the least-squares multipliers are used; or (ii) the most reasonable choices of quasi-normal component and multipliers are made for a class of discretized optimal control problems. The former result is in agreement with the result obtained by El-Alem [12].

Gill, Murray, and Wright [17] and El-Alem [10] considered in their analyses that $\nabla_x \ell(x_k, \lambda_k)$ is $\mathcal{O}(\|s_k\|)$. In the latter work this assumption is used to prove local convergence results, and in the former to establish properties of an augmented Lagrangian merit function. We point out that this assumption implies that $\nabla_x \ell(x_k, \lambda_k)^T s_k^{\mathsf{n}}$ is $\mathcal{O}(\delta_k \|C(x_k)\|)$ since s_k is $\mathcal{O}(\delta_k)$ and we assume that s_k^{n} is $\mathcal{O}(\|C(x_k)\|)$.

We also prove that if the algorithm converges to a point where the reduced Hessian is positive definite, then the penalty parameter ρ_k is uniformly bounded and the trust-region radius δ_k is uniformly bounded away from zero, a desired property of a trust-region algorithm. In this case, particular choices of the multipliers and of the components s^{n} and s^{t} lead us to a q-quadratic rate of convergence in x.

A detailed treatment of the global convergence theory is given in Vicente [35].

The structure of the trust-region subproblem for the linearized constraints can be exploited to obtain some interesting results. We introduce a quasi-normal component that satisfies a fraction of optimal decrease on the trust-region subproblem for the linearized constraints. We show that the (exact) normal component shares this property. We also prove that if the algorithm is well behaved (for instance, if the trust radius is uniformly bounded away from zero), then this subproblem has a natural tendency to fall into the so-called hard case.

We review the notation used in this paper. The Lagrangian function associated with the ECO problem (1.1) is defined by $\ell(x, \lambda) = f(x) + \lambda^T C(x)$, where $\lambda \in \mathbb{R}^m$ is the Lagrange multiplier vector. The matrix $\nabla C(x)$ is given by $\nabla C(x) = (\nabla c_1(x) \cdots \nabla c_m(x))$, where $\nabla c_i(x)$ represents the gradient of the function $c_i(x)$. Let $\nabla^2 f(x)$ and $\nabla^2 c_i(x)$ be the Hessian matrices of $f(x)$ and $c_i(x)$, respectively. We use subscripted indices to represent the evaluation of a function at a particular point of the sequences $\{x_k\}$ and $\{\lambda_k\}$. For instance, f_k represents $f(x_k)$ and ℓ_k is the same as $\ell(x_k, \lambda_k)$. The vector and matrix norms used are the ℓ_2 norms, and I_l represents the identity matrix of order l. Finally, $\lambda_1(A)$ denotes the smallest eigenvalue of the symmetric matrix A.

The structure of this paper is as follows. In section 2, we introduce the trust-region subproblems and outline the general trust-region algorithm and the general assumptions. In section 3, we present the global convergence theory. A class of discretized optimal control problems is introduced in section 4 as a justification for the general form of our algorithms and theory. In sections 5 and 6, we analyze respectively the behavior of the trust radius and the local rates of convergence. The trust-region subproblem for the linearized constraints is studied in section 7. We end this paper with some summary conclusions.

2. Algorithm and general assumptions.

The trust-region algorithm analyzed by Dennis, El-Alem, and Maciel [7] for the solution of the ECO problem (1.1) consists of computing, at each iteration k, a step s_k decomposed as $s_k = s_k^{\mathsf{n}} + s_k^{\mathsf{t}}$, where the components s_k^{n} and s_k^{t} are required to satisfy given conditions. If the step s_k is accepted, the algorithm continues by setting x_{k+1} to $x_k + s_k$. If the step is rejected then $x_{k+1} = x_k$.

2.1. Decomposition of the step.

Suppose that $\|C_k\| \neq 0$. The component s_k^{n} is called the quasi-normal (or quasi-vertical) component of s_k and is required to satisfy a fraction of Cauchy decrease on the trust-region subproblem for the linearized constraints defined by

$$\begin{array}{rl} \text{minimize} & \frac{1}{2}\|\nabla C_k^T s^{\mathsf{n}} + C_k\|^2 \\ \text{subject to} & \|s^{\mathsf{n}}\| \le r\delta_k, \end{array}$$

where $r \in (0,1)$ and δ_k is the trust radius. In other words, s_k^{n} has to satisfy

$$\|C_k\|^2 - \|\nabla C_k^T s_k^{\mathsf{n}} + C_k\|^2 \ge \sigma^{\mathsf{n}} \left(\|C_k\|^2 - \|\nabla C_k^T\, c_k^{\mathsf{n}} + C_k\|^2 \right), \tag{2.1}$$

where $\sigma^{\mathsf{n}} > 0$ and c_k^{n} is the so-called Cauchy point for this trust-region subproblem, i.e., c_k^{n} is the optimal solution of

$$\begin{array}{rl} \text{minimize} & \frac{1}{2}\|\nabla C_k^T c^{\mathsf{n}} + C_k\|^2 \\ \text{subject to} & c^{\mathsf{n}} \in span\{-\nabla C_k C_k\}, \\ & \|c^{\mathsf{n}}\| \le r\delta_k, \end{array}$$

and therefore

$$c_k^{\mathsf{n}} = \begin{cases} -\frac{\|\nabla C_k C_k\|^2}{\|\nabla C_k^T \nabla C_k C_k\|^2} \nabla C_k C_k & \text{if } \frac{\|\nabla C_k C_k\|^3}{\|\nabla C_k^T \nabla C_k C_k\|^2} \le r\delta_k, \\ -\frac{r\delta_k}{\|\nabla C_k C_k\|} \nabla C_k C_k & \text{otherwise.} \end{cases}$$

The component s_k^{n} is also required to satisfy the condition

$$\|s_k^{\mathsf{n}}\| \le \kappa_1 \|C_k\|, \tag{2.2}$$

where κ_1 is a positive constant independent of the iterate k of the algorithm. This condition is saying that close to feasibility the quasi-normal component has to be small.

In this paper, we require the quasi-normal component s_k^{n} also to satisfy

$$\nabla_x \ell_k^T s_k^{\mathsf{n}} \leq \kappa_2 \|C_k\| \delta_k, \tag{2.3}$$

where κ_2 is a positive constant independent of the iterates. The important consequence of this condition is that if $\|C_k\|$ is small compared with δ_k, then any increase of the quadratic model of the Lagrangian along the quasi-normal component s_k^{n} is of $\mathcal{O}(\delta_k^2)$.

The two choices of s_k^{n} given in sections 4.1 and 4.2 satisfy conditions (2.1), (2.2), and (2.3). Other choices have been suggested in [7], [20].

The component s_k^{t} is the tangential (or horizontal) component, and it must satisfy

$$\nabla C_k^T s_k^{\mathsf{t}} = 0;$$

i.e., it must lie in the null space $\mathcal{N}(\nabla C_k^T)$ of ∇C_k^T. Let W_k be an $n \times (n-m)$ matrix whose columns form a basis for $\mathcal{N}(\nabla C_k^T)$. Let also

$$q_k(s) = \ell_k + \nabla_x {\ell_k}^T s + \frac{1}{2} s^T H_k s$$

be a quadratic model of ℓ at (x_k, λ_k), where H_k is an approximation to $\nabla^2_{xx}\ell(x_k, \lambda_k)$.

Since for any $s^{\mathsf{t}} \in \mathcal{N}(\nabla C_k^T)$, there exists a $\bar{s}^{\mathsf{t}} \in \mathbb{R}^{n-m}$ such that $s^{\mathsf{t}} = W_k \bar{s}^{\mathsf{t}}$, we consider also $\bar{q}_k^{\mathsf{t}}(\bar{s}^{\mathsf{t}})$, which is given by

$$\bar{q}_k^{\mathsf{t}}(\bar{s}^{\mathsf{t}}) = q_k(s_k^{\mathsf{n}} + W_k \bar{s}^{\mathsf{t}}) = q_k(s_k^{\mathsf{n}}) + \bar{g}_k^T \bar{s}^{\mathsf{t}} + \frac{1}{2} (\bar{s}^{\mathsf{t}})^T \bar{H}_k (\bar{s}^{\mathsf{t}})$$

with $\bar{H}_k = W_k^T H_k W_k$, $\bar{g}_k = W_k^T \nabla q_k(s_k^{\mathsf{n}})$ and $q_k(s_k^{\mathsf{n}}) = \ell_k + \nabla_x {\ell_k}^T s_k^{\mathsf{n}} + \frac{1}{2}(s_k^{\mathsf{n}})^T H_k (s_k^{\mathsf{n}})$.

If $\|\bar{g}_k\| \neq 0$, $\bar{s}_k^{\mathsf{t}}$ is required to satisfy a fraction of Cauchy decrease for the trust-region subproblem

$$\begin{array}{rl} \text{minimize} & \bar{q}_k^{\mathsf{t}}(\bar{s}^{\mathsf{t}}) \\ \text{subject to} & \|s_k^{\mathsf{n}} + W_k \bar{s}^{\mathsf{t}}\| \leq \delta_k. \end{array}$$

Note that this is not a standard trust-region subproblem because s_k^{n} might not be orthogonal to $\mathcal{N}(\nabla C_k^T)$ and hence $\bar{s}^{\mathsf{t}} = 0$ might not be the center of the trust region. The steepest-descent direction at $\bar{s}^{\mathsf{t}} = 0$ associated with $\bar{q}_k^{\mathsf{t}}(\bar{s}^{\mathsf{t}})$ in the ℓ_2 norm is $-\bar{g}_k$. If we take into account the scaling matrix W_k, then the steepest-descent direction in the $\|W_k \cdot \|$ norm is given by $-(W_k^T W_k)^{-1}\bar{g}_k$. We consider the steepest-descent direction $-\bar{g}_k$ for $\bar{q}_k^{\mathsf{t}}(\bar{s}^{\mathsf{t}})$ on $\{\bar{s}^{\mathsf{t}} \in \mathbb{R}^{n-m} : \ \|s_k^{\mathsf{n}} + W_k \bar{s}^{\mathsf{t}}\| \leq \delta_k\}$ and require $\bar{s}_k^{\mathsf{t}}$ to satisfy

$$q_k(s_k^{\mathsf{n}}) - q_k(s_k^{\mathsf{n}} + W_k \bar{s}_k^{\mathsf{t}}) \geq \bar{\sigma}^{\mathsf{t}} \left(q_k(s_k^{\mathsf{n}}) - q_k(s_k^{\mathsf{n}} + W_k \bar{c}_k^{\mathsf{t}}) \right), \tag{2.4}$$

where $\bar{\sigma}^{\mathsf{t}} > 0$, and $\bar{c}_k^{\mathsf{t}}$ is the Cauchy point for the ℓ_2 norm given by

$$\bar{c}_k^{\mathsf{t}} = \begin{cases} -\frac{\|\bar{g}_k\|^2}{\bar{g}_k^T \bar{H}_k \bar{g}_k} \bar{g}_k & \text{if } \frac{\|\bar{g}_k\|^2 \|W_k \bar{g}_k\|}{\bar{g}_k^T \bar{H}_k \bar{g}_k} \leq \bar{\delta}_k \ \text{ and } \ \bar{g}_k^T \bar{H}_k \bar{g}_k > 0, \\ -\frac{\bar{\delta}_k}{\|W_k \bar{g}_k\|} \bar{g}_k & \text{otherwise,} \end{cases}$$

with $\bar{\delta}_k = \| - \tau_{max} W_k \bar{g}_k\|$ and

$$\tau_{max} = \operatorname{argmax}\{\tau : \ \|s_k^{\mathsf{n}} - \tau W_k \bar{g}_k\| \leq \delta_k\}.$$

This is equivalent to saying that τ_{max} is the maximum steplength along $s_k^{\mathsf{n}} - \tau W_k \bar{g}_k$ allowed inside the trust region defined by δ_k. It is easy to verify that

$$\bar{\delta}_k \in \Big((1-r)\delta_k, (1+r)\delta_k \Big).$$

The results given in this paper hold also if $\bar{c}_k^{\mathsf{t}}$ is defined along $-(W_k^T W_k)^{-1}\bar{g}_k$ provided the sequence $\{\|(W_k^T W_k)^{-1}\|\}$ is bounded. They are valid also if the coupled trust-region constraint $\|s_k^{\mathsf{n}} + W_k \bar{s}^{\mathsf{t}}\| \leq \delta_k$ is decoupled as $\|\bar{s}^{\mathsf{t}}\| \leq \delta_k$. In this latter case the parameter r defining the quasi-normal component s_k^{n} can have any positive value.

A step $\bar{s}_k^{\mathsf{t}}$ that satisfies this requirement can be computed by using Powell's dogleg algorithm [25] or by the conjugate-gradient algorithm adapted for trust regions by Steihaug [32] and Toint [33] (see also [7], [8], [21]).

In order to establish global convergence to a point satisfying the second-order necessary optimality conditions, we need $\bar{s}_k^{\mathsf{t}}$ to satisfy a fraction of optimal decrease on the following trust-region subproblem:

$$\begin{aligned} &\text{minimize} && \bar{q}_k^{\mathsf{t}}(\bar{s}^{\mathsf{t}}) \\ &\text{subject to} && \|W_k \bar{s}^{\mathsf{t}}\| \leq \tilde{\delta}_k, \end{aligned} \tag{2.5}$$

where

$$\tilde{\delta}_k = \begin{cases} \bar{\delta}_k & \text{if } \|\bar{g}_k\| \neq 0 \\ (1-r)\delta_k & \text{otherwise.} \end{cases}$$

In other words, we require $\bar{s}_k^{\mathsf{t}}$ to satisfy the following conditions:

$$\begin{aligned} &\bar{q}_k^{\mathsf{t}}(0) - \bar{q}_k^{\mathsf{t}}(\bar{s}_k^{\mathsf{t}}) \geq \beta_1^{\mathsf{t}} \left(\bar{q}_k^{\mathsf{t}}(0) - \bar{q}_k^{\mathsf{t}}(\bar{s}_k^*) \right), \\ &\|W_k \bar{s}_k^{\mathsf{t}}\| \leq \beta_2^{\mathsf{t}} \tilde{\delta}_k, \end{aligned} \tag{2.6}$$

where β_1^{t}, $\beta_2^{\mathsf{t}} > 0$, and $\bar{s}_k^*$ is the optimal solution of the trust-region subproblem (2.5). This can be accomplished by applying the algorithm of Moré and Sorensen [23] or by using the algorithms recently proposed by Rendl and Wolkowicz [28] and Sorensen [31].

If $\bar{s}_k^{\mathsf{t}}$ satisfies a fraction of optimal decrease on the trust-region subproblem (2.5), then

$$\|s_k\| \leq \|s_k^{\mathsf{n}}\| + \|W_k \bar{s}_k^{\mathsf{t}}\| \leq r\delta_k + \beta_2^{\mathsf{t}} \tilde{\delta}_k \leq (r + \beta_2^{\mathsf{t}}(1+r))\delta_k.$$

If $\bar{s}_k^{\mathsf{t}}$ is required to satisfy only a fraction of Cauchy decrease, then $\|s_k\| = \|s_k^{\mathsf{n}} + W_k \bar{s}_k^{\mathsf{t}}\| \leq \delta_k$. We can combine both cases and write

$$\|s_k\| = \|s_k^{\mathsf{n}} + W_k \bar{s}_k^{\mathsf{t}}\| \leq \kappa_0 \delta_k, \tag{2.7}$$

where $\kappa_0 = \max\{r + \beta_2^{\mathsf{t}}(1+r), 1\}$.

It is also important to note that the definition of $\tilde{\delta}_k$ assures that the fraction of optimal decrease (2.6) implies the fraction of Cauchy decrease (2.4) provided $\beta_2^{\mathsf{t}} \geq 1$.

2.2. General trust-region algorithm. We introduce now the merit function and the corresponding actual and predicted decreases. The merit function used is the augmented Lagrangian

$$L(x, \lambda; \rho) = f(x) + \lambda^T C(x) + \rho C(x)^T C(x),$$

where ρ is the penalty parameter. The actual decrease $ared(s_k; \rho_k)$ at the iteration k is given by

$$ared(s_k; \rho_k) = L(x_k, \lambda_k; \rho_k) - L(x_{k+1}, \lambda_{k+1}; \rho_k).$$

The predicted decrease (see [7]) is the following:

$$pred(s_k; \rho_k) = L(x_k, \lambda_k; \rho_k) - \left(q_k(s_k) + \Delta\lambda_k^T(\nabla C_k^T s_k + C_k) + \rho_k \|\nabla C_k^T s_k + C_k\|^2\right).$$

To update the penalty parameter ρ_k we use the scheme proposed by El-Alem [11].

The Lagrange multipliers λ_k are required to satisfy

$$\|\Delta\lambda_k\| = \|\lambda_{k+1} - \lambda_k\| \leq \kappa_3 \delta_k, \tag{2.8}$$

where κ_3 is a positive constant independent of k. This condition is not necessary for global convergence to a stationary point.

The general trust-region algorithm is given below.

Algorithm 2.1 (general trust-region algorithm).

```
1 Choose $x_0$, $\delta_0$, $\lambda_0$, $H_0$, and $W_0$. Set $\rho_{-1} \geq 1$. Choose $\alpha_1$, $\eta_1$, $\delta_{min}$, $\delta_{max}$, $\bar{\rho}$,
  and $r$ such that $0 < \alpha_1, \eta_1 < 1$, $0 < \delta_{min} \leq \delta_{max}$, $\bar{\rho} > 0$, and $r \in (0, 1)$.
2 For $k = 0, 1, 2, \ldots$ do
    2.1 If $\|C_k\| + \|W_k^T \nabla_x \ell_k\| + \gamma_k = 0$, where $\gamma_k$ is given in (2.10), stop the
        algorithm and use $x_k$ as a solution for the ECO problem (1.1).
    2.2 Set $s_k^{\mathsf{n}} = s_k^{\mathsf{t}} = 0$.
        If $\|C_k\| \neq 0$ then compute $s_k^{\mathsf{n}}$ satisfying (2.1), (2.2), (2.3), and $\|s_k^{\mathsf{n}}\| \leq$
        $r\delta_k$.
        If $\|W_k^T \nabla_x \ell_k\| + \gamma_k \neq 0$ then compute $\bar{s}_k^{\mathsf{t}}$ satisfying (2.6).
        Set $s_k = s_k^{\mathsf{n}} + s_k^{\mathsf{t}} = s_k^{\mathsf{n}} + W_k \bar{s}_k^{\mathsf{t}}$.
    2.3 Compute $\lambda_{k+1}$ satisfying (2.8).
    2.4 Compute $pred(s_k; \rho_{k-1})$:

            $q_k(0) - q_k(s_k) - \Delta\lambda_k^T(\nabla C_k^T s_k + C_k) + \rho_{k-1}\left(\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|^2\right).$

        If $pred(s_k; \rho_{k-1}) \geq \frac{\rho_{k-1}}{2}\left(\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|^2\right)$ then set $\rho_k = \rho_{k-1}$.
        Otherwise set

            $\rho_k = 2\left(\frac{q_k(s_k) - q_k(0) + \Delta\lambda_k^T(\nabla C_k^T s_k + C_k)}{\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|^2}\right) + \bar{\rho}.$

    2.5 If $\frac{ared(s_k;\rho_k)}{pred(s_k;\rho_k)} < \eta_1$, set $\delta_{k+1} = \alpha_1 \|s_k\|$ and reject $s_k$.
        Otherwise accept $s_k$ and choose $\delta_{k+1}$ such that

            $\max\{\delta_{min}, \delta_k\} \leq \delta_{k+1} \leq \delta_{max}.$

    2.6 If $s_k$ was rejected set $x_{k+1} = x_k$ and $\lambda_{k+1} = \lambda_k$. Otherwise set $x_{k+1} =$
        $x_k + s_k$ and $\lambda_{k+1} = \lambda_k + \Delta\lambda_k$.
```

It is important to understand that the role of δ_{min} is just to reset δ_k after a step s_k has been accepted. During the course of finding such a step the trust radius can be decreased below δ_{min}. To our knowledge Zhang, Kim, and Lasdon [37] were the first to suggest this modification. We remark that the rules to update the trust radius in the previous algorithm can be much more complicated but those given suffice to prove convergence results and to understand the trust-region mechanism.

As a direct consequence of the way the penalty parameter is updated, we have the following result.

LEMMA 2.1. *The sequence* $\{\rho_k\}$ *satisfies*

$$\begin{aligned} &\rho_k \geq \rho_{k-1} \geq 1 \quad \text{and} \\ &pred(s_k;\rho_k) \geq \frac{\rho_k}{2}\Big(\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|^2\Big). \end{aligned} \tag{2.9}$$

In order to establish global convergence results, we use the general assumptions given in [7]. These are Assumptions A.1–A.4. However, for global convergence to a point that satisfies the second-order necessary optimality conditions, we also need Assumption A.5. We assume that for all iterations k, x_k and $x_k + s_k$ are in Ω, where Ω is an open subset of $\mathbb{R}^n$.

General assumptions.

A.1 The functions f, c_i, $i = 1, \ldots, m$, are twice continuously differentiable in Ω.

A.2 The gradient matrix $\nabla C(x)$ has full column rank for all $x \in \Omega$.

A.3 The functions f, ∇f, $\nabla^2 f$, C, ∇C, $\nabla^2 c_i$, $i = 1, \ldots, m$, are bounded in Ω. The matrix $(\nabla C(x)^T \nabla C(x))^{-1}$ is uniformly bounded in Ω.

A.4 The sequences $\{W_k\}$, $\{H_k\}$, and $\{\lambda_k\}$ are bounded.

A.5 The Hessian approximation H_k is exact, i.e., $H_k = \nabla^2_{xx}\ell_k$, and $\nabla^2 f$ and $\nabla^2 c_i$, $i = 1, \ldots, m$, are Lipschitz continuous in Ω.

Assumptions A.3 and A.4 are equivalent to the existence of positive constants $\nu_0, \ldots, \nu_9$ such that $|f(x)| \leq \nu_0$, $\|\nabla f(x)\| \leq \nu_1$, $\|\nabla^2 f(x)\| \leq \nu_2$, $\|C(x)\| \leq \nu_3$, $\|\nabla C(x)\| \leq \nu_4$, $\|(\nabla C(x)^T \nabla C(x))^{-1}\| \leq \nu_5$, $\|\nabla^2 c_i(x)\| \leq \nu_6$, $i = 1, \ldots, m$, for all $x \in \Omega$, and $\|W_k\| \leq \nu_7$, $\|H_k\| \leq \nu_8$, and $\|\lambda_k\| \leq \nu_9$ for all k.

2.3. Predicted decrease along the tangential component.

Consider again the trust-region subproblem (2.5). We can use the general assumptions and the structure of this subproblem to obtain a lower bound on the predicted decrease $q_k(s_k^{\mathsf{n}}) - q_k(s_k^{\mathsf{n}} + s_k^{\mathsf{t}})$ along the tangential component of the step.

It follows from the Karush–Kuhn–Tucker conditions that there exists a $\gamma_k \geq 0$ such that

$$\begin{aligned} &\bar{H}_k + \gamma_k W_k^T W_k \text{ is positive semidefinite,} \\ &\Big(\bar{H}_k + \gamma_k W_k^T W_k\Big)\bar{s}_k^* = -\bar{g}_k, \text{ and} \\ &\gamma_k\Big(\tilde{\delta}_k - \|W_k \bar{s}_k^*\|\Big) = 0. \end{aligned} \tag{2.10}$$

(It turns out that these conditions are also sufficient for $\bar{s}_k^*$ to solve the trust-region subproblem (2.5); see Gay [15] and Sorensen [30].) As a consequence of this we can write

$$\bar{q}_k^{\mathsf{t}}(0) - \bar{q}_k^{\mathsf{t}}(\bar{s}_k^*) = \frac{1}{2}\left(\|R_k \bar{s}_k^*\|^2 + \gamma_k \tilde{\delta}_k^2\right) \geq \frac{1}{2}\gamma_k \tilde{\delta}_k^2,$$

where $\bar{H}_k + \gamma_k W_k^T W_k = R_k^T R_k$. Hence, we have

$$q_k(s_k^{\mathsf{n}}) - q_k(s_k^{\mathsf{n}} + s_k^{\mathsf{t}}) = \bar{q}_k^{\mathsf{t}}(0) - \bar{q}_k^{\mathsf{t}}(\bar{s}_k^{\mathsf{t}}) \geq \beta_1^{\mathsf{t}}\left(\bar{q}_k^{\mathsf{t}}(0) - \bar{q}_k^{\mathsf{t}}(\bar{s}_k^*)\right) \geq \frac{1}{2}\beta_1^{\mathsf{t}}(1-r)^2\gamma_k\delta_k^2. \tag{2.11}$$

3. Global convergence. Dennis, El-Alem, and Maciel [7] have proved under Assumptions A.1–A.4 and conditions (2.1), (2.2), and (2.4) that

$$\liminf_{k\to+\infty} \left(\|W_k^T \nabla_x \ell_k\| + \|C_k\|\right) = 0. \tag{3.1}$$

In this section we assume that $\bar{s}_k^{\mathsf{t}}$ satisfies the fraction of optimal decrease (2.6) on the trust-region subproblem (2.5), as well as conditions (2.3), (2.8), and A.5 on s_k^{n}, λ_k, and H_k, respectively, and show that (3.1) can be extended to

$$\liminf_{k\to+\infty} \left(\|W_k^T \nabla_x \ell_k\| + \|C_k\| + \gamma_k\right) = 0. \tag{3.2}$$

The proof of (3.2), although simpler, has the same structure as the proof given in [7].

We prove the result by contradiction, under the supposition that

$$\|W_k^T \nabla_x \ell_k\| + \|C_k\| + \gamma_k > \epsilon_{tol} \tag{3.3}$$

for all k. We start by analyzing the fraction of Cauchy and optimal decrease conditions.

LEMMA 3.1. *Let the general assumptions hold. Then*

$$\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|^2 \geq \kappa_4 \|C_k\| \min\{\kappa_5\|C_k\|, r\delta_k\} \tag{3.4}$$

and

$$q_k(s_k^{\mathsf{n}}) - q_k(s_k) \geq \kappa_6 \|\bar{g}_k\| \min\{\kappa_7\|\bar{g}_k\|, \kappa_8\delta_k\}, \tag{3.5}$$

and, moreover, since $\bar{s}_k^{\mathsf{t}}$ satisfies a fraction of optimal decrease for the trust-region subproblem (2.5),

$$q_k(s_k^{\mathsf{n}}) - q_k(s_k) \geq \kappa_9 \gamma_k \delta_k^2, \tag{3.6}$$

where $\kappa_4, \ldots, \kappa_9$ are positive constants independent of the iterate k.

Proof. The conditions (3.4) and (3.5) are an application of Powell's result (see [26, Theorem 4], [22, Lemma 4.8]) followed by the general assumptions. The condition (3.6) is a restatement of (2.11) with $\kappa_9 = \frac{1}{2}\beta_1^{\mathsf{t}}(1-r)^2$. □

The following inequality is needed in the forthcoming lemmas.

LEMMA 3.2. *If the general assumptions hold, then*

$$q_k(0) - q_k(s_k^{\mathsf{n}}) - \Delta\lambda_k^T(\nabla C_k^T s_k + C_k) \geq -\kappa_{10}\|C_k\|\delta_k, \tag{3.7}$$

where κ_{10} is a positive constant independent of k.

Proof. The term $q_k(0) - q_k(s_k^{\mathsf{n}})$ can be bounded using (2.2), (2.3), and Assumption A.4 in the following way:

$$\begin{aligned} q_k(0) - q_k(s_k^{\mathsf{n}}) &= -\nabla_x \ell_k^T s_k^{\mathsf{n}} - \frac{1}{2}(s_k^{\mathsf{n}})^T H_k (s_k^{\mathsf{n}}) \\ &\geq -\kappa_2\|C_k\|\delta_k - \frac{1}{2}\|H_k\|\,\|s_k^{\mathsf{n}}\|^2 \\ &\geq -\kappa_2\|C_k\|\delta_k - \frac{1}{2}\nu_8 r\kappa_1\|C_k\|\delta_k. \end{aligned}$$

On the other hand, it follows from (2.8) and $\|\nabla C_k^T s_k + C_k\| \le \|C_k\|$ that

$$-\Delta\lambda_k^T(\nabla C_k^T s_k + C_k) \ge -\kappa_3 \|C_k\| \delta_k.$$

If we combine these two bounds we get (3.7) with $\kappa_{10} = \kappa_2 + \frac{1}{2}\nu_8 r \kappa_1 + \kappa_3$. □

The following technical lemma gives us upper bounds on the difference between the actual decrease and the predicted decrease. The proof follows similar arguments as the proof of Lemma 6.3 in [11].

LEMMA 3.3. *Let the general assumptions hold. There exist positive constants* $\bar\kappa_1, \ldots, \bar\kappa_7$ *independent of* k, *such that*

$$\begin{aligned} |ared(s_k;\rho_k) - pred(s_k;\rho_k)| \le{}& \bar\kappa_1 \|s_k\|^3 + \bar\kappa_2 \|\Delta\lambda_k\| \, \|s_k\|^2 \\ &+ \rho_k \Big(\bar\kappa_3 \|s_k\|^3 + \bar\kappa_4 \|C_k\| \, \|s_k\|^2 \Big) \end{aligned} \tag{3.8}$$

and

$$\begin{aligned} |ared(s_k;\rho_k) - pred(s_k;\rho_k)| \le{}& \bar\kappa_5 \|\Delta\lambda_k\| \, \|s_k\|^2 \\ &+ \rho_k \Big(\bar\kappa_6 \|s_k\|^3 + \bar\kappa_7 \|C_k\| \, \|s_k\|^2 \Big). \end{aligned} \tag{3.9}$$

Proof. If we add and subtract $\ell(x_{k+1}, \lambda_k)$ to $ared(s_k;\rho_k) - pred(s_k;\rho_k)$ and expand $\ell(\cdot, \lambda_k)$ around x_k we get

$$\begin{aligned} ared(s_k;\rho_k) - pred(s_k;\rho_k) ={}& \frac{1}{2} s_k^T \Big(H_k - \nabla_{xx}^2 \ell(x_k + \pi_k^1 s_k, \lambda_k) \Big) s_k \\ &+ \Delta\lambda_k^T(-C_{k+1} + C_k + \nabla C_k^T s_k) \\ &- \rho_k \Big(\|C_{k+1}\|^2 - \|\nabla C_k^T s_k + C_k\|^2 \Big) \end{aligned}$$

for some $\pi_k^1 \in (0,1)$. Again, using the Taylor expansion we can write

$$\begin{aligned} ared(s_k;\rho_k) - pred(s_k;\rho_k) ={}& \frac{1}{2} s_k^T \Big(H_k - \nabla_{xx}^2 \ell(x_k + \pi_k^1 s_k, \lambda_k) \Big) s_k \\ &- \frac{1}{2} \sum_{i=1}^m (\Delta\lambda_k)_i s_k^T \nabla^2 c_i(x_k + \pi_k^2 s_k) s_k \\ &- \rho_k \left(\sum_{i=1}^m c_i(x_k + \pi_k^3 s_k)(s_k)^T \nabla^2 c_i(x_k + \pi_k^3 s_k)(s_k) \right. \\ &+ (s_k)^T \nabla C(x_k + \pi_k^3 s_k) \nabla C(x_k + \pi_k^3 s_k)^T (s_k) \\ &\left. - (s_k)^T \nabla C(x_k) \nabla C(x_k)^T (s_k) \right), \end{aligned}$$

where $\pi_k^2,\ \pi_k^3 \in (0,1)$. Now we expand $c_i(x_k + \pi_k^3 s_k)$ around $c_i(x_k)$. This and the general assumptions give us the estimate (3.8) for some positive constants $\bar\kappa_1, \ldots, \bar\kappa_4$.

The inequality (3.9) follows from (3.8) and $\rho_k \ge 1$. □

The following three lemmas bound the predicted decrease. They correspond respectively to Lemmas 7.6, 7.7, and 7.8 given in [7].

LEMMA 3.4. *Let the general assumptions hold. Then the predicted decrease in the merit function satisfies*

$$\begin{aligned} pred(s_k;\rho) \geq{}& \kappa_6\|\bar{g}_k\| \min\{\kappa_7\|\bar{g}_k\|, \kappa_8\delta_k\} - \kappa_{10}\|C_k\|\delta_k \\ &+\rho\left(\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|\right)^2, \end{aligned} \tag{3.10}$$

and also

$$pred(s_k;\rho) \;\geq\; \kappa_9\gamma_k\delta_k^2 - \kappa_{10}\|C_k\|\delta_k + \;\rho\left(\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|\right)^2, \tag{3.11}$$

for any $\rho > 0$.

Proof. The two conditions (3.10) and (3.11) follow from a direct application of (3.7) and from the two different lower bounds (3.5) and (3.6) on $q_k(s_k^{\mathsf{n}}) - q_k(s_k)$. □

LEMMA 3.5. *Let the general assumptions hold, and assume that* $\|W_k^T\nabla_x\ell_k\| + \|C_k\| + \gamma_k > \epsilon_{tol}$. *If* $\|C_k\| \leq \alpha\delta_k$, *where* α *satisfies*

$$\alpha \;\leq\; \min\left\{\frac{\epsilon_{tol}}{3\delta_{max}}, \frac{\epsilon_{tol}}{6\nu_7\nu_8\kappa_1\delta_{max}}, \frac{\kappa_6\epsilon_{tol}}{12\kappa_{10}\delta_{max}} \min\left\{\frac{\kappa_7\epsilon_{tol}}{6\delta_{max}}, \kappa_8\right\}, \frac{\kappa_9\epsilon_{tol}}{6\kappa_{10}}\right\}, \tag{3.12}$$

then the predicted decrease in the merit function satisfies either

$$pred(s_k;\rho) \;\geq\; \frac{\kappa_6}{2}\|\bar{g}_k\| \min\{\kappa_7\|\bar{g}_k\|, \kappa_8\delta_k\} + \rho\left(\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|^2\right) \tag{3.13}$$

or

$$pred(s_k;\rho) \geq \frac{\kappa_9}{2}\gamma_k\delta_k^2 + \rho\left(\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|^2\right) \tag{3.14}$$

for any $\rho > 0$.

Proof. From $\|W_k^T\nabla_x\ell_k\| + \|C_k\| + \gamma_k > \epsilon_{tol}$ and the first bound on α given by (3.12), we get

$$\|W_k^T\nabla_x\ell_k\| + \gamma_k > \frac{2}{3}\epsilon_{tol}.$$

Thus either $\|W_k^T\nabla_x\ell_k\| > \frac{1}{3}\epsilon_{tol}$ or $\gamma_k > \frac{1}{3}\epsilon_{tol}$. Let us first assume that $\|W_k^T\nabla_x\ell_k\| > \frac{1}{3}\epsilon_{tol}$. Then it follows from the second bound on α given by (3.12) that

$$\begin{aligned} \|\bar{g}_k\| &= \|W_k^T\nabla_x\ell_k + W_k^T H_k s_k^{\mathsf{n}}\| \\ &\geq \|W_k^T\nabla_x\ell_k\| - \|W_k^T H_k s_k^{\mathsf{n}}\| \\ &\geq \frac{1}{3}\epsilon_{tol} - \nu_7\nu_8\kappa_1\|C_k\| \\ &\geq \frac{1}{6}\epsilon_{tol}. \end{aligned}$$

Using this, (3.10), $\delta_k \leq \delta_{max}$, and the third bound on α given by (3.12), we obtain

$$\begin{aligned} pred(s_k;\rho) \geq{}& \frac{\kappa_6}{2}\|\bar{g}_k\| \min\{\kappa_7\|\bar{g}_k\|, \kappa_8\delta_k\} + \frac{\kappa_6\epsilon_{tol}}{12}\min\left\{\frac{\kappa_7\epsilon_{tol}}{6}, \kappa_8\delta_k\right\} \\ &- \kappa_{10}\delta_{max}\|C_k\| + \rho\left(\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|^2\right) \\ \geq{}& \frac{\kappa_6}{2}\|\bar{g}_k\| \min\{\kappa_7\|\bar{g}_k\|, \kappa_8\delta_k\} + \rho\left(\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|^2\right). \end{aligned}$$

Now suppose that $\gamma_k > \frac{1}{3}\epsilon_{tol}$. To establish (3.14), we combine (3.11) and the last bound on α given by (3.12) and get

$$\begin{aligned} pred(s_k;\rho) &\geq \frac{\kappa_9}{2}\gamma_k\delta_k^2 + \left(\frac{\kappa_9}{6}\epsilon_{tol}\delta_k - \kappa_{10}\|C_k\|\right)\delta_k + \rho\left(\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|^2\right) \\ &\geq \frac{\kappa_9}{2}\gamma_k\delta_k^2 + \rho\left(\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|^2\right). \qquad \square \end{aligned}$$

We can set ρ to ρ_{k-1} in Lemma 3.5 and conclude that if $\|W_k^T\nabla_x\ell_k\| + \|C_k\| + \gamma_k > \epsilon_{tol}$ and $\|C_k\| \leq \alpha\delta_k$, then the penalty parameter at the current iterate does not need to be increased. See step 2.4 of Algorithm 2.1.

The proof of the next lemma follows the argument given in the proof of Lemma 3.5 to show that either $\|\bar{g}_k\| > \frac{1}{6}\epsilon_{tol}$ or $\gamma_k > \frac{1}{3}\epsilon_{tol}$ holds.

LEMMA 3.6. *Let the general assumptions hold, and assume that* $\|W_k^T\nabla_x\ell_k\| + \|C_k\| + \gamma_k > \epsilon_{tol}$. *If* $\|C_k\| \leq \alpha\delta_k$, *where* α *satisfies* (3.12), *then there exists a constant* $\kappa_{11} > 0$ *such that*

$$pred(s_k;\rho_k) \geq \kappa_{11}\delta_k^2. \tag{3.15}$$

Proof. By Lemma 3.5 we know that either (3.13) or (3.14) holds. Now we set $\rho = \rho_k$. In the first case we use $\|\bar{g}_k\| > \frac{1}{6}\epsilon_{tol}$ and get

$$\begin{aligned} pred(s_k;\rho_k) &\geq \frac{\kappa_6\epsilon_{tol}}{12}\min\left\{\frac{\kappa_7\epsilon_{tol}}{6}, \kappa_8\delta_k\right\} \\ &\geq \frac{\kappa_6\epsilon_{tol}}{12}\min\left\{\frac{\kappa_7\epsilon_{tol}}{6\delta_{max}}, \kappa_8\right\}\delta_k \\ &\geq \frac{\kappa_6\epsilon_{tol}}{12\delta_{max}}\min\left\{\frac{\kappa_7\epsilon_{tol}}{6\delta_{max}}, \kappa_8\right\}\delta_k^2. \end{aligned}$$

In the second case we use $\gamma_k > \frac{1}{3}\epsilon_{tol}$, to obtain

$$pred(s_k;\rho_k) \geq \frac{\kappa_9\epsilon_{tol}}{6}\delta_k^2.$$

Hence (3.15) holds with

$$\kappa_{11} = \min\left\{\frac{\kappa_6\epsilon_{tol}}{12\delta_{max}}\min\left\{\frac{\kappa_7\epsilon_{tol}}{6\delta_{max}}, \kappa_8\right\}, \frac{\kappa_9\epsilon_{tol}}{6}\right\}. \qquad \square$$

Next we prove under the supposition (3.3) that the penalty parameter ρ_k is bounded.

LEMMA 3.7. *Let the general assumptions hold. If* $\|W_k^T\nabla_x\ell_k\| + \|C_k\| + \gamma_k > \epsilon_{tol}$ *for all k, then*

$$\rho_k \leq \rho_*,$$

where ρ_* *does not depend on k, and thus* $\{\rho_k\}$ *and* $\{L_k\}$ *are bounded sequences.*

Proof. If ρ_k is increased at iteration k, then it is updated according to the rule

$$\rho_k = 2\left(\frac{q_k(s_k) - q_k(0) + \Delta\lambda_k^T(\nabla C_k^T s_k + C_k)}{\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|^2}\right) + \bar{\rho}.$$

We can write

$$\begin{aligned}\frac{\rho_k}{2}\left(\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|^2\right) = \nabla_x \ell(x_k, \lambda_k)^T s_k^{\mathsf{n}} + \frac{1}{2}(s_k^{\mathsf{n}})^T H_k (s_k^{\mathsf{n}}) \\ -(q_k(s_k^{\mathsf{n}}) - q_k(s_k)) + \Delta\lambda_k^T(\nabla C_k^T s_k + C_k) \\ +\frac{\bar{\rho}}{2}\Big(\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|^2\Big).\end{aligned}$$

By applying (3.4) to the left-hand side and (3.5) and (3.7) to the right-hand side, we obtain

$$\begin{aligned}\frac{\rho_k}{2}\kappa_4\|C_k\| \min\{\kappa_5\|C_k\|, r\delta_k\} &\leq \kappa_{10}\delta_k\|C_k\| + \frac{\bar{\rho}}{2}\Big(-2(\nabla C_k C_k)^T s_k - \|\nabla C_k^T s_k\|^2\Big) \\ &\leq (\kappa_{10} + \bar{\rho}\kappa_0\nu_4)\delta_k\|C_k\|.\end{aligned}$$

If ρ_k is increased at iteration k, then from Lemma 3.5 we certainly know that $\|C_k\| > \alpha\delta_k$, where α satisfies (3.12). Now we use this fact to establish that

$$\Big(\frac{\kappa_4}{2}\min\{\kappa_5\alpha, r\}\Big)\,\rho_k \leq \kappa_{10} + \bar{\rho}\kappa_0\nu_4.$$

We have proved that $\{\rho_k\}$ is bounded. From this and the general assumptions we conclude that $\{L_k\}$ is also bounded. □

We can prove also under the supposition (3.3), that the trust radius is bounded away from zero.

LEMMA 3.8. *Let the general assumptions hold. If* $\|W_k^T\nabla_x\ell_k\| + \|C_k\| + \gamma_k > \epsilon_{tol}$ *for all k, then*

$$\delta_k \geq \delta_* > 0,$$

where δ_* *does not depend on k.*

Proof. If s_{k-1} was an acceptable step, then $\delta_k \geq \delta_{min}$. If not then $\delta_k = \alpha_1\|s_{k-1}\|$, and we consider the cases $\|C_{k-1}\| \leq \alpha\delta_{k-1}$ and $\|C_{k-1}\| > \alpha\delta_{k-1}$, where α satisfies (3.12). In both cases we use the fact

$$1 - \eta_1 \leq \left|\frac{ared(s_{k-1};\rho_{k-1})}{pred(s_{k-1};\rho_{k-1})} - 1\right|.$$

Case I. $\|C_{k-1}\| \leq \alpha\delta_{k-1}$. From Lemma 3.6, inequality (3.15) holds for $k = k-1$. Thus we can use $\|s_{k-1}\| \leq \kappa_0\delta_{k-1}$, (2.8), and (3.9) with $k = k-1$ to obtain

$$\left|\frac{ared(s_{k-1};\rho_{k-1})}{pred(s_{k-1};\rho_{k-1})} - 1\right| \leq \frac{(\bar{\kappa}_5\kappa_0\kappa_3\delta_{k-1}^2 + \rho_*\bar{\kappa}_6\kappa_0^2\delta_{k-1}^2 + \rho_*\bar{\kappa}_7\alpha\kappa_0\delta_{k-1}^2)\|s_{k-1}\|}{\kappa_{11}\delta_{k-1}^2}.$$

Thus $\delta_k = \alpha_1\|s_{k-1}\| \geq \frac{\alpha_1(1-\eta_1)\kappa_{11}}{\bar{\kappa}_5\kappa_0\kappa_3 + \rho_*\bar{\kappa}_6\kappa_0^2 + \rho_*\bar{\kappa}_7\alpha\kappa_0} \equiv \kappa_{12}$.

Case II. $\|C_{k-1}\| > \alpha\delta_{k-1}$. In this case from (2.9) and (3.4) with $k = k-1$ we get

$$\begin{aligned}pred(s_{k-1};\rho_{k-1}) &\geq \frac{\rho_{k-1}}{2}\kappa_4\|C_{k-1}\|\min\{\kappa_5\|C_{k-1}\|, r\delta_{k-1}\} \\ &\geq \rho_{k-1}\kappa_{13}\delta_{k-1}\|C_{k-1}\| \\ &\geq \rho_{k-1}\alpha\kappa_{13}\delta_{k-1}^2,\end{aligned}$$

where $\kappa_{13} = \frac{\kappa_4}{2}\min\{\kappa_5\alpha, r\}$. Again we use $\rho_{k-1} \geq 1$, (2.8), and (3.9) with $k = k-1$, and this time the last two lower bounds on $pred(s_{k-1};\rho_{k-1})$, and write

$$\begin{aligned}\left|\frac{ared(s_{k-1};\rho_{k-1})}{pred(s_{k-1};\rho_{k-1})} - 1\right| &\leq \frac{\rho_{k-1}(\bar{\kappa}_5\kappa_0\kappa_3 + \bar{\kappa}_6\kappa_0^2)\delta_{k-1}^2\|s_{k-1}\|}{\rho_{k-1}\alpha\kappa_{13}\delta_{k-1}^2} \\ &\quad + \frac{\rho_{k-1}\bar{\kappa}_7\kappa_0\delta_{k-1}\|C_{k-1}\|\,\|s_{k-1}\|}{\rho_{k-1}\kappa_{13}\delta_{k-1}\|C_{k-1}\|} \\ &\leq \left(\frac{\bar{\kappa}_5\kappa_0\kappa_3 + \bar{\kappa}_6\kappa_0^2 + \bar{\kappa}_7\alpha\kappa_0}{\alpha\kappa_{13}}\right)\|s_{k-1}\|.\end{aligned}$$

Hence $\delta_k = \alpha_1\|s_{k-1}\| \geq \frac{\alpha_1(1-\eta_1)\alpha\kappa_{13}}{\bar{\kappa}_5\kappa_0\kappa_3 + \bar{\kappa}_6\kappa_0^2 + \bar{\kappa}_7\alpha\kappa_0} \equiv \kappa_{14}$.

The result follows by setting $\delta_* = \min\{\delta_{min}, \kappa_{12}, \kappa_{14}\}$. □

The next result is needed also for the forthcoming Theorem 3.10.

LEMMA 3.9. *Let the general assumptions hold. If* $\|W_k^T\nabla_x\ell_k\| + \|C_k\| + \gamma_k > \epsilon_{tol}$ *for all k, then an acceptable step is always found in finitely many trial steps.*

Proof. Let us prove the assertion by contradiction. Assume that for a given $\bar{k}$, $x_k = x_{\bar{k}}$ for all $k \geq \bar{k}$. This means that $\lim_{k\to+\infty}\delta_k = 0$ and all steps are rejected after iteration $\bar{k}$. See steps 2.5 and 2.6 of Algorithm 2.1. We can consider the cases $\|C_k\| \leq \alpha\delta_k$ and $\|C_k\| > \alpha\delta_k$, where α satisfies (3.12), and appeal to arguments similar to those used in Lemma 3.8 to conclude that in any case

$$\left|\frac{ared(s_k;\rho_k)}{pred(s_k;\rho_k)} - 1\right| \leq \kappa_{15}\delta_k, \quad k \geq \bar{k}\,,$$

where κ_{15} is a positive constant independent of the iterates. Since we are assuming that $\lim_{k\to+\infty}\delta_k = 0$, we have $\lim_{k\to+\infty}\frac{ared(s_k;\rho_k)}{pred(s_k;\rho_k)} = 1$, and this contradicts the rules that update the trust radius. See step 2.5 of Algorithm 2.1. □

Now we finally can state our first asymptotic result.

THEOREM 3.10. *Under the general assumptions, the sequence of iterates* $\{x_k\}$ *generated by the Algorithm* 2.1 *satisfies*

$$\liminf_{k\to+\infty}\left(\|W_k^T\nabla_x\ell_k\| + \|C_k\| + \gamma_k\right) = 0. \tag{3.16}$$

Proof. Suppose that there exists an $\epsilon_{tol} > 0$ such that $\|W_k^T\nabla_x\ell_k\| + \|C_k\| + \gamma_k > \epsilon_{tol}$ for all k.

At each iteration k either $\|C_k\| \leq \alpha\delta_k$ or $\|C_k\| > \alpha\delta_k$, where α satisfies (3.12). In the first case we appeal to Lemmas 3.6 and 3.8 and obtain

$$pred(s_k;\rho_k) \geq \kappa_{11}\delta_*^2.$$

If $\|C_k\| > \alpha\delta_k$, we have from $\rho_k \geq 1$, (2.9), (3.4), and Lemma 3.8 that

$$pred(s_k;\rho_k) \geq \frac{\kappa_4}{2}\alpha\min\{\kappa_5\alpha, r\}\delta_*^2.$$

Hence there exists a positive constant κ_{16} not depending on k such that $pred(s_k;\rho_k) \geq \kappa_{16}$. From Lemma 3.9, we can ignore the rejected steps and work only with successful iterates. So, without loss of generality, we have

$$L_k - L_{k+1} = ared(s_k;\rho_k) \geq \eta_1 pred(s_k;\rho_k) \geq \eta_1\kappa_{16}.$$

Now, if we let k go to infinity, this contradicts the boundedness of $\{L_k\}$. □

From this result we can state our global convergence result: existence of a limit point of the sequence of iterates generated by the algorithm satisfying the second-order necessary optimality conditions. This result generalizes those obtained for unconstrained optimization by Sorensen [30] and Moré and Sorensen [23].

THEOREM 3.11. *Let the general assumptions hold. Assume that $W(x)$ and $\lambda(x)$ are continuous functions and $\lambda_k = \lambda(x_k)$ for all k.*

If $\{x_k\}$ is a bounded sequence generated by Algorithm 2.1, *then there exists a limit point x_* such that*

- $C(x_*) = 0$,
- $W(x_*)^T \nabla f(x_*) = 0$, *and*
- $\nabla^2_{xx}\ell(x_*, \lambda(x_*))$ *is positive semidefinite on* $\mathcal{N}(\nabla C(x_*)^T)$.

Moreover, if $\lambda(x_*)$ is such that $\nabla_x \ell(x_*, \lambda(x_*)) = 0$ then x_* satisfies the second-order necessary optimality conditions.

Proof. Let $\{k_i\}$ be the index subsequence considered in (3.16). Since $\{x_{k_i}\}$ is bounded, it has a subsequence $\{x_{k_j}\}$ that converges to a point x_* and for which

$$\lim_{j\to+\infty} \left(\|W_{k_j}^T \nabla_x \ell_{k_j}\| + \|C_{k_j}\| + \gamma_{k_j} \right) = 0. \tag{3.17}$$

Now from this and the continuity of $C(x)$, we get $C(x_*) = 0$. Then we use the continuity of $W(x)$ and $\nabla f(x)$ to obtain

$$W(x_*)^T \nabla f(x_*) = 0.$$

Since $\lambda_1(\cdot)$ is a continuous function, we can use (2.10), $\lim_{j\to+\infty} \gamma_{k_j} = 0$, the continuity of $W(x)$, $\lambda(x)$, and of the second derivatives of $f(x)$ and $c_i(x)$, $i = 1, \ldots, m$, to obtain

$$\lambda_1 \left(W(x_*)^T \nabla^2_{xx} \ell(x_*, \lambda(x_*)) W(x_*) \right) \geq 0.$$

This shows that $\nabla^2_{xx}\ell(x_*, \lambda(x_*))$ is positive semidefinite on $\mathcal{N}(\nabla C(x_*)^T)$. □

The continuity of an orthogonal null-space basis has been discussed in [1], [5], [16]. A class of nonorthogonal null-space basis is described in section 4.1.

The equation $\nabla_x \ell(x_*, \lambda(x_*)) = 0$ is satisfied for consistent updates of the Lagrange multipliers like the least-squares update (4.7) or the adjoint update (4.3).

4. Examples.

4.1. A class of discretized optimal control problems.

We now introduce an important class of ECO problems where we can find convenient matrices W_k, quasi-normal components s_k^{n}, and multipliers λ_k satisfying all the requirements needed for our analysis. The numerical solution of many discretized optimal control problems involves solving the ECO problem

$$\begin{array}{ll} \text{minimize} & f(y, u) \\ \text{subject to} & C(y, u) = 0, \end{array} \tag{4.1}$$

where $y \in \mathbb{R}^m$, $u \in \mathbb{R}^{n-m}$ and $x = \binom{y}{u}$ (see [8], [19], [20]). The variables in y are the state variables and the variables in u are the control variables. Other applications include parameter identification, inverse, and flow problems and design optimization. In many situations there are bounds on the control variables, but this is not considered here. Another interesting aspect of these problems is that $\nabla C(x)^T$ can be partitioned as

$$\nabla C(x)^T = \left(\begin{array}{cc} C_y(x) & C_u(x) \end{array} \right),$$

where $C_y(x)$ is a square matrix of order m.

In this class of problems the following assumption traditionally is made:

(4.2) The partial Jacobian $C_y(x)$ is nonsingular and its inverse is uniformly bounded in Ω.

As a consequence of this, the columns of

$$W(x) = \left(\begin{array}{c} -C_y(x)^{-1}C_u(x) \\ I_{n-m} \end{array} \right)$$

form a basis for the null space of $\nabla C(x)^T$.

The usual choice for λ_k in these problems is the so-called adjoint multipliers

$$\lambda_k = -C_y(x_k)^{-T}\nabla_y f(x_k). \tag{4.3}$$

It follows directly from the continuity of $\nabla C(x)$ and the uniformly boundedness of $C_y(x)^{-1}$ that $W(x)$ varies continuously with x. Furthermore, $\lambda(x) = -C_y(x)^{-T}\nabla_y f(x)$ is a continuous function of x with bounded derivatives.

Using the structure of the problem we can define the quasi-normal component s_k^{n} (see references [8], [19], [20]) as

$$s_k^{\mathsf{n}} = \left(\begin{array}{c} -\varsigma_k C_y(x_k)^{-1}C_k \\ 0 \end{array} \right), \tag{4.4}$$

where

$$\varsigma_k = \begin{cases} 1 & \text{if } \|C_y(x_k)^{-1}C_k\| \le r\delta_k, \\ \dfrac{r\delta_k}{\|C_y(x_k)^{-1}C_k\|} & \text{otherwise.} \end{cases}$$

As we will see in section 7, the quasi-normal component (4.4) satisfies a fraction of optimal decrease and hence a fraction of Cauchy decrease on the trust-region subproblem for the linearized constraints.

Other choices for quasi-normal components are given in [20]. All these quasi-normal components are of the form

$$s_k^{\mathsf{n}} = \left(\begin{array}{c} (s_k^{\mathsf{n}})_y \\ 0 \end{array} \right). \tag{4.5}$$

LEMMA 4.1. *If* s_k^{n} *verifies* (4.5) *and* λ_k *is given by* (4.3), *then conditions* (2.3) *and* (2.8) *are satisfied.*

Proof. From (4.3) and (4.5) we can see that

$$\nabla_x \ell_k^T s_k^{\mathsf{n}} = \left(\begin{array}{c} 0 \\ \nabla_u f(x_k) + C_u(x_k)^T\lambda_k \end{array} \right)^T \left(\begin{array}{c} (s_k^{\mathsf{n}})_y \\ 0 \end{array} \right) = 0$$

and condition (2.3) is trivially satisfied. Condition (2.8) follows from the existence of bounded derivatives for $\lambda(x) = -C_y(x)^{-T}\nabla_y f(x)$ in Ω. □

4.2. The normal component and the least-squares multipliers. Consider again the general ECO problem (1.1). If the component s_k^{n} of the step s_k is orthogonal to the null space of ∇C_k^T, then it is a multiple of $\nabla C_k(\nabla C_k^T \nabla C_k)^{-1} C_k$. If we also require that s_k^{n} lies inside the trust region of radius $r\delta_k$, then it is given by

$$s_k^{\mathsf{n}} = \begin{cases} -\nabla C_k(\nabla C_k^T \nabla C_k)^{-1} C_k & \text{if } \ \|\nabla C_k(\nabla C_k^T \nabla C_k)^{-1} C_k\| \le r\delta_k, \\ -\xi_k \nabla C_k(\nabla C_k^T \nabla C_k)^{-1} C_k & \text{otherwise,} \end{cases} \tag{4.6}$$

where $\xi_k = \frac{r\delta_k}{\|\nabla C_k(\nabla C_k^T \nabla C_k)^{-1} C_k\|}$. If the quasi-normal component s_k^{n} of the step is given by (4.6), then it is called normal. As we will see in the section 7, the normal component (4.6) satisfies a fraction of optimal decrease and hence a fraction of Cauchy decrease on the trust-region subproblem for the linearized constraints.

LEMMA 4.2. *The quasi-normal component* (4.6) *and the least-squares update*

$$\lambda_k = -(\nabla C_k^T \nabla C_k)^{-1} \nabla C_k^T \nabla f_k \tag{4.7}$$

satisfy conditions (2.3) *and* (2.8).

Proof. It can be easily confirmed that $\nabla_x \ell_k^T s_k^{\mathsf{n}} = 0$. The condition (2.8) holds since $\lambda(x) = -(\nabla C(x)^T \nabla C(x))^{-1} \nabla C(x)^T \nabla f(x)$ has bounded derivatives in Ω. □

5. The behavior of the trust radius. In sections 5 and 6 we no longer need to consider that the tangential component $\bar{s}_k^{\mathsf{t}}$ satisfies a fraction of optimal decrease on the trust-region subproblem (2.5). It suffices to assume the fraction of Cauchy decrease condition (2.4). We assume that the component s_k^{n} satisfies conditions (2.1) and (2.2).

We need to strengthen conditions (2.3) and (2.8) in the following way:

$$\nabla_x \ell_k^T s_k^{\mathsf{n}} \le \kappa_2' \|C_k\| \, \|s_k\|, \tag{5.1}$$

$$\|\Delta\lambda_k\| = \|\lambda_{k+1} - \lambda_k\| \le \kappa_3' \|s_k\|, \tag{5.2}$$

$$\|s_k^{\mathsf{n}}\| \le \kappa_4' \|s_k\|, \tag{5.3}$$

where κ_2', κ_3', and κ_4' are positive constants independent of the iterates. The choices of s_k^{n} and λ_k suggested in section 4 satisfy these requirements as well. See Lemmas 4.1 and 4.2 for the first two conditions. It is obvious that the normal component (4.6) satisfy (5.3). The quasi-normal component (4.4) also satisfies (5.3) (see [35, Lemma 5.6.1]).

The next theorems show that if $\lim_{k\to+\infty} x_k = x_*$ and $\nabla_{xx}^2 \ell(x_*, \lambda(x_*))$ is positive definite on $\mathcal{N}(\nabla C(x_*)^T)$, then the penalty parameter ρ_k is uniformly bounded and the trust radius δ_k is uniformly bounded away from zero.

THEOREM 5.1. *Let the general assumptions hold and $W(x)$ and $\lambda(x)$ be continuous. If $\{x_k\}$ converges to x_* and $\nabla_{xx}^2 \ell(x_*, \lambda(x_*))$ is positive definite on $\mathcal{N}(\nabla C(x_*)^T)$, then $\{\rho_k\}$ is a bounded sequence.*

Proof. First since $\nabla_{xx}^2 \ell(x_*, \lambda(x_*))$ is positive definite on $\mathcal{N}(\nabla C(x_*)^T)$ and $\nabla^2 f(x)$, $\nabla^2 c_i(x)$, $i = 1, \ldots, m$, $W(x)$, and $\lambda(x)$ are continuous functions of x, there exists a neighborhood $\mathcal{N}(x_*)$ of x_* and a $\bar{\gamma} > 0$ such that for any x in $\mathcal{N}(x_*)$,

$$\lambda_1\left(W(x)^T \nabla_{xx}^2 \ell(x, \lambda(x)) W(x)\right) \ge \bar{\gamma}.$$

Since $\bar{q}_k^{\mathsf{t}}(\bar{s}_k^{\mathsf{t}}) - \bar{q}_k^{\mathsf{t}}(0) \le 0$ we can write

$$\frac{1}{2}(\bar{s}_k^{\mathsf{t}})^T \bar{H}_k (\bar{s}_k^{\mathsf{t}}) \le -(\bar{s}_k^{\mathsf{t}})^T \bar{g}_k \le \|\bar{s}_k^{\mathsf{t}}\| \, \|\bar{g}_k^{\mathsf{t}}\|.$$

Thus for all k such that $x_k \in \mathcal{N}(x_*)$ we have

$$\frac{1}{2}\bar{\gamma}\|\bar{s}_k^{\mathsf{t}}\|^2 \leq \|\bar{s}_k^{\mathsf{t}}\|\,\|\bar{g}_k\|,$$

and this implies

$$\|s_k^{\mathsf{t}}\| \leq \frac{2\nu_7}{\bar{\gamma}}\|\bar{g}_k\|. \tag{5.4}$$

Now by using (3.5) and (5.4), we have for all k such that $x_k \in \mathcal{N}(x_*)$ that

$$\begin{aligned} q_k(s_k^{\mathsf{n}}) - q_k(s_k) &\geq \kappa_6\|\bar{g}_k\|\min\{\kappa_7\|\bar{g}_k\|, \kappa_8\delta_k\} \\ &\geq \kappa_{17}\|s_k^{\mathsf{t}}\|^2, \end{aligned} \tag{5.5}$$

where $\kappa_{17} = \frac{\kappa_6\bar{\gamma}}{2\nu_7}\min\{\frac{\kappa_7\bar{\gamma}}{2\nu_7}, \frac{\kappa_8}{1+r}\}$.

Now let $\|C_k\| \leq \alpha'\|s_k\|$ where the positive constant α' is defined later. Using similar arguments as in Lemma 3.2, it follows from (2.2), (5.1), (5.2), $\|C_k\| \leq \alpha'\|s_k\|$, and Assumption A.4 that

$$q_k(0) - q_k(s_k^{\mathsf{n}}) - \Delta\lambda_k^T(\nabla C_k^T s_k + C_k) \geq -\kappa_{10}'\|C_k\|\,\|s_k\|, \tag{5.6}$$

where $\kappa_{10}' = \kappa_2' + \frac{1}{2}\nu_8\kappa_1^2\alpha' + \kappa_3'$.

From (2.2) and $\|C_k\| \leq \alpha'\|s_k\|$ we also get

$$\begin{aligned} \|s_k\|^2 \leq \left(\|s_k^{\mathsf{n}}\| + \|s_k^{\mathsf{t}}\|\right)^2 &\leq 2\|s_k^{\mathsf{n}}\|^2 + 2\|s_k^{\mathsf{t}}\|^2 \\ &\leq 2\alpha'\kappa_1^2\|C_k\|\,\|s_k\| + 2\|s_k^{\mathsf{t}}\|^2, \end{aligned}$$

which together with (5.5) and (5.6) implies

$$\begin{aligned} \mathit{pred}(s_k;\rho) \geq \frac{1}{4}\kappa_{17}\|s_k\| &+ \left(\frac{1}{4}\kappa_{17}\|s_k\| - (\alpha'\kappa_1^2\kappa_{17} + \kappa_{10}')\|C_k\|\right)\|s_k\| \\ &+ \rho\left(\|C_k\|^2 - \|\nabla C_k^T s_k + C_k\|^2\right) \end{aligned} \tag{5.7}$$

for all $\rho > 0$. We now need to impose the following condition on α':

$$\alpha' \leq \frac{\kappa_{17}}{4\alpha'\kappa_1^2\kappa_{17} + 4\kappa_{10}'}. \tag{5.8}$$

Now we set $\rho = \rho_{k-1}$ in (5.7) and conclude that the penalty parameter does not need to be increased if $\|C_k\| \leq \alpha'\|s_k\|$ (see step 2.4 of Algorithm 2.1). Hence, if ρ_k is increased, then $\|C_k\| > \alpha'\|s_k\|$ holds, and by using (5.1)–(5.3) we obtain

$$q_k(0) - q_k(s_k^{\mathsf{n}}) - \Delta\lambda_k^T(\nabla C_k^T s_k + C_k) \geq -\kappa_{10}''\|C_k\|\,\|s_k\|, \tag{5.9}$$

with $\kappa_{10}'' = \kappa_2' + \frac{1}{2}\nu_8\kappa_1\kappa_4' + \kappa_3'$. Recall from the proof of Lemma 3.7 that if ρ_k is increased, then

$$\frac{\rho_k}{2}\kappa_4\|C_k\|\min\left\{\kappa_5\|C_k\|, \frac{r}{\kappa_0}\|s_k\|\right\} \leq (\kappa_{10}'' + \bar{\rho}\nu_4)\|s_k\|\,\|C_k\|,$$

which in turn implies

$$\left(\frac{\kappa_4}{2}\min\left\{\kappa_5\alpha', \frac{r}{\kappa_0}\right\}\right)\rho_k \leq \kappa_{10}'' + \bar{\rho}\nu_4 \iff \rho_k \leq \rho_*'.$$

This completes the proof of the theorem. □

THEOREM 5.2. *Let the general assumptions hold and $W(x)$ and $\lambda(x)$ be continuous. If $\{x_k\}$ converges to x_* and $\nabla^2_{xx}\ell(x_*, \lambda(x_*))$ is positive definite on $\mathcal{N}(\nabla C(x_*)^T)$, then δ_k is uniformly bounded away from zero and eventually all iterations are successful.*

Proof. The proof of the theorem is based on the boundedness of $\{\rho_k\}$. We consider the cases $\|C_k\| > \alpha'\|s_k\|$ and $\|C_k\| \le \alpha'\|s_k\|$, where α' satisfies (5.8).

If $\|C_k\| > \alpha'\|s_k\|$, then from (2.7), (2.9), and (3.4), we find that

$$pred(s_k;\rho_k) \ge \rho_k \frac{\kappa_4}{2}\|C_k\| \min\{\kappa_5\|C_k\|, r\delta_k\} \ge \rho_k\kappa_{18}\|s_k\|^2, \tag{5.10}$$

where $\kappa_{18} = \frac{\kappa_4\alpha'}{2}\min\{\kappa_5\alpha', \frac{r}{\kappa_0}\}$. In this case it follows from (3.9), (5.10), and $\rho_k \ge 1$ that

$$\left|\frac{ared(s_k;\rho_k)}{pred(s_k;\rho_k)} - 1\right| \le \left(\frac{\bar{\kappa}_5\kappa_3'}{\kappa_{18}} + \frac{\bar{\kappa}_6}{\kappa_{18}}\right)\|s_k\| + \frac{\bar{\kappa}_7}{\kappa_{18}}\|C_k\|. \tag{5.11}$$

Now, suppose that $\|C_k\| \le \alpha'\|s_k\|$. From (5.7) with $\rho = \rho_k$ we obtain

$$pred(s_k;\rho_k) \ge \frac{\kappa_{17}}{4}\|s_k\|^2.$$

Now we use (3.9) and $\rho_k \le \rho_*$ to get

$$\left|\frac{ared(s_k;\rho_k)}{pred(s_k;\rho_k)} - 1\right| \le \left(\frac{4\bar{\kappa}_5\kappa_3'}{\kappa_{17}} + \frac{4\bar{\kappa}_6\rho_*}{\kappa_{17}}\right)\|s_k\| + \frac{4\bar{\kappa}_7\rho_*}{\kappa_{17}}\|C_k\|. \tag{5.12}$$

It follows from Theorem 8.4 in [7] that

$$\liminf_{k\to+\infty} \left(\|W_k^T\nabla_x\ell_k\| + \|C_k\|\right) = 0.$$

From this result, the continuity of $C(x)$, and the convergence of $\{x_k\}$ we obtain $\lim_{k\to+\infty}\|C_k\| = 0$.

Finally from (5.11), (5.12), and the limits $\lim_{k\to+\infty} x_k = x_*$, $\lim_{k\to+\infty}\lambda_k = \lambda(x_*)$, and $\lim_{k\to+\infty}\|C_k\| = 0$, we finally get

$$\lim_{k\to+\infty}\left|\frac{ared(s_k;\rho_k)}{pred(s_k;\rho_k)}\right| = 1,$$

which by the rules for updating the trust radius in step 2.5 of Algorithm 2.1 shows that δ_k is uniformly bounded away from zero. □

6. Local rate of convergence. In order to obtain q-quadratic local rates of convergence, we require the reduced tangential component $\bar{s}_k^{\mathsf{t}}$ to satisfy (2.4) and the following condition:

$$\text{if } \bar{H}_k \text{ is positive definite and } \|\bar{H}_k^{-1}\bar{g}_k\| \le \bar{\delta}_k \text{ then } \bar{s}_k^{\mathsf{t}} = -\bar{H}_k^{-1}\bar{g}_k. \tag{6.1}$$

6.1. Discretized optimal control formulation. Consider again problem (4.1) and assume that this problem has the structure described in section 4.1. We can now use Theorem 5.2 to obtain a local rate of convergence.

THEOREM 6.1. *Suppose that the ECO problem is of the form* (4.1). *Let the general assumptions and Assumption* (4.2) *hold and assume that $\{x_k\}$ converges to x_*. In addition to this, let $\bar{s}_k^{\mathsf{t}}$, s_k^{n}, and λ_k be given by* (6.1), (4.4), *and* (4.3).

If $\nabla^2_{xx}\ell(x_*, \lambda_*)$ is positive definite on $\mathcal{N}(\nabla C(x_*)^T)$, where

$$\lambda_* = -C_y(x_*)^{-T}\nabla_y f(x_*),$$

then x_k converges q-quadratically to x_*.

Proof. It can be shown by appealing to Theorem 8.4 in [7] that $\nabla_x \ell(x_*, \lambda_*) = 0$. It follows from Theorem 5.2 that δ_k is uniformly bounded away from zero. Thus there exists a positive integer $\bar{k}$ such that for all $k \geq \bar{k}$, $\bar{s}^{\mathsf{t}}_k = -\bar{H}_k^{-1}\bar{g}_k$ and $s^{\mathsf{n}}_k = \left(\begin{smallmatrix} -C_y(x_k)^{-1}C_k \\ 0 \end{smallmatrix}\right)$. Now the rate of convergence follows from [19]. □

6.2. Normal component and least-squares multipliers. Consider the general ECO problem (1.1) again, and suppose that the quasi-normal component is the normal component (4.6) and λ_k is given by (4.7).

We can now use Theorem 5.2 to obtain the desired local rate of convergence. It is assumed that the orthogonal null-space basis is a continuous function of x.

THEOREM 6.2. *Let the general assumptions hold and assume that $\{x_k\}$ converges to x_*. In addition to this, let $\bar{s}^{\mathsf{t}}_k$, s^{n}_k, and λ_k be given by* (6.1), (4.6), *and* (4.7).

If $\nabla^2_{xx}\ell(x_*, \lambda_*)$ is positive definite on $\mathcal{N}(\nabla C(x_*)^T)$, where

$$\lambda_* = -\left(\nabla C(x_*)^T \nabla C(x_*)\right)^{-1} \nabla C(x_*)^T \nabla f(x_*),$$

then x_k converges q-quadratically to x_*.

Proof. It can be shown by appealing to Theorem 8.4 in [7] that $\nabla_x \ell(x_*, \lambda_*) = 0$. It follows from Theorem 5.2 that δ_k is uniformly bounded away from zero. Thus there exists a positive integer $\bar{k}$ such that for all $k \geq \bar{k}$, $\bar{s}^{\mathsf{t}}_k = -\bar{H}_k^{-1}\bar{g}_k$ and $s^{\mathsf{n}}_k = -\nabla C_k(\nabla C_k^T \nabla C_k)^{-1}C_k$. The q-quadratic rate of convergence follows from [18], [36]. □

7. The trust-region subproblem for the linearized constraints.

In this section we investigate a few aspects of the trust-region subproblem for the linearized constraints

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\|\nabla C_k^T s^{\mathsf{n}} + C_k\|^2 \\ &\text{subject to} && \|s^{\mathsf{n}}\| \leq r\delta_k. \end{aligned} \tag{7.1}$$

First we prove that the normal component (4.6) and the quasi-normal component (4.4) always give a fraction of optimal decrease on this trust-region subproblem.

THEOREM 7.1. *Let the general assumptions hold. Then*

(i) *The normal component* (4.6) *satisfies a fraction of optimal decrease on the trust-region subproblem for the linearized constraints*; *i.e., there exists a positive constant β^{n}_1 such that*

$$\|C_k\|^2 - \|\nabla C_k^T s^{\mathsf{n}}_k + C_k\|^2 \geq \beta^{\mathsf{n}}_1 \left(\|C_k\|^2 - \|\nabla C_k^T s^*_k + C_k\|^2\right), \tag{7.2}$$

where s^*_k is the optimal solution of (7.1).

(ii) *In addition, assume Assumption* (4.2). *The quasi-normal component* (4.4) *satisfies the fraction of optimal decrease* (7.2).

Proof. (i) If $\|\nabla C_k(\nabla C_k^T \nabla C_k)^{-1}C_k\| \leq r\delta_k$, then s^{n}_k solves (7.1) and the result holds for any positive value of β^{n}_1 in $(0, 1]$. If this is not the case, then

$$\|C_k\|^2 - \|\nabla C_k^T s^{\mathsf{n}}_k + C_k\|^2 = \xi_k(2 - \xi_k)\|C_k\|^2 \geq \xi_k\|C_k\|^2 \geq \frac{r\delta_k}{\nu_4\nu_5}\|C_k\|, \tag{7.3}$$

since $\|\nabla C_k(\nabla C_k^T \nabla C_k)^{-1} C_k\| \le \nu_4 \nu_5 \|C_k\|$ and $\xi_k \le 1$.

We also have

$$\begin{aligned}\|C_k\|^2 - \|\nabla C_k^T s_k^* + C_k\|^2 &= -2(\nabla C_k C_k)^T s_k^* - (s_k^*)^T(\nabla C_k \nabla C_k^T)(s_k^*)\\ &\le 2\nu_4 \|C_k\|\,\|s_k^*\| + \nu_4^2 \|s_k^*\|^2\\ &\le 2\nu_4 r\delta_k \|C_k\| + \nu_4^2 r\delta_k \|s_k^*\|\\ &\le (2\nu_4 r + \nu_4^3 \nu_5 r)\delta_k \|C_k\|,\end{aligned}$$

since $\|\nabla C_k(\nabla C_k^T \nabla C_k)^{-1}\|\|C_k\| > r\delta_k \ge \|s_k^*\|$. Combining this last inequality with (7.3) we get

$$\|C_k\|^2 - \|\nabla C_k^T s_k^{\mathsf{n}} + C_k\|^2 \ge \frac{1}{\nu_4^2 \nu_5 (2 + \nu_4^2 \nu_5)} \left(\|C_k\|^2 - \|\nabla C_k^T s_k^* + C_k\|^2\right)$$

and this completes the proof of (i).

(ii) If $\|C_y(x_k)^{-T} C_k\| \le r\delta_k$ then s_k^{n} solves (7.1) and (7.2) holds for any positive value of β_1^{n}. If this is not the case, we have

$$\begin{aligned}\|C_k\|^2 - \|\nabla C_k^T s_k^{\mathsf{n}} + C_k\|^2 &= \|C_k\|^2 - \left\| -\varsigma_k \nabla C_k^T \begin{pmatrix} C_y(x_k)^{-1} C_k \\ 0 \end{pmatrix} + C_k\right\|^2\\ &= \varsigma_k(2 - \varsigma_k)\|C_k\|^2\\ &\ge \frac{r\delta_k}{\nu_{10}} \|C_k\|,\end{aligned} \tag{7.4}$$

where ν_{10} is the uniform bound on $\|C_y(x_k)^{-1}\|$. Now the rest of the proof follows as in (i). ▯

As a consequence of this theorem, we have immediately that the normal component (4.6) and the quasi-normal component (4.4) give a fraction of Cauchy decrease on the trust-region subproblem for the linearized constraints.

To compute a step s_k^{n} that gives a fraction of optimal decrease on the trust-region subproblem for the linearized constraints we can also use the techniques proposed in [23], [28], [31].

In the next theorem we show that the trust-region subproblem (7.1), due to its particular structure, tends to fall in the hard case in the latest stages of the algorithm. This result is relevant in our opinion since the algorithms proposed in [23], [28], [31] deal with the hard case.

The trust-region subproblem (7.1) can be rewritten as

$$\begin{aligned}&\text{minimize} \quad \frac{1}{2} C_k^T C_k + (\nabla C_k C_k)^T s^{\mathsf{n}} + \frac{1}{2}(s^{\mathsf{n}})^T(\nabla C_k \nabla C_k^T)(s^{\mathsf{n}})\\ &\text{subject to} \quad \|s^{\mathsf{n}}\| \le r\delta_k.\end{aligned} \tag{7.5}$$

The matrix $\nabla C_k \nabla C_k^T$ is always positive semidefinite and, under the general assumptions, has rank m. Let $E_k(0)$ denote the eigenspace associated with the eigenvalue 0, i.e., $E_k(0) = \{v_k \in \mathbb{R}^n : \ \nabla C_k \nabla C_k^T v_k = 0\}$. The hard case is defined by the two following conditions:

(a) $(v_k)^T(\nabla C_k C_k) = 0$ for all v_k in $E_k(0)$ and

(b) $\|(\nabla C_k \nabla C_k^T + \mu I_n)^{-1} \nabla C_k C_k\| < r\delta_k$ for all $\mu > 0$.

Theorem 7.2. *Under the general assumptions, if* $\lim_{k \to +\infty} \frac{\|C_k\|}{\delta_k} = 0$ *then there exists a* k_h *such that, for all* $k \ge k_h$*, the trust-region subproblem* (7.5) *falls in the hard case as defined above by* (a) *and* (b).

Proof. First we show that (a) holds at every iteration of the algorithm. If $v_k \in E_k(0)$,

$$\nabla C_k \nabla C_k^T v_k = 0.$$

Multiplying both sides by $(\nabla C_k^T \nabla C_k)^{-1} \nabla C_k^T$ gives us

$$\nabla C_k^T v_k = 0.$$

Thus $(v_k)^T(\nabla C_k C_k) = 0$ for all v_k in $E_k(0)$.

Now we prove that there exists a k_h such that (b) holds for every $k \geq k_h$. Since $g_k(\mu) = \|(\nabla C_k \nabla C_k^T + \mu I_n)^{-1} \nabla C_k C_k\|$ is a monotone strictly decreasing function of μ for $\mu > 0$,

$$\lim_{\mu \to 0^+} g_k(\mu) < r\delta_k$$

is equivalent to $g_k(\mu) < r\delta_k$, for all $\mu > 0$. Also, from the singular value decomposition of ∇C_k, we obtain

$$\lim_{\mu \to 0^+} g_k(\mu) = \| \lim_{\mu \to 0^+} (\nabla C_k \nabla C_k^T + \mu I_n)^{-1} \nabla C_k C_k\| = \|\nabla C_k (\nabla C_k^T \nabla C_k)^{-1} C_k\|.$$

Hence $g_k(\mu) < r\delta_k$ holds for all $\mu > 0$ if and only if $\|\nabla C_k(\nabla C_k^T \nabla C_k)^{-1} C_k\| < r\delta_k$.

Now since $\lim_{k \to +\infty} \frac{\|C_k\|}{\delta_k} = 0$, there exists a k_h such that $\|C_k\| < \frac{r}{\nu_4 \nu_5} \delta_k$ for all $k \geq k_h$. Thus $\|\nabla C_k(\nabla C_k^T \nabla C_k)^{-1} C_k\| \leq \nu_4 \nu_5 \|C_k\| < r\delta_k$ for all $k \geq k_h$, and this completes the proof of the theorem. □

We complete this section with the following corollary.

COROLLARY 7.3. *Under the general assumptions, if* $\lim_{k \to +\infty} \|C_k\| = 0$ *and the trust radius is uniformly bounded away from zero, then there exists a* k_h *such that, for all* $k \geq k_h$*, the trust-region subproblem* (7.5) *falls in the hard case as defined above by (a) and (b).*

Proof. If $\lim_{k \to +\infty} \|C_k\| = 0$ and the trust radius is uniformly bounded away from zero, then $\lim_{k \to +\infty} \frac{\|C_k\|}{\delta_k} = 0$ and Theorem 7.2 can be applied. □

8. Concluding remarks.

In Theorems 3.10 and 3.11 we have established global convergence to a point satisfying the second-order necessary optimality conditions for the general trust-region-based algorithm considered in this paper. In Theorem 5.2 we have proved that the trust radius is, under sufficient second-order optimality conditions, bounded away from zero. With the help of this result we analyzed local rates of convergence for different choices of steps and multipliers. We believe that these results complement the theory developed by Dennis, El-Alem, and Maciel in [7] that proves global convergence to a stationary point. We have also given a detailed analysis of the trust-region subproblem for the linearized constraints.

Acknowledgments. We thank Mahmoud El-Alem with whom we had many discussions about the contents of this paper. We are also grateful to our referees for their careful and insightful reading of this paper.

REFERENCES

[1] R. H. BYRD AND R. B. SCHNABEL, *Continuity of the null space basis and constrained optimization*, Math. Programming, 35 (1986), pp. 32–41.

[2] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, *A trust region algorithm for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 24 (1987), pp. 1152–1170.

[3] R. G. Carter, *On the global convergence of trust region algorithms using inexact gradient information*, SIAM J. Numer. Anal., 28 (1991), pp. 251–265.

[4] M. Celis, J. E. Dennis, and R. A. Tapia, *A trust region strategy for nonlinear equality constrained optimization*, in Numerical Optimization 1984, SIAM, Philadelphia, PA, 1985, pp. 71–82.

[5] T. F. Coleman and D. C. Sorensen, *A note on the computation of an orthonormal basis for the null space of a matrix*, Math. Programming, 29 (1984), pp. 234–242.

[6] T. F. Coleman and W. Yuan, *A New Trust Region Algorithm for Equality Constrained Optimization*, Tech. report TR95–1477, Department of Computer Science, Cornell University, Ithaca, NY, 1995.

[7] J. E. Dennis, M. El-Alem, and M. C. Maciel, *A global convergence theory for general trust-region–based algorithms for equality constrained optimization*, SIAM J. Optim., 7 (1997), pp. 177–207.

[8] J. E. Dennis, M. Heinkenschloss, and L. N. Vicente, *Trust-Region Interior-Point SQP Algorithms for a Class of Nonlinear Programming Problems*, Tech. report TR94–45, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1994 (revised January 1997).

[9] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[10] M. El-Alem, *A Global Gonvergence Theory for a Class of Trust Region Algorithms for Constrained Optimization*, Tech. report TR88–5, Ph.D. thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1988.

[11] M. El-Alem, *A global convergence theory for the Celis–Dennis–Tapia trust–region algorithm for constrained optimization*, SIAM J. Numer. Anal., 28 (1991), pp. 266–290.

[12] M. El-Alem, *Convergence to a Second-Order Point for a Trust-Region Algorithm with a Nonmonotonic Penalty Parameter for Constrained Optimization*, Tech. report TR95–28, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1995.

[13] M. El-Alem, *A robust trust–region algorithm with a non-monotonic penalty parameter scheme for constrained optimization*, SIAM J. Optim., 5 (1995), pp. 348–378.

[14] M. El-Hallabi, *A Global Convergence Theory for Arbitrary Norm Trust-Region Algorithms for Equality Constrained Optimization*, Tech. report TR93–60, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1993 (revised May 1995).

[15] D. M. Gay, *Computing optimal locally constrained steps*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 186–197.

[16] P. E. Gill, W. Murray, M. Saunders, G. W. Stewart, and M. H. Wright, *Properties of a representation of a basis for the null space*, Math. Programming, 33 (1985), pp. 172–186.

[17] P. E. Gill, W. Murray, and M. H. Wright, *Some Theoretical Properties of an Augmented Lagrangian Merit Function*, Tech. report SOL 86–6, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA, 1986.

[18] J. Goodman, *Newton's method for constrained optimization*, Math. Programming, 33 (1985), pp. 162–171.

[19] M. Heinkenschloss, *Projected sequential quadratic programming methods*, SIAM J. Optim., 6 (1996), pp. 373–417.

[20] M. Heinkenschloss and L. N. Vicente, *Analysis of Inexact Trust-Region Interior-Point SQP Algorithms*, Tech. report TR95–18, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1995 (revised April 1996).

[21] M. Lalee, J. Nocedal, and T. Plantenga, *On the implementation of an algorithm for large-scale equality constrained optimization*, SIAM J. Optim., 8 (1998), to appear.

[22] J. J. Moré, *Recent developments in algorithms and software for trust regions methods*, in Mathematical Programming. The State of Art, A. Bachem, M. Grotschel, and B. Korte, eds., Springer-Verlag, New York, 1983, pp. 258–287.

[23] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[24] E. O. Omojokon, *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints*, Ph.D. thesis, Department of Computer Science, University of Colorado, Boulder, CO, 1989.

[25] M. J. D. Powell, *A new algorithm for unconstrained optimization*, in Nonlinear Programming, J. B. Rosen, O. L. Mangasarian, and K. Ritter, eds., Academic Press, New York, 1970.

[26] M. J. D. Powell, *Convergence properties of a class of minimization algorithms*, in Nonlinear Programming 2, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic

Press, New York, 1975, pp. 1–27.

[27] M. J. D. Powell and Y. Yuan, *A trust region algorithm for equality constrained optimization*, Math. Programming, 49 (1991), pp. 189–211.

[28] F. Rendl and H. Wolkowicz, *A semidefinite framework for trust region subproblems with applications to large scale minimization*, Tech. report CORR 94–32, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, 1994.

[29] G. A. Shultz, R. B. Schnabel, and R. H. Byrd, *A family of trust–region–based algorithms for unconstrained minimization with strong global convergence properties*, SIAM J. Numer. Anal., 22 (1985), pp. 47–67.

[30] D. C. Sorensen, *Newton's method with a model trust region modification*, SIAM J. Numer. Anal., 19 (1982), pp. 409–426.

[31] D. C. Sorensen, *Minimization of a large scale quadratic function subject to an spherical constraint*, SIAM J. Optim., 7 (1997), pp. 141–161.

[32] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[33] P. L. Toint, *Towards an efficient sparsity exploiting Newton method for minimization*, in Sparse Matrices and Their Uses, I. S. Duff, ed., Academic Press, New York, 1981, pp. 57–87.

[34] A. Vardi, *A trust region algorithm for equality constrained minimization: Convergence properties and implementation*, SIAM J. Numer. Anal., 22 (1985), pp. 575–591.

[35] L. N. Vicente, *Trust-Region Interior-Point Algorithms for a Class of Nonlinear Programming Problems*, Ph.D. thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1996.

[36] M. H. Wright, *Numerical Methods for Nonlinearly Constrained Optimization*, Computer Science Dept. report CS–76–566, Ph.D. thesis, Stanford University, Stanford, CA, 1976.

[37] J. Zhang, N. Kim, and L. Lasdon, *An improved successive linear programming algorithm*, Management Sci., 31 (1985), pp. 1312–1331.

SIAM J. OPTIM.
Vol. 7, No. 4, pp. 951–965, November 1997

ALTERNATING PROJECTION-PROXIMAL METHODS FOR CONVEX PROGRAMMING AND VARIATIONAL INEQUALITIES[*]

PAUL TSENG[†]

Abstract. We consider a mixed problem composed in part of finding a zero of a maximal monotone operator and in part of solving a monotone variational inequality problem. We propose a solution method for this problem that alternates between a proximal step (for the maximal monotone operator part) and a projection-type step (for the monotone variational inequality part) and analyze its convergence and rate of convergence. This method extends a decomposition method of Chen and Teboulle [*Math. Programming*, 64 (1994), pp. 81–101] for convex programming and yields, as a by-product, new decomposition methods.

Key words. maximal monotone operator, monotone variational inequality, proximal point method, projection-type method, error bound, linear convergence

AMS subject classifications. 49J40, 49M45, 90C25, 90C33

PII. S1052623495279797

1. Introduction.

Since its proposal by Martinet and its comprehensive study by Rockafellar [24], [25], the proximal point method and its dual version in the context of convex programming, the method of multipliers, have received much study (see [1], [2], [11], [14] and references therein). One major direction of study has been in the development of decomposition methods for convex programming, as exemplified by the method of partial inverse [30], the alternating direction method of multipliers [6], [7], [15], and the alternating minimization algorithm [9], [12], [32]. Recently, Chen and Teboulle [5] proposed a new proximal-based decomposition method for solving convex programs with the separable structure:

$$\begin{aligned} &\text{minimize} \quad f_1(x_1) + f_2(x_2) \\ &\text{subject to} \quad Ax_1 - x_2 = 0, \end{aligned} \tag{1}$$

where f_1 and f_2 are closed proper convex functions on, respectively, $\Re^l$ and $\Re^m$ and $A \in \Re^{m\times l}$. In their method, proximal point iterations are applied to the subdifferential of the Lagrangian $L(x_1, x_2, y) = f_1(x_1) + f_2(x_2) + y^T(Ax_1 - x_2)$ alternately with the multipliers y fixed and with the variables (x_1, x_2) fixed. More specifically, the exact version of their method generates a sequence $\{(x_1^k, x_2^k, y^k, \hat{y}^k)\}_{k=0,1,...}$ according to the iteration (cf. [5, (2.9)–(2.12)]):

$$\hat{y}^k = y^k + \alpha_k(Ax_1^k - x_2^k), \tag{2}$$

$$x_1^{k+1} = \arg\min_{x_1\in\Re^l}\{f_1(x_1) + (\hat{y}^k)^T Ax_1 + \|x_1 - x_1^k\|^2/(2\alpha_k)\}, \tag{3}$$

$$x_2^{k+1} = \arg\min_{x_2\in\Re^m}\{f_2(x_2) - (\hat{y}^k)^T x_2 + \|x_2 - x_2^k\|^2/(2\alpha_k)\}, \tag{4}$$

$$y^{k+1} = y^k + \alpha_k(Ax_1^{k+1} - x_2^{k+1}) \tag{5}$$

for $k = 0, 1, ...$ The method has convergence properties similar to those of the proximal point method but with the additional requirement that α_k be less than

[*] Received by the editors January 11, 1995; accepted for publication (in revised form) July 2, 1996. This work was supported by National Science Foundation grant CCR-9311621.
http://www.siam.org/journals/siopt/7-4/27979.html

[†] Department of Mathematics, University of Washington, Seattle, WA 98195 (tseng@math.washington.edu).

$1/(2\max\{\|A\|,1\})$. More importantly, it has nice features not known in previous decomposition methods: its subproblem decomposes into two disjoint problems, one in x_1 and the other in x_2 (see (3), (4)), both of which have strongly convex objective function and decompose according to the separable structure of f_1 and f_2, respectively. Curiously, despite its striking proximal features, the method of Chen and Teboulle did not appear to fit into any of the known algorithmic frameworks such as the proximal point method or operator splitting methods.

Motivated by the work of Chen and Teboulle, in this paper we show that their method may be viewed naturally as an alternating version of the proximal point method and the extragradient method [13], [17]. Moreover, their method can be generalized to solve much broader classes of problems and to yield new decomposition methods for convex programming and variational inequalities. Specifically, we consider the general problem of finding a zero of a multivalued function $T : \Re^n \times \Re^m \rightrightarrows \Re^n \times \Re^m$ of the special form

$$T(x,y) = F(x,y) \times (G(x,y) + N_Y(y)), \tag{6}$$

where Y is a nonempty closed convex set in $\Re^m$ (N_Y denotes the normal cone mapping to Y); $F : \Re^n \times \Re^m \rightrightarrows \Re^n$ and $G : \Re^n \times \Re^m \rightrightarrows \Re^m$ are multivalued functions such that G is single valued and continuous on $\Re^n \times Y$, $(x,y) \mapsto F(x,y) \times G(x,y)$ is maximal monotone; and $\mathrm{ri}\{y : F(x,y) \times G(x,y) \neq \emptyset \text{ for some } x\} \supseteq Y$. (It follows from the results in [22] that T is maximal monotone.) We consider a general method whereby, for any initial $(x,y) \in \Re^n \times Y$ and any continuous function $G_1 : \Re^n \times Y \mapsto \Re^m$ such that $G_1(x,\cdot)$ is monotone for all x, we iteratively update (x,y) by solving

$$\hat{y} = [y - \alpha(G_1(x,\hat{y}) - G_1(x,y) + G(x,y))]_Y^+ \tag{7}$$

for some $\hat{y}$, and then solving

$$x - x^{new} \in \alpha F(x^{new}, \hat{y}) \tag{8}$$

for x^{new} and setting

$$y^{new} = [y - \alpha G(x^{new}, \hat{y})]_Y^+, \tag{9}$$

where α is a chosen positive stepsize and $[\cdot]_Y^+$ denotes the orthogonal projection onto Y. Thus, the idea of the method is to alternately apply one backward Euler step (i.e., proximal step) to the general multivalued part of T (namely F) and two forward–backward Euler steps (i.e., projection-type step) to the part of T with the variational inequality structure (namely $G + N_Y$). The method of Chen and Teboulle may be viewed as a special case of this method with $G_1 \equiv 0$ and applied to the case of T with

$$\begin{array}{rclrcl} x &=& (x_1,x_2), \qquad & F(x_1,x_2,y) &=& \partial_{x_1,x_2} L(x_1,x_2,y), \\ Y &=& \Re^m, \qquad & G(x_1,x_2,y) &=& -\partial_y L(x_1,x_2,y) \end{array}$$

(see the remark following Theorem 2.4 for detailed discussions).

A key advantage of this method over existing proximal-based methods [6], [9], [12], [15], [30], [32] is that it can readily exploit separable structures that occur, for example, in large-scale problems with a dynamic or a stochastic nature [28]. In particular, equations (7) and (9) (with G_1 suitably chosen) decompose according to the Cartesian product structure of Y, and the inclusion (8) decomposes according to the separable structure of F (e.g., $F(x,y) = F_1(x_1,y) \times \cdots \times F_n(x_n,y)$). To illustrate, consider the minimax problem

$$\min_{x\in\Re^n} \max_{y\in Y}\{f(x) - g(y) + y^T(Ax-b)\},$$

where f is a closed proper convex function on $\Re^n$, g is a continuously differentiable convex function on $\Re^m$, Y is a nonempty closed convex set in $\Re^m$, and $A \in \Re^{m\times n}$, $b \in \Re^m$. Suppose that f is separable, i.e., $f(x_1,...,x_n) = f_1(x_1) + \cdots + f_n(x_n)$ for some closed proper convex functions $f_1,...,f_n$ on $\Re$, and Y is a box (i.e., the Cartesian product of closed intervals). Problems possessing such a separable structure arise in discrete-time deterministic optimal control [4], [28] and in the scheduling of hydroelectric power generation under uncertainty [29], with x comprising certain state and control variables and $Ax = b$ modeling the linear dynamic linking the state and control variables. The minimax problem corresponds to $0 \in T(x,y)$ with

$$F(x,y) = \partial f(x) + A^T y, \qquad G(x,y) = b - Ax + \nabla g(y).$$

Applying (7)–(8) with $G_1 \equiv 0$ to this special case of T yields a method that iteratively updates (x,y) according to the equations:

$$\begin{aligned}\hat{y} &= [y - \alpha(b - Ax + \nabla g(y))]^+_Y,\\ x^{new} &= \arg\min_{\xi\in\Re^n}\{f(\xi) + \hat{y}^T A\xi + \|\xi - x\|^2/(2\alpha)\},\\ y^{new} &= [y - \alpha(b - Ax^{new} + \nabla g(\hat{y}))]^+_Y.\end{aligned}$$

Since f is separable, the computation of x^{new} decomposes into n independent convex programs in one variable with a strongly convex objective function. In fact, if $f_1,...,f_n$ are quadratic functions defined on closed intervals, then x^{new} is obtainable in closed form. Also, due to the product structure of Y, both $\hat{y}$ and y^{new} are obtainable in closed form. To our knowledge, existing proximal-based methods cannot decompose the computation of x^{new} at such a fine level, due to the presence of a quadratic term of the form $\|A\xi - b\|^2$ in the computation. In addition to $G_1 \equiv 0$, many other choices of G_1 are possible. If we had instead chosen $G_1 \equiv G$, then $\hat{y}$ would be computed according to

$$\hat{y} \;=\; \arg\min_{\psi\in Y}\left\{(b-Ax)^T\psi + g(\psi) + \|\psi - y\|^2/(2\alpha)\right\}.$$

This is a strongly convex program with box constraints, and if g is separable, then it decomposes into m independent convex programs in one variable. If we had instead chosen $G_1(x,y) = c - Ax + Qy$ for some symmetric positive semidefinite $Q \in \Re^{m\times m}$ and some $c \in \Re^m$, then $\hat{y}$ would be computed according to

$$\hat{y} \;=\; \arg\min_{\psi\in Y}\left\{(c-Ax)^T\psi + \psi^T Q\psi/2 + \|\psi - y\|^2/(2\alpha)\right\}.$$

This is a strongly convex quadratic program with box constraints, and if Q is a diagonal matrix, then $\hat{y}$ is obtainable in closed form. In general, choosing G_1 entails a tradeoff between the work in computing $\hat{y}$ and the speed of convergence. For further discussions of applications, see Examples 3–5.

Throughout, we denote by $\Re^n$ the space of n-dimensional real column vectors and by superscript T the transpose (of vectors or matrices). We denote by $\|x\|$ the 2-norm of a vector x (i.e., $\|x\| = \sqrt{x^T x}$) and, for any $A \in \Re^{m\times n}$, by $\|A\|$ the 2-norm of A induced by the vector 2-norm (i.e., $\|A\| = \max_{x:\|x\|=1} \|Ax\|$). We denote by I either the identity matrix or the identity mapping, and by linear convergence we mean R-linear convergence as defined in [19]. We say that a function $\phi : Z \mapsto \Re$, where $Z \subseteq \Re^n$, is *locally bounded* on Z if $\phi(z^k), k = 0, 1, ...$ is bounded for every convergent sequence $z^k \in Z$, $k = 0, 1....$

2. Algorithm description and convergence analysis. In this section we formally describe our method, based on alternating between two projection-type steps and one proximal step, for finding a zero of T of the form (6), and we present associated convergence and rate of convergence analysis.

Alternating projection-proximal (APP) method. *Choose any continuous function $G_1 : \Re^n \times Y \mapsto \Re^m$ such that $G_1(x, \cdot)$ is monotone for all $x \in \Re^n$ and choose any $(x^0, y^0) \in \Re^n \times Y$. For $k = 0, 1, ...,$ we generate (x^{k+1}, y^{k+1}) from (x^k, y^k) by choosing an $\alpha_k \in (0, \infty)$ and letting*

$$x^{k+1} = \xi^k(\alpha_k), \qquad y^{k+1} = \psi^k(\alpha_k), \tag{10}$$

where, for each $\alpha \in (0, \infty)$, $\hat{\psi}^k(\alpha)$ denotes the unique vector in $\Re^m$ satisfying

$$\hat{\psi}^k(\alpha) = [y^k - \alpha(G_1(x^k, \hat{\psi}^k(\alpha)) - G_1(x^k, y^k) + G(x^k, y^k))]^+_Y, \tag{11}$$

$\xi^k(\alpha)$ denotes the unique vector in $\Re^n$ satisfying

$$x^k - \xi^k(\alpha) \in \alpha F(\xi^k(\alpha), \hat{\psi}^k(\alpha)), \tag{12}$$

and $\psi^k(\alpha)$ is the vector in $\Re^m$ given by

$$\psi^k(\alpha) = [y^k - \alpha G(\xi^k(\alpha), \hat{\psi}^k(\alpha))]^+_Y. \tag{13}$$

It can be seen that $\hat{\psi}^k(\alpha)$ is the result of applying one iteration of the proximal point method (with stepsize α) at y^k to the maximal monotone operator

$$y \mapsto G_1(x^k, y) - G_1(x^k, y^k) + G(x^k, y^k) + N_Y(y), \tag{14}$$

so $\hat{\psi}^k(\alpha)$ is well defined and unique (see [24]). Similarly, $\xi^k(\alpha)$ is the result of applying one iteration of the proximal point method (with stepsize α) at x^k to the maximal monotone operator

$$x \mapsto F(x, \hat{\psi}^k(\alpha)), \tag{15}$$

so $\xi^k(\alpha)$ is well defined and unique. That the mapping (15) is maximal monotone follows from $\hat{\psi}^k(\alpha) \in Y \subseteq \mathrm{ri}\{y : F(x, y) \times G(x, y) \neq \emptyset \text{ for some } x\}$ and the following result suggested by Rockafellar [27].

Lemma 2.1. *Let $F : \Re^n \times \Re^m \rightrightarrows \Re^n$ and $G : \Re^n \times \Re^m \rightrightarrows \Re^m$ be multivalued functions such that $(x, y) \mapsto F(x, y) \times G(x, y)$ is maximal monotone. For any $\bar{y} \in \mathrm{ri}(\{y : F(x, y) \times G(x, y) \neq \emptyset$ for some $x\})$, the mapping $x \mapsto F(x, \bar{y})$ is maximal monotone.*

Proof. Let $M = \{(x, \bar{y}) : x \in \Re^n\}$ and let

$$T_1(x, y) = F(x, y) \times G(x, y) \quad \text{and} \quad T_2(x, y) = \begin{cases} M^\perp & \text{if } (x, y) \in M, \\ \emptyset & \text{else.} \end{cases}$$

Also, denote $D = \mathrm{dom}T_1$ and $C = \mathrm{cl}D$. Since T_1 is maximal monotone, by a result of Minty [18] (also see [3, Remark 2.1]), C is convex and $C \supseteq D \supseteq \mathrm{ri}C$. Thus,

$$\bar{y} \in \mathrm{ri}(\{y : (x, y) \in D \text{ for some } x\}) \subseteq \mathrm{ri}(\{y : (x, y) \in C \text{ for some } x\}),$$

so, for any $\bar{x} \in \mathrm{ri}(\{x : (x, \bar{y}) \in C\})$, applying [23, Theorem 6.8] yields that $(\bar{x}, \bar{y}) \in \mathrm{ri}C = \mathrm{ri}D$. Since $(\bar{x}, \bar{y}) \in M$ trivially, it follows that

$$\mathrm{ri}(\mathrm{dom}T_1) \cap \mathrm{ri}(\mathrm{dom}T_2) = \mathrm{ri}D \cap M \neq \emptyset.$$

Then, since both T_1 and T_2 are maximal monotone, Theorem 2 in [22] yields that $T_1 + T_2$ is maximal monotone or, equivalently, $x \mapsto F(x, \bar{y})$ is maximal monotone. □

The choice of the function G_1 (which affects the choice of $\hat{\psi}^k(\alpha)$ and the work in computing $\hat{\psi}^k(\alpha)$) and the stepsizes α_k, $k = 0, 1, ...$ (which affects the convergence of the method) are key to the performance of the APP method. A choice of G_1 that yields the least amount of work in computing $\hat{\psi}^k(\alpha)$ is

$$G_1 \equiv 0. \tag{16}$$

A choice of G_1 that requires more work in computing $\hat{\psi}^k(\alpha)$ is

$$G_1 \equiv G. \tag{17}$$

An intermediate choice is

$$G_1(x, y) = By \quad \forall y \in \Re^m, \tag{18}$$

where $B \in \Re^{m \times m}$ is positive semidefinite. If Y is a box, we can choose B to be either upper or lower triangular (such as the upper or lower triangular part of the Jacobian of G_1 with respect to y at (x^k, y^k), assuming G_1 is differentiable), in which case $\hat{\psi}^k(\alpha)$ may be computed via back-solve in the order of m^2 arithmetic operations. In general, one may need to experiment with a number of choices of G_1 before settling on one that is suitable for the intended application.

The choice of α_k is trickier for it cannot be too large (or the APP method might diverge) and it cannot be too small (or the convergence might be too slow). In certain applications, an estimate of a "reasonable" α_k may be obtained (see Theorem 2.4(b)). However, in practice, a form of Armijo–Goldstein line search (cf. [1], [10], [17]) would be more useful. Specifically, we will consider choosing α_k to be the largest $\alpha \in \{\sigma, \sigma\beta, \sigma\beta^2, ...\}$ such that $(\hat{\psi}^k(\alpha), \xi^k(\alpha), \psi^k(\alpha))$ given by (11)–(13) satisfies

(19)

$$\begin{aligned} 2\alpha\|\psi^k(\alpha) - \hat{\psi}^k(\alpha)\|\|G_1(x^k, \hat{\psi}^k(\alpha)) - G_1(x^k, y^k) + G(x^k, y^k) - G(\xi^k(\alpha), \hat{\psi}^k(\alpha))\| \\ \leq \quad (1 - \epsilon)(\|x^k - \xi^k(\alpha)\|^2 + \|y^k - \hat{\psi}^k(\alpha)\|^2 + \|\hat{\psi}^k(\alpha) - \psi^k(\alpha)\|^2), \end{aligned}$$

where β and ϵ are chosen scalars in $(0, 1)$ and σ is a chosen scalar in $(0, \infty)$. (A variant is to choose α_k (for $k > 0$) to be the largest $\alpha \in \{\alpha_{k-1}, \alpha_{k-1}\beta, \alpha_{k-1}\beta^2, ...\}$ satisfying (19). The resulting α_k, though more conservative, is cheaper to find since typically $\alpha = \alpha_{k-1}$ will satisfy (19). The convergence results below hold for this variant also.)

We will show that (19) is satisfied by all α sufficiently small, so α_k chosen in the above manner is well defined. To this end, we need the following lemma, stating a basic property of the proximal mapping (see [3, Proposition 2.6]), whose proof is included for completeness.

LEMMA 2.2. *Let S be any maximal monotone operator on $\Re^l$. For any $x \in \operatorname{dom} S$, we have*

$$\|x - (I + \alpha S)^{-1}(x)\|/\alpha \leq \min_{u \in S(x)} \|u\| \quad \forall \alpha > 0. \tag{20}$$

Proof. Fix any $x \in \operatorname{dom} S$ and, for each $\alpha > 0$, let $z_\alpha = (I + \alpha S)^{-1}(x)$. Then we have $(x - z_\alpha)/\alpha \in S(z_\alpha)$ so that, for any $u \in S(x)$,

$$\begin{aligned} \|x - z_\alpha\|^2/\alpha &= (x - z_\alpha)^T(x - z_\alpha)/\alpha \\ &= (x - z_\alpha)^T[(x - z_\alpha)/\alpha - u] + (x - z_\alpha)^T u \\ &\leq (x - z_\alpha)^T u \\ &\leq \|x - z_\alpha\|\|u\|, \end{aligned}$$

where the first inequality follows from S being monotone. Thus,

$$\|x - z_\alpha\| \leq \alpha\|u\| \quad \forall u \in S(x),$$

and (20) is proven. (The minimum in (20) is attained since $S(x)$ is a closed set.) □

The inequality in (20) is sharp as $\alpha \to 0$. To see this, note that, by (20), $z_\alpha = (I + \alpha S)^{-1}(x) \to x$ as $\alpha \to 0$, so it follows from $(x - z_\alpha)/\alpha \in S(z_\alpha)$ and the closed property of S [3, Proposition 2.5] that any cluster point of $(x - z_\alpha)/\alpha$ as $\alpha \to 0$ is in $S(x)$, implying

$$\liminf_{\alpha \to 0} \|x - (I + \alpha S)^{-1}(x)\|/\alpha \ \geq \ \min_{u \in S(x)} \|u\|. \tag{21}$$

By (20), the inequality in (21) holds with equality.

We will also need the following lemma stating some known properties of the projection mapping $[\cdot]_Y^+$. Parts (a), (b), (c) and (d) of this lemma are borrowed from, respectively, [34, Equation (1.8)], [34, Lemma 1.1], [10, Lemma 1], and [17, Appendix].

LEMMA 2.3. *For Y a nonempty closed convex set in $\Re^m$, the following hold.*

(a) *For any $u \in \Re^m$ and any $v \in \Re^m$, $\|[u]_Y^+ - [v]_Y^+\| \leq \|u - v\|$.*

(b) *For any $u \in \Re^m$, $z = [u]_Y^+$ satisfies $0 \leq (y - z)^T(z - u)$ for all $y \in Y$.*

(c) *For any $y \in Y$, any $d \in \Re^m$, and any $\alpha \in (0, 1]$, $\|y - [y - \alpha d]_Y^+\|/\alpha \geq \|y - [y - d]_Y^+\|$.*

(d) *For any $u \in \Re^m$ and any $v \in Y$, $\|[u]_Y^+ - v\|^2 \leq \|u - v\|^2 - \|u - [u]_Y^+\|^2$.*

Below we state and prove our main convergence result, showing that, under mild assumptions, the aforementioned Armijo–Goldstein line search rule (see (19)) is well defined and that the APP method using this stepsize rule is convergent. Moreover, if $Y = \Re^m$ and T^{-1} is locally upper Lipschitzian (see (24)), the method is linearly convergent. The convergence analysis is based on those for the proximal point method [24] and the extragradient method [17]. The rate of convergence analysis is based on that for certain projection-type methods for monotone variational inequalities [33]. A key to these analyses is a certain Féjer-convergence property of $\{(x^k, y^k)\}$; namely, the square of the Euclidean distance from (x^k, y^k) to any solution is monotonically decreasing with k and the amount of decrease is proportional to $\|(x^{k+1}, y^{k+1}) - (x^k, y^k)\|^2$ (see (23)).

THEOREM 2.4. *Consider a multivalued function $T : \Re^n \times \Re^m \rightrightarrows \Re^n \times \Re^m$ of the form (6), where Y is a nonempty closed convex set in $\Re^m$, $F : \Re^n \times \Re^m \rightrightarrows \Re^n$*

and $G : \Re^n \times \Re^m \rightrightarrows \Re^m$ are multivalued functions such that G is single valued and continuous on $\Re^n \times Y$, $(x, y) \mapsto F(x, y) \times G(x, y)$ is maximal monotone, and $\mathrm{ri}\{y :$ *$F(x, y) \times G(x, y) \neq \emptyset$ for some $x\} \supseteq Y$. Assume that, for every $x \in \Re^n$, the function $y \mapsto \min_{u \in F(x,y)} \|u\|$ is locally bounded on its domain $\{y : F(x, y) \neq \emptyset\}$. Denote $\Sigma = T^{-1}(0) = \{(x, y) : 0 \in T(x, y)\}$. Let G_1 and $\{(x^k, y^k)\}_{k=0,1,...}$ be generated by the APP method with α_k chosen to be the largest $\alpha \in \{\sigma, \sigma\beta, \sigma\beta^2, ...\}$ such that $(\hat{\psi}^k(\alpha), \xi^k(\alpha), \psi^k(\alpha))$ given by* (11)–(13) *satisfies* (19), *where $\beta \in (0, 1)$, $\epsilon \in (0, 1)$ and $\sigma \in (0, \infty)$. Then the following hold.*

(a) *α_k is well defined for all k.*

(b) *If G_1 and G are Lipschitz continuous on $\Re^n \times Y$ (with constants $L_1 \geq 0$ and $L \geq 0$, respectively), then $\{\alpha_k\}$ is bounded below by a positive scalar and, in particular,*

$$(22) \quad \alpha_k \geq \begin{cases} \sigma & \text{if } \sigma \leq (1-\epsilon)/\sqrt{(L_1 + L)^2 + L^2}, \\ \beta(1-\epsilon)/\sqrt{(L_1 + L)^2 + L^2} & \text{otherwise.} \end{cases}$$

(c) *If Σ is nonempty, then for any $(x^*, y^*) \in \Sigma$ and any $k \in \{0, 1, ...\}$ we have*

$$(23) \quad \begin{aligned} & \|x^{k+1} - x^*\|^2 + \|y^{k+1} - y^*\|^2 \\ \leq\ & \|x^k - x^*\|^2 + \|y^k - y^*\|^2 - \epsilon(\|x^k - x^{k+1}\|^2 + \|y^k - \hat{y}^k\|^2 + \|\hat{y}^k - y^{k+1}\|^2). \end{aligned}$$

If in addition either (i) *$\{\alpha_k\}$ is bounded below by a positive scalar or* (ii) *the function $(x, y) \mapsto \min_{u \in F(x,y)} \|u\|$ is locally bounded on its domain $\{(x, y) : F(x, y) \neq \emptyset\}$, then $\{(x^k, y^k)\}$ converges to an element of Σ.*

(d) *If Σ is nonempty, $Y = \Re^m$ and there exists $\tau > 0$ and $\delta > 0$ such that*

$$(24) \qquad T^{-1}(u) \subseteq T^{-1}(0) + \tau\|u\|B \quad \forall u \text{ with } \|u\| \leq \delta,$$

where $B = \{x : \|x\| \leq 1\}$, and $\{\alpha_k\}$ is bounded below by a positive scalar, then $d((x^k, y^k), \Sigma) \to 0$ linearly as $k \to \infty$, where we denote $d(z, \Sigma) = \min_{z^* \in \Sigma} \|z - z^*\|$.

Proof. (a) Fix any $k \in \{0, 1, ...\}$. To see that α_k is well defined, note that if $(x^k, y^k) \in \Sigma$, then $\alpha_k = \sigma$ (since both sides of (19) equal zero for any $\alpha > 0$), so it suffices to assume that $(x^k, y^k) \notin \Sigma$, i.e., $\min_{u \in F(x^k, y^k)} \|u\|^2 + \|y^k - [y^k - G(x^k, y^k)]_Y^+\|^2 > 0$. We have from applying Lemma 2.2 with S being the maximal monotone operator (14) that $\hat{\psi}^k(\alpha) \to y^k$ as $\alpha \to 0$. We also have from applying Lemma 2.2 with S being the maximal monotone operator (15) that

$$\|x^k - \xi^k(\alpha)\|/\alpha \leq \min_{u \in F(x^k, \hat{\psi}^k(\alpha))} \|u\|,$$

so the assumption that $y \mapsto \min_{u \in F(x,y)} \|u\|$ is locally bounded on its domain yields

$$\lim_{\alpha \to 0} \sup \|x^k - \xi^k(\alpha)\|/\alpha < \infty,$$

implying $\xi^k(\alpha) \to x^k$ as $\alpha \to 0$. We also have from (11) and (13) and Lemma 2.3(a) that

$$\begin{aligned} & \|\psi^k(\alpha) - \hat{\psi}^k(\alpha)\|/\alpha \\ & = \|[y^k - \alpha G(\xi^k(\alpha), \hat{\psi}^k(\alpha))]_Y^+ \\ & - [y^k - \alpha(G_1(x^k, \hat{\psi}^k(\alpha)) - G_1(x^k, y^k) + G(x^k, y^k))]_Y^+\|/\alpha \\ & \leq \| - G(\xi^k(\alpha), \hat{\psi}^k(\alpha)) + G_1(x^k, \hat{\psi}^k(\alpha)) - G_1(x^k, y^k) + G(x^k, y^k)\|, \end{aligned}$$

so, since G_1 and G are continuous on $\Re^n \times Y$, we see that the right-hand side tends to zero as $\alpha \to 0$. Thus, the left-hand side of (19) divided by α^2 tends to zero as $\alpha \to 0$. On the other hand, we have from (13) and Lemma 2.3(c) that, for all $\alpha \in (0,1]$,

$$\begin{aligned}\|y^k - \hat{\psi}^k(\alpha)\|/\alpha &= \|y^k - [y^k - \alpha(G_1(x^k, \hat{\psi}^k(\alpha)) - G_1(x^k, y^k) + G(x^k, y^k))]_Y^+\|/\alpha \\ &\geq \|y^k - [y^k - (G_1(x^k, \hat{\psi}^k(\alpha)) - G_1(x^k, y^k) + G(x^k, y^k))]_Y^+\| \\ &\to \|y^k - [y^k - G(x^k, y^k)]_Y^+\| \quad \text{as} \quad \alpha \to 0.\end{aligned}$$

We also have from (12), the fact $(\xi^k(\alpha), \hat{\psi}^k(\alpha)) \to (x^k, y^k)$ as $\alpha \to 0$, and the closed property of F [3, Proposition 2.5] that any cluster point of $(x^k - \xi^k(\alpha))/\alpha$ as $\alpha \to 0$ is in $F(x^k, y^k)$, yielding (cf. (21))

$$\liminf_{\alpha \to 0} \|x^k - \xi^k(\alpha)\|/\alpha \;\geq\; \min_{u \in F(x^k, y^k)} \|u\|.$$

So we have $\lim_{\alpha \to 0} \inf\{\|x^k - \xi^k(\alpha)\|^2/\alpha^2 + \|y^k - \hat{\psi}^k(\alpha)\|^2/\alpha^2\} > 0$. This implies that the right-hand side of (19) divided by α^2 does not tend to zero as $\alpha \to 0$. Hence, (19) holds whenever α is sufficiently small, implying α_k is well defined.

(b) Suppose that G_1 and G are Lipschitz continuous on $\Re^n \times Y$ with Lipschitz constant $L_1 \geq 0$ and $L \geq 0$, respectively. Then, for any $k \in \{0, 1, ...\}$, we have for each $\alpha \in (0, \infty)$ that the left-hand side of (19) is bounded above by

$$\begin{aligned}&2\alpha\|\psi^k(\alpha) - \hat{\psi}^k(\alpha)\|[L_1\|\hat{\psi}^k(\alpha) - y^k\| + L\|(x^k, y^k) - (\xi^k(\alpha), \hat{\psi}^k(\alpha))\|] \\ &\leq 2\alpha\|\psi^k(\alpha) - \hat{\psi}^k(\alpha)\|[(L_1 + L)\|\hat{\psi}^k(\alpha) - y^k\| + L\|x^k - \xi^k(\alpha)\|] \\ &\leq \alpha\sqrt{(L_1 + L)^2 + L^2}[\|\psi^k(\alpha) - \hat{\psi}^k(\alpha)\|^2 + \|\hat{\psi}^k(\alpha) - y^k\|^2 + \|x^k - \xi^k(\alpha)\|^2],\end{aligned}$$

where the second inequality uses the inequality $2a[\mu b + \lambda c] \leq \sqrt{\mu^2 + \lambda^2}[a^2 + b^2 + c^2]$. Thus, (19) holds whenever the right-hand side of the above inequality is below the right-hand side of (19), which in turn holds whenever $\alpha \leq (1 - \epsilon)/\sqrt{(L_1 + L)^2 + L^2}$. Since α_k is the largest $\alpha \in \{\sigma, \sigma\beta, ...\}$ for which (19) holds, it follows that (22) holds.

(c) Fix any $(x^*, y^*) \in \Sigma$ and any $k \in \{0, 1, ...\}$. Let $\hat{y}^k = \hat{\psi}^k(\alpha_k)$, so that, by (10) and (12)–(13) with $\alpha = \alpha_k$,

$$x^k - x^{k+1} \in \alpha_k F(x^{k+1}, \hat{y}^k), \tag{25}$$

$$y^{k+1} = [y^k - \alpha_k G(x^{k+1}, \hat{y}^k)]_Y^+. \tag{26}$$

We have from (11) with $\alpha = \alpha_k$ and Lemma 2.3(b) that

$$0 \leq (y - \hat{y}^k)^T(\alpha_k(G_1(x^k, \hat{y}^k) - G_1(x^k, y^k) + G(x^k, y^k)) + \hat{y}^k - y^k) \quad \forall y \in Y.$$

Similarly, since $(x^*, y^*) \in \Sigma$ so that $y^* = [y^* - \alpha_k G(x^*, y^*)]_Y^+$, we have from Lemma 2.3(b) that

$$0 \leq \alpha_k (y - y^*)^T G(x^*, y^*) \quad \forall y \in Y.$$

Taking $y = y^{k+1}$ in the first inequality and $y = \hat{y}^k$ in the second inequality, we obtain, respectively,

$$\begin{aligned}0 &\leq (y^{k+1} - \hat{y}^k)^T(\alpha_k(G_1(x^k, \hat{y}^k) - G_1(x^k, y^k) + G(x^k, y^k)) + \hat{y}^k - y^k) \\ &= \alpha_k(y^{k+1} - \hat{y}^k)^T(G_1(x^k, \hat{y}^k) - G_1(x^k, y^k) + G(x^k, y^k) - G(x^{k+1}, \hat{y}^k)) \\ &\quad + \alpha_k(y^{k+1} - \hat{y}^k)^T G(x^{k+1}, \hat{y}^k) + (y^{k+1} - \hat{y}^k)^T(\hat{y}^k - y^k),\end{aligned}$$

and

$$0 \le \alpha_k(\hat{y}^k - y^*)^T G(x^*, y^*).$$

Also, we have trivially that

$$\|x^{k+1} - x^*\|^2 = \|x^k - x^*\|^2 - \|x^k - x^{k+1}\|^2 + 2(x^{k+1} - x^k)^T(x^{k+1} - x^*)$$

and from (26) and Lemma 2.3(d) that

$$\begin{aligned}\|y^{k+1} - y^*\|^2 &= \|[y^k - \alpha_k G(x^{k+1}, \hat{y}^k)]_Y^+ - y^*\|^2 \\ &\le \|y^k - \alpha_k G(x^{k+1}, \hat{y}^k) - y^*\|^2 - \|y^k - \alpha_k G(x^{k+1}, \hat{y}^k) - y^{k+1}\|^2 \\ &= \|y^k - y^*\|^2 - \|y^k - y^{k+1}\|^2 + 2\alpha_k(y^* - y^{k+1})^T G(x^{k+1}, \hat{y}^k) \\ &= \|y^k - y^*\|^2 - \|y^k - y^{k+1}\|^2 + 2\alpha_k(y^* - \hat{y}^k)^T G(x^{k+1}, \hat{y}^k) \\ &+ 2\alpha_k(\hat{y}^k - y^{k+1})^T G(x^{k+1}, \hat{y}^k).\end{aligned}$$

Multiplying the first two of the previous four inequalities by 2 and adding them to the last two of the previous four inequalities yields

$$\begin{aligned}&\|x^{k+1} - x^*\|^2 + \|y^{k+1} - y^*\|^2 \\ &\le \|x^k - x^*\|^2 + \|y^k - y^*\|^2 - \|x^k - x^{k+1}\|^2 - \|y^k - y^{k+1}\|^2 \\ &+ 2(x^{k+1} - x^k)^T(x^{k+1} - x^*) + 2\alpha_k(y^* - \hat{y}^k)^T(G(x^{k+1}, \hat{y}^k) - G(x^*, y^*)) \\ &+ 2\alpha_k(y^{k+1} - \hat{y}^k)^T(G_1(x^k, \hat{y}^k) - G_1(x^k, y^k) + G(x^k, y^k) - G(x^{k+1}, \hat{y}^k)) \\ &+ 2(y^{k+1} - \hat{y}^k)^T(\hat{y}^k - y^k) \\ &\le \|x^k - x^*\|^2 + \|y^k - y^*\|^2 - \|x^k - x^{k+1}\|^2 - \|y^k - y^{k+1}\|^2 \\ &+ 2\alpha_k(y^{k+1} - \hat{y}^k)^T(G_1(x^k, \hat{y}^k) - G_1(x^k, y^k) + G(x^k, y^k) - G(x^{k+1}, \hat{y}^k)) \\ &+ 2(y^{k+1} - \hat{y}^k)^T(\hat{y}^k - y^k) \\ &= \|x^k - x^*\|^2 + \|y^k - y^*\|^2 - \|x^k - x^{k+1}\|^2 - \|y^k - \hat{y}^k\|^2 - \|\hat{y}^k - y^{k+1}\|^2 \\ &+ 2\alpha_k(y^{k+1} - \hat{y}^k)^T(G_1(x^k, \hat{y}^k) - G_1(x^k, y^k) + G(x^k, y^k) - G(x^{k+1}, \hat{y}^k)) \\ &\le \|x^k - x^*\|^2 + \|y^k - y^*\|^2 - \|x^k - x^{k+1}\|^2 - \|y^k - \hat{y}^k\|^2 - \|\hat{y}^k - y^{k+1}\|^2 \\ &+ 2\alpha_k\|y^{k+1} - \hat{y}^k\|\|G_1(x^k, \hat{y}^k) - G_1(x^k, y^k) + G(x^k, y^k) - G(x^{k+1}, \hat{y}^k)\|,\end{aligned}$$

where the second inequality follows from using (25) and $0 \in \alpha_k F(x^*, y^*)$ (since $(x^*, y^*) \in \Sigma$) as well as the monotone property of $F \times G$. This, together with (10) and (19) with $\alpha = \alpha_k$, yields (23).

Since (23) holds for $k = 0, 1, ...$ and any $(x^*, y^*) \in \Sigma$, the sequence $\{(x^k, y^k)\}_{k=0,1,...}$ is bounded and hence contains a subsequence $\{(x^k, y^k)\}_{k\in K}$, where $K \subseteq \{0, 1, 2, ...\}$, converging to some limit point (x^∞, y^∞). If $(x^\infty, y^\infty) \in \Sigma$, then, by letting $(x^*, y^*) = (x^\infty, y^\infty)$ in (23), we would obtain that $\{\|(x^k, y^k) - (x^\infty, y^\infty)\|\}$ is monotonically decreasing and contains a subsequence tending to zero, so the entire sequence must converge to zero. Below, we show that $(x^\infty, y^\infty) \in \Sigma$ if either (i) $\{\alpha_k\}$ is bounded below by a positive scalar or (ii) the function $(x, y) \mapsto \min_{u\in F(x,y)} \|u\|$ is locally bounded on its domain $\{(x, y) : F(x, y) \neq \emptyset\}$. In case (i), since (see (23)) $\{\|x^k - x^{k+1}\|^2 + \|y^k - \hat{y}^k\|^2 + \|\hat{y}^k - y^{k+1}\|^2\} \to 0$ and $F \times G$ is a closed mapping [3, Proposition 2.5], we would then have from (25)–(26) that

$$0 \in \alpha_\infty F(x^\infty, y^\infty), \qquad y^\infty = [y^\infty - \alpha_\infty G(x^\infty, y^\infty)]_Y^+,$$

where α_∞ denotes any cluster point of $\{\alpha_k\}_{k\in K}$. This implies $(x^\infty, y^\infty) \in \Sigma$. In case (ii), if $\{\alpha_k\}_{k\in K}$ contains a subsequence that is bounded below by a positive scalar, then an argument analogous to that used in case (i) would yield $(x^\infty, y^\infty) \in \Sigma$. Otherwise, $\{\alpha_k\}_{k\in K} \to 0$ and we will argue that $(x^\infty, y^\infty) \in \Sigma$ by contradiction. Suppose $(x^\infty, y^\infty) \notin \Sigma$, i.e., $\min_{u\in F(x^\infty,y^\infty)} \|u\|^2 + \|y^\infty - [y^\infty - G(x^\infty, y^\infty)]_Y^+\|^2 > 0$. Since $\{\alpha_k\}_{k\in K} \to 0$, then for all $k \in K$ sufficiently large we have $\alpha_k < \sigma$, so our choice of α_k implies $(\hat\psi^k(\alpha), \xi^k(\alpha), \psi^k(\alpha))$ given by (11)–(13) does not satisfy (19) for $\alpha = \bar\alpha_k$, where $\bar\alpha_k = \alpha_k/\beta$, i.e.,

(27)

$$\begin{aligned}&2\bar\alpha_k\|\psi^k(\bar\alpha_k) - \hat\psi^k(\bar\alpha_k)\|\|G_1(x^k, \hat\psi^k(\bar\alpha_k)) - G_1(x^k, y^k) + G(x^k, y^k) - G(\xi^k(\bar\alpha_k), \hat\psi^k(\bar\alpha_k))\|\\ &\qquad > \quad (1-\epsilon)(\|x^k - \xi^k(\bar\alpha_k)\|^2 + \|y^k - \hat\psi^k(\bar\alpha_k)\|^2 + \|\hat\psi^k(\bar\alpha_k) - \psi^k(\bar\alpha_k)\|^2).\end{aligned}$$

Applying Lemma 2.2 with S being the maximal monotone operator (14) with $\alpha = \bar\alpha_k$, we obtain

$$\|y^k - \hat\psi^k(\bar\alpha_k)\|/\bar\alpha_k \;\le\; \min_{u\in G(x^k,y^k)+N_Y(y^k)} \|u\| \;\le\; \|G(x^k, y^k)\|,$$

where the second inequality follows from the fact that $0 \in N_Y(y^k)$. Since $\{\bar\alpha_k\}_{k\in K} \to 0$ and $\{(x^k, y^k)\}_{k\in K} \to (x^\infty, y^\infty)$ and G is continuous on $\Re^n \times Y$, this implies

$$\{\hat\psi^k(\bar\alpha_k)\}_{k\in K} \to y^\infty.$$

We also have from applying Lemma 2.2 with S being the maximal monotone operator (15) that

$$\|x^k - \xi^k(\bar\alpha_k)\|/\bar\alpha_k \;\le\; \min_{u\in F(x^k,\hat\psi^k(\bar\alpha_k))} \|u\|$$

so the assumption that $(x, y) \mapsto \min_{u\in F(x,y)} \|u\|$ is locally bounded on its domain yields

$$\limsup_{k\to\infty, k\in K} \|x^k - \xi^k(\bar\alpha_k)\|/\bar\alpha_k < \infty,$$

implying $\{\xi^k(\bar\alpha_k)\}_{k\in K} \to x^\infty$. We have from (11) and (13) and Lemma 2.3(a) that

$$\begin{aligned}&\|\psi^k(\bar\alpha_k) - \hat\psi^k(\bar\alpha_k)\|/\bar\alpha_k\\ &= \|[y^k - \bar\alpha_k G(\xi^k(\bar\alpha_k), \hat\psi^k(\bar\alpha_k))]_Y^+\\ &- [y^k - \bar\alpha_k(G_1(x^k, \hat\psi^k(\bar\alpha_k)) - G_1(x^k, y^k) + G(x^k, y^k))]_Y^+\|/\bar\alpha_k\\ &\le \| - G(\xi^k(\bar\alpha_k), \hat\psi^k(\bar\alpha_k)) + G_1(x^k, \hat\psi^k(\bar\alpha_k)) - G_1(x^k, y^k) + G(x^k, y^k)\|,\end{aligned}$$

so, since G_1 and G are continuous on $\Re^n \times Y$, we see that the right-hand side tends to zero as $k \to \infty, k \in K$. Thus, the left-hand side of (27) divided by $\bar\alpha_k^2$ tends to zero as $k \to \infty, k \in K$. On the other hand, we have from (11) and Lemma 2.3(c) that, for all $k \in K$ sufficiently large so that $\bar\alpha_k \in (0, 1]$,

$$\begin{aligned}&\|y^k - \hat\psi^k(\bar\alpha_k)\|/\bar\alpha_k\\ &= \|y^k - [y^k - \bar\alpha_k(G_1(x^k, \hat\psi^k(\bar\alpha_k)) - G_1(x^k, y^k) + G(x^k, y^k))]_Y^+\|/\bar\alpha_k\\ &\ge \|y^k - [y^k - (G_1(x^k, \hat\psi^k(\bar\alpha_k)) - G_1(x^k, y^k) + G(x^k, y^k))]_Y^+\|\\ &\to \|y^\infty - [y^\infty - G(x^\infty, y^\infty)]_Y^+\| \quad \text{as} \quad k \to \infty, k \in K.\end{aligned}$$

We also have from (12), the fact $\{(\xi^k(\bar{\alpha}_k), \hat{\psi}^k(\bar{\alpha}_k))\}_{k\in K} \to (x^\infty, y^\infty)$, and the closed property of F [3, Proposition 2.5] that any cluster point of $\{(x^k - \xi^k(\bar{\alpha}_k))/\bar{\alpha}_k\}_{k\in K}$ is in $F(x^\infty, y^\infty)$, yielding

$$\lim_{k\to\infty, k\in K} \inf \|x^k - \xi^k(\bar{\alpha}_k)\|/\bar{\alpha}_k \ \geq \ \min_{u\in F(x^\infty, y^\infty)} \|u\|.$$

So we have $\lim_{k\to\infty,k\in K} \inf\{\|x^k - \xi^k(\bar{\alpha}_k)\|^2/(\bar{\alpha}_k)^2 + \|y^k - \hat{\psi}^k(\bar{\alpha}_k)\|^2/(\bar{\alpha}_k)^2\} > 0$, implying the right-hand side of (27) divided by $(\bar{\alpha}_k)^2$ does not tend to zero as $k \to \infty, k \in K$, a contradiction of (27) holding for all $k \in K$ sufficiently large.

(d) Suppose that $Y = \Re^m$ and there exist $\tau > 0$ and $\delta > 0$ such that (24) holds. It can be seen that (24) is equivalent to

$$(28) \qquad d(z, \Sigma) \leq \tau \min_{u\in T(z)} \|u\| \quad \forall z \in \text{dom } T \text{ with } \min_{u\in T(z)} \|u\| \leq \delta.$$

(The minimum is attained by the closed property of T [3, Proposition 2.5].) Also, since $Y = \Re^m$, we have from (10) and (12)–(13) with $\alpha = \alpha_k$ that

$$(x^k - x^{k+1}, y^k - y^{k+1})/\alpha_k \in (F(x^{k+1}, \hat{y}^k), G(x^{k+1}, \hat{y}^k)) = T(x^{k+1}, \hat{y}^k)$$

for all k, where we denote $\hat{y}^k = \psi^k(\alpha_k)$. Since $\{\alpha_k\}$ is bounded away from zero and (see (23)) $\{(x^k - x^{k+1}, y^k - y^{k+1})\} \to 0$, we have that the norm of the above left-hand side is below δ for all k sufficiently large, in which case (28) yields

$$d((x^{k+1}, \hat{y}^k), \Sigma) \leq \tau \|(x^k - x^{k+1}, y^k - y^{k+1})\|/\alpha_k,$$

implying

$$\begin{aligned}
&d((x^{k+1}, y^{k+1}), \Sigma)^2 \\
&\leq [d((x^{k+1}, \hat{y}^k), \Sigma) + \|\hat{y}^k - y^{k+1}\|]^2 \\
&\leq [\tau\|(x^k - x^{k+1}, y^k - y^{k+1})\|/\alpha_k + \|\hat{y}^k - y^{k+1}\|]^2 \\
&\leq 2[\tau\|(x^k - x^{k+1}, y^k - y^{k+1})\|/\alpha_k]^2 + 2\|\hat{y}^k - y^{k+1}\|^2 \\
&= 2(\tau/\alpha_k)^2\|x^k - x^{k+1}\|^2 + 2(\tau/\alpha_k)^2\|y^k - y^{k+1}\|^2 + 2\|\hat{y}^k - y^{k+1}\|^2 \\
&\leq 2(\tau/\alpha_k)^2\|x^k - x^{k+1}\|^2 + 4(\tau/\alpha_k)^2\|y^k - \hat{y}^k\|^2 + [4(\tau/\alpha_k)^2 + 2]\|\hat{y}^k - y^{k+1}\|^2,
\end{aligned}$$

where the last two inequalities use the identity $(a + b)^2 \leq 2a^2 + 2b^2$. Since (23) holds for any k and any $(x^*, y^*) \in \Sigma$, by letting (x^*, y^*) be the element of Σ nearest to (x^k, y^k) in Euclidean norm, we obtain from (23) and the above inequality that

$$\begin{aligned}
&d((x^{k+1}, y^{k+1}), \Sigma)^2 \\
&\leq \|x^{k+1} - x^*\|^2 + \|y^{k+1} - y^*\|^2 \\
&\leq \|x^k - x^*\|^2 + \|y^k - y^*\|^2 \\
&\quad - \epsilon(\|x^k - x^{k+1}\|^2 + \|y^k - \hat{y}^k\|^2 + \|\hat{y}^k - y^{k+1}\|^2) \\
&= d((x^k, y^k), \Sigma)^2 - \epsilon(\|x^k - x^{k+1}\|^2 + \|y^k - \hat{y}^k\|^2 + \|\hat{y}^k - y^{k+1}\|^2) \\
&\leq d((x^k, y^k), \Sigma)^2 - \epsilon d((x^{k+1}, y^{k+1}), \Sigma)^2/[4(\tau/\alpha_k)^2 + 2]
\end{aligned}$$

and hence

$$[\epsilon/[4(\tau/\alpha_k)^2 + 2] + 1]d((x^{k+1}, y^{k+1}), \Sigma)^2 \leq d((x^k, y^k), \Sigma)^2.$$

This holds for all k sufficiently large and, since $\{\alpha_k\}$ is bounded away from zero, it follows that $d((x^k, y^k), \Sigma) \to 0$ linearly as $k \to \infty$. $\square$

The assumptions on F and G in Theorem 2.4 are quite mild and, in particular, the assumption that $y \mapsto \min_{u \in F(x,y)} \|u\|$ is locally bounded on its domain is satisfied for all our applications (see Examples 1–5) in which F has the special form

$$F(x,y) = F_1(x) + F_2(y)$$

with $F_1 : \Re^n \rightrightarrows \Re^n$ and $F_2 : \Re^m \rightrightarrows \Re^n$ lower semicontinuous (in fact, affine) on Y. If in addition $F_1 = \Phi_1 + N_X$, with $\Phi_1 : \Re^n \rightrightarrows \Re^n$ maximal monotone and X a nonempty closed convex subset of int(dom Φ_1), then $(x,y) \mapsto \min_{u \in F(x,y)} \|u\|$ is locally bounded on its domain (since Φ_1 is locally bounded on int(dom Φ_1) [3, Proposition 2.9], $0 \in N_X(x)$ for all $x \in X$, and F_2 is lower semicontinuous on Y).

We remark that part (d) of Theorem 2.4 still holds if G_1 and G are assumed to be Lipschitz continuous only on $(\Re^n \times Y) \cap (\Sigma + \delta' B)$ for some $\delta' > 0$. (This is because, by part (c), (x^k, y^k) is in this set for all k sufficiently large.) Also, we note that the assumption of part (d) (see (24)) is weaker than [5, Assumption B] since it does not assume in addition $T^{-1}(0)$ is a singleton. (The assumption that $T^{-1}(0)$ is a singleton precludes the possibility of multiple solutions.) The locally upper Lipschitzian property of T^{-1} (as embodied by (24)) and its equivalent formulation as a local error bound (see (28)) are discussed in [16], [20], [21]. It is an open question whether the results of part (d) can be extended to the case where $Y \neq \Re^m$. Finally, as in [5], [6], [24], Theorem 2.4 can be extended to hold for an inexact version of the APP method and to a Hilbert space setting, but for simplicity we do not consider this more general case.

As we noted in the introduction, the exact version of the decomposition method of Chen and Teboulle (2)–(5) may be viewed as a special case of the APP method with $G_1 \equiv 0$ and applied to the special case of T with

$$F(x_1, x_2, y) = \begin{bmatrix} \partial f_1(x_1) + A^T y \\ \partial f_2(x_2) - y \end{bmatrix}, \quad G(x_1, x_2, y) = x_2 - Ax_1, \quad Y = \Re^m.$$

Noting that G_1 and G are Lipschitz continuous on $\Re^{2n} \times Y$ with Lipschitz constants of $L_1 = 0$ and $L = \|[-A\ I]\| = \sqrt{1 + \|A\|^2} \leq \sqrt{2}\max\{\|A\|, 1\}$, respectively, we see that the upper bound of $(1-\epsilon)/(2\max\{\|A\|, 1\})$ on α_k, as specified in [5], is less than or equal to the constant $(1-\epsilon)/\sqrt{(L_1 + L)^2 + L^2}$ specified in (22). Thus, the stepsize choice in [5] corresponds to a conservative version of the Armijo–Goldstein stepsize choice considered in Theorem 2.4.

The APP method is also closely related to many well-known methods for variational inequality and for finding zero of a maximal monotone operator.

Example 1. In the special case where $m = 0$ (i.e., no y term), the APP method reduces to the proximal point method for solving $0 \in F(x)$ [11], [14], [24], [25]:

$$x^{k+1} = (I + \alpha_k F)^{-1}(x^k),$$

for $k = 0, 1, ...$

Example 2. In the special case where $n = 0$ (i.e., no x term), the APP method reduces to a function-splitting method for solving $0 \in G(y) + N_Y(y)$:

$$\begin{aligned} \hat{y}^k &= [y^k - \alpha_k(G_1(\hat{y}^k) - G_1(y^k) + G(y^k))]^+_Y, \\ y^{k+1} &= [y^k - \alpha_k G_1(\hat{y}^k)]^+_Y \end{aligned}$$

for $k = 0, 1, \ldots.$ Moreover, in the case G_1 is chosen according to (16), (17), or (18), this method reduces to, respectively, the extragradient method, the proximal point method, and a certain matrix-splitting method (see [16] and [33] for discussions of related methods).

In addition to the decomposition method of Chen and Teboulle, we can also derive new decomposition methods by applying the APP method appropriately.

Example 3. Consider the special case of T with

$$x = (x_1, x_2),\ F(x_1, x_2, y) = \begin{bmatrix} T_1(x_1) + A^T y \\ T_2(x_2) + B^T y \end{bmatrix},\ G(x_1, x_2, y) = b - Ax_1 - Bx_2,\ Y = \Re^m,$$

where T_1 and T_2 are maximal monotone operators on $\Re^{n_1}$ and $\Re^{n_2}$, respectively, and $A \in \Re^{m\times n_1}$, $B \in \Re^{m\times n_2}$, $b \in \Re^m$. The special case where $T_1 = \partial f_1$, $T_2 = \partial f_2$, $B = -I$, and $b = 0$ yields the convex program (1). The special case where $n_1 = n_2$, $A = -B = I$, and $b = 0$ yields the problem of finding a zero of $T_1 + T_2$. Applying the APP method with, say, $G_1 \equiv 0$ to this special case of T yields the new splitting method:

$$\begin{aligned} \hat{y}^k &= y^k + \alpha_k(Ax_1^k + Bx_2^k - b), \\ x_1^{k+1} &= (I + \alpha_k T_1)^{-1}(x^k - \alpha_k A^T \hat{y}^k), \\ x_2^{k+1} &= (I + \alpha_k T_2)^{-1}(x^k - \alpha_k B^T \hat{y}^k), \\ y^{k+1} &= y^k + \alpha_k(Ax_1^{k+1} + Bx_2^{k+1} - b) \end{aligned}$$

for $k = 0, 1, \ldots.$ In contrast to the Douglas–Rachford method (see [6], [15]), this method takes backward steps for T_1 and T_2 simultaneously (rather than serially) and, in contrast to the forward–backward method (see [4], [9], [32]), this method does not require either T_1 or T_2 or their inverse to be single valued and strongly monotone.

Example 4. Consider the minimax problem discussed in section 1:

$$\min_{x\in\Re^n} \max_{y\in Y} \{f(x) - g(y) + y^T Ax\},$$

where f is a closed proper convex function on $\Re^n$, g is a continuously differentiable convex function on $\Re^m$, Y is a nonempty closed convex set in $\Re^m$, and $A \in \Re^{m\times n}$ (assuming, without loss of generality, that $b = 0$). This problem corresponds to $0 \in T(x, y)$ with

$$F(x, y) = \partial f(x) + A^T y, \qquad G(x, y) = \nabla g(y) - Ax$$

and applying the APP method with, say, $G_1 \equiv 0$ to this special case of T yields the new method:

$$\begin{aligned} \hat{y}^k &= [y^k + \alpha_k(Ax^k - \nabla g(y^k))]_Y^+, \\ x^{k+1} &= \arg\min_{x\in\Re^n} \{f(x) + (\hat{y}^k)^T Ax + \|x - x^k\|^2/(2\alpha_k)\}, \\ y^{k+1} &= [y^k + \alpha_k(Ax^{k+1} - \nabla g(\hat{y}^k))]_Y^+ \end{aligned}$$

for $k = 0, 1, \ldots.$ In cases where Y has a Cartesian product structure or f has a separable structure (as in certain discrete time deterministic optimal control problems [28]), the above computation further decomposes. In contrast to previous decomposition methods for the extended linear-quadratic programming problem [26], [31], [35], the above method does not require f and g to be strongly convex or to be convex quadratic on some closed convex set.

Example 5. Consider the problem studied in [8] of minimizing $f(x) + g(x)$ over $x \in \Re^n$, where f is a closed proper convex function on $\Re^n$ and g is a continuously differentiable convex function on $\Re^n$. We can rewrite this problem in the form

$$\begin{array}{ll} \text{minimize} & f(x_1) + g(x_2) \\ \text{subject to} & x_1 - x_2 = 0 \end{array}$$

and apply the Chen–Teboulle method (2)–(5). However, this would require solving two minimization problems per iteration. Instead, we rewrite this problem in the form $0 \in T(x, y)$ with

$$y = (y_1, y_2), \quad F(x, y_1, y_2) = \partial f(x) + y_2, \quad G(x, y_1, y_2) = \begin{bmatrix} \nabla g(y_1) - y_2 \\ -x + y_1 \end{bmatrix}, \quad Y = \Re^{2n},$$

and applying the APP method with, say, $G_1 \equiv 0$ to this special case of T yields the new decomposition method:

$$\begin{aligned} \hat{y}_1^k &= y_1^k + \alpha_k(y_2^k - \nabla g(y_1^k)), \\ \hat{y}_2^k &= y_2^k + \alpha_k(x^k - y_1^k), \\ x^{k+1} &= \arg\min_{x \in \Re^n} \{f(x) + (\hat{y}_2^k)^T x + \|x - x^k\|^2/(2\alpha_k)\}, \\ y_1^{k+1} &= y_1^k + \alpha_k(\hat{y}_2^k - \nabla g(\hat{y}_1^k)), \\ y_2^{k+1} &= y_2^k + \alpha_k(x^{k+1} - \hat{y}_1^k) \end{aligned}$$

for $k = 0, 1, ...$ In cases where f has a separable structure, the above computation further decomposes. The above method has similar subproblems as the trust-region method of [8] but differs from the latter in the stepsize rule and the assumptions needed for convergence (see Theorem 2.4(c)).

Under additional assumptions on the problem (such as Lipschitz continuity of ∇g on Y), convergence and/or linear convergence of the methods in Examples 4 and 5 can be established by applying Theorem 2.4. Finally, we remark that there recently has been much study of proximal point methods using a nonquadratic proximal term, and it may be that the APP method and the associated convergence results extend to this setting.

Acknowledgment. I thank the two referees for their helpful comments which led to significant improvements on the original paper.

REFERENCES

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

[2] D. P. Bertsekas and J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice–Hall, Englewood Cliffs, NJ, 1989.

[3] H. Brézis, *Opérateurs Maximaux Monotones*, North–Holland, Amsterdam, Netherlands, 1973.

[4] H.-G. Chen, *Forward-Backward Splitting Techniques: Theory and Applications*, Ph.D. thesis, Department of Applied Mathematics, University of Washington, Seattle, WA, 1994.

[5] G. Chen and M. Teboulle, *A proximal-based decomposition method for convex minimization problems*, Math. Programming, 64 (1994), pp. 81–101.

[6] J. Eckstein and D. P. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Math. Programming, 55 (1992), pp. 293–318.

[7] J. Eckstein and M. Fukushima, *Some reformulations and applications of the alternating direction method of multipliers*, in Large Scale Optimization: State of the Art, W. W. Hager, D. W. Hearn, and P. M. Pardalos, eds., Kluwer Academic Publishers, Norwell, MA, 1994, pp. 115–134.

[8] M. Fukushima, M. Haddou, V. H. Nguyen, J.-J. Strodiot, T. Sugimoto, and Y. Yamakawa, *A parallel descent algorithm for convex programming*, Comput. Optim. Appl., 5 (1996), pp. 5–37.

[9] D. Gabay, *Applications of the method of multipliers to variational inequalities*, in Augmented Lagrangian Methods: Applications to the Solution of Boundary-Valued Problems, M. Fortin and R. Glowinski, eds., North–Holland, Amsterdam, 1983, pp. 299–331.

[10] E. M. Gafni and D. P. Bertsekas, *Two–metric projection methods for constrained optimization*, SIAM J. Control Optim., 22 (1984), pp. 936–964.

[11] O. Güler, *On the convergence of the proximal point algorithm for convex minimization*, SIAM J. Control Optim., 29 (1991), pp. 403–419.

[12] S. P. Han and G. Lou, *A parallel algorithm for a class of convex programs*, SIAM J. Control Optim., 26 (1988), pp. 345–355.

[13] G. M. Korpelevich, *The extragradient method for finding saddle points and other problems*, Matecon, 12 (1976), pp. 747–756.

[14] B. Lemaire, *The proximal algorithm*, Internat. Ser. Numer. Math., 87 (1989), pp. 73–87.

[15] P. L. Lions and B. Mercier, *Splitting algorithms for the sum of two nonlinear operators*, SIAM J. Numer. Anal., 16 (1979), pp. 964–979.

[16] Z.-Q. Luo and P. Tseng, *Error bounds and convergence analysis of feasible descent methods: A general approach*, Ann. Oper. Res., 46 (1993), pp. 157–178.

[17] P. Marcotte, *Application of Khobotov's algorithm to variational inequalities and network equilibrium problems*, Inform. Systems Oper. Res., 29 (1991), pp. 258–270.

[18] G. Minty, *On the maximal domain of a "monotone" function*, Michigan Math. J., 8 (1961), pp. 135–137.

[19] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[20] J.-S. Pang, *A posteriori error bounds for the linearly–constrained variational inequality problem*, Math. Oper. Res., 12 (1987), pp. 474–484.

[21] S. M. Robinson, *Some continuity properties of polyhedral multifunctions*, Math. Programming Study, 14 (1981), pp. 206–214.

[22] R. T. Rockafellar, *On the maximality of sums of nonlinear monotone operators*, Trans. Amer. Math. Soc., 149 (1970), pp. 75–88.

[23] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[24] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.

[25] R. T. Rockafellar, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Math. Oper. Res., 1 (1976), pp. 97–116.

[26] R. T. Rockafellar, *Computational schemes for large-scale problems in extended linear-quadratic programming*, Math. Programming, 48 (1990), pp. 447–474.

[27] R. T. Rockafellar, Private communication, December 1994.

[28] R. T. Rockafellar and R. J.-B. Wets, *Generalized linear-quadratic problems of deterministic and stochastic optimal control in discrete time*, SIAM J. Control Optim., 28 (1990), pp. 810–822.

[29] D. Salinger, *A Splitting Algorithm for Multistage Stochastic Programming with Application to Hydropower Scheduling*, Ph.D. thesis, Department of Applied Mathematics, University of Washington, Seattle, WA, 1997.

[30] J. E. Spingarn, *Applications of the method of partial inverses to convex programming*, Math. Programming, 32 (1985), pp. 199-223.

[31] P. Tseng, *Further applications of a splitting algorithm to decomposition in variational inequalities and convex programming*, Math. Programming, 48 (1990), pp. 249–263.

[32] P. Tseng, *Applications of a splitting algorithm to decomposition in convex programming and variational inequalities*, SIAM J. Control Optim., 29 (1991), pp. 119–138.

[33] P. Tseng, *On linear convergence of iterative methods for the variational inequality problem*, J. Comput. Appl. Math., 60 (1995), pp. 237–252.

[34] 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, 1971, pp. 237–424.

[35] C. Zhu and R. T. Rockafellar, *Primal-dual projected gradient algorithms for extended linear-quadratic programming*, SIAM J. Optim., 3 (1993), 751–783.

SIAM J. Optim.
Vol. 7, No. 4, pp. 966–978, November 1997

ABADIE'S CONSTRAINT QUALIFICATION, METRIC REGULARITY, AND ERROR BOUNDS FOR DIFFERENTIABLE CONVEX INEQUALITIES[*]

WU LI[†]

Abstract. In this paper we study differentiable convex inequalities and prove that metric regularity and Abadie's constraint qualification (CQ) are equivalent for such inequalities. For convex quadratic inequalities, we show that metric regularity, the existence of a global error bound, and Abadie's CQ are mutually equivalent. As a consequence, we derive two new characterizations of weak sharp minima of a convex quadratic programming problem.

Key words. differentiable convex inequalities, Abadie's constraint qualification, convex quadratic inequalities, convex quadratic programs, error bounds, metric regularity, weak sharp minima

AMS subject classifications. Primary: 90C20; Secondary: 90C30

PII. S1052623495287927

1. Introduction.

Consider a nonempty convex subset S of $\mathbb{R}^n$ defined by the following convex inequalities:

$$g(x) \le 0, \tag{1}$$

where $g(x)$ is a mapping from $\mathbb{R}^n$ to $\mathbb{R}^m$ and each component $g_i(x)$ of $g(x)$ is a convex function on $\mathbb{R}^n$. Most likely one has to resort to some iterative method for finding an approximate solution of (1). One important criterion for accuracy of an approximate solution x is the amount of constraint violation: $\|(g(x))_+\|$. Here z_+ is a vector whose ith component is $\max\{0, z_i\}$ and $\|\cdot\|$ denotes the 2-norm on $\mathbb{R}^m$ (i.e., $\|x\|^2 = \sum_{i=1}^n |x_i|^2$). There are both practical and theoretical reasons for studying the following estimate of the distance from any point x in $\mathbb{R}^n$ to the feasible set S:

$$\mathrm{dist}(x, S) \le \gamma \cdot \|(g(x))_+\|, \tag{2}$$

where γ is a positive constant and $\mathrm{dist}(x, S) := \min_{y\in S} \|x - y\|$. When $g(x)$ is affine, (2) is Hoffman's error estimate for approximate solutions of a system of linear inequalities [10]. Error estimation was crucial for establishing linear convergence of various descent methods for solving linearly constrained optimization problems [21, 22, 23, 24, 25, 26, 28, 19, 14, 16, 17, 18]. From a practical point of view, (2) guarantees that the distance from an approximate solution x to S is bounded by a multiple of $\|(g(x))_+\|$, an explicit measurement of infeasibility. Roughly speaking, one might expect that $\mathrm{dist}(x, S)$ decreases proportionally as $\|(g(x))_+\|$. However, the proportional constant γ might be large and result in an undesirable situation: $\|(g(x))_+\|$ is quite small, but x might be far away from the feasible set S. This is similar to the ill conditioning of a system of linear equations. Therefore, in order to know the accuracy of an approximate solution in terms of its distance to the feasible set, it is important to know what is the exact value of γ in estimation (2). Mangasarian defined the conditioning number of the inequality system (1) as the smallest γ for

[*] Received by the editors June 19, 1995; accepted for publication (in revised form) June 7, 1996. http://www.siam.org/journals/siopt/7-4/28792.html

[†] Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA 23529 (wuli@math.odu.edu). The research of this author is partially supported by the AFOSR under grant 94-NM-390 and by the NASA/Langley Research Center under grant NCC-1-68 Supplement-16.

which estimation (2) holds for all x [27]. There are quite a few papers devoted to the study of the conditioning number of a system of linear equalities and inequalities [7, 29, 4, 15, 5, 12, 13].

Generally, (2) does not hold if $g(x)$ is not affine. Robinson proved that (2) holds if S is bounded and has a nonempty interior [30]. For an unbounded feasible set S, Mangasarian [27] established (2) under the assumption that $g_i(x)$ are differentiable convex functions and (1) satisfies Slater's condition (i.e., there exists a point $\bar{x}$ such that $g(\bar{x}) < 0$) as well as an asymptotic CQ. Auslender and Crouzeix extended both Robinson's and Mangasarian's results by introducing a more general asymptotic CQ that can be applied to nondifferentiable convex functions $g_i(x)$. They derived (2) under Slater's condition and their asymptotic CQ [2]. However, asymptotic CQs are difficult to verify. It was not clear from Auslender and Crouzeix's result whether or not (2) holds if $g_i(x)$ are convex quadratic functions. It was proved recently by Luo and Luo [20] that (2) holds if $g_i(x)$ are convex linear/quadratic functions and there exists a feasible point $\bar{x}$ of (1) such that $g_i(\bar{x}) < 0$ whenever $g_i(x)$ is not affine. That is, for convex linear/quadratic functions, (2) holds when Slater's condition holds for nonlinear constraints. Shortly after, Pang and Wang showed that (2) might not hold for convex quadratic inequalities if Slater's condition fails [33]. They introduced an interesting concept called the degree of singularity of an inequality system and proved that if $g_i(x)$ are convex linear/quadratic functions and the degree of singularity of (1) is d, then

$$\text{(3)} \qquad \operatorname{dist}(x, S) \leq \rho \cdot \left(\|(g(x))_+\| + \|(g(x))_+\|^{2^{-d}} \right) \qquad \text{for } x \in \mathbb{R}^n.$$

They also showed by examples that the above estimate is sharp in the sense that for each $d = 0, 1, \dots$, there exists a convex quadratic inequality system such that [33]

$$\inf_{\epsilon > 0} \sup_{0 < \operatorname{dist}(x,S) \leq \epsilon} \frac{\operatorname{dist}(x, S)}{\|(g(x))_+\| + \|(g(x))_+\|^{2^{-d}}} = 1.$$

Note that the degree of singularity of (1) is always bounded by $(m+1)$. Therefore, (3) always holds with $d = m+1$ [33]. This provides a general error bound for approximate solutions of a convex quadratic inequality system, even though it might not be as sharp as one expects.

From Luo–Luo's and Wang–Pang's works [20, 33] we can appreciate the importance of Slater's condition in error estimate (2) for approximate solutions of a convex quadratic inequality system. However, one can easily construct a convex quadratic inequality system that satisfies (2) but does not satisfy Slater's condition: $g_1(x_1, x_2) = x_1 + x_2$, $g_2(x_1, x_2) = -(x_1 + x_2)$ and $g_3(x_1, x_2) = (x_1 + x_2)^2$. (It is a trivial case since the nonlinear constraint is superfluous. For nontrivial examples, see section 4.) This simple example raises a natural question: what is the characterization of a convex quadratic inequality system that satisfies (2)? It was this question that led us to the discovery of some intrinsic connections among several seemingly unrelated concepts: Abadie's CQ, metric regularity, global error bounds, and weak sharp minimum property.

The paper is organized as follows. In section 2, we give a detailed discussion of Abadie's CQ since it plays a key role in this paper. The main result in section 3 is the equivalence of Abadie's CQ and metric regularity for a differentiable convex inequality system. In section 4, we apply this characterization of metric regularity to derive a characterization of a convex quadratic inequality system that satisfies (2):

error estimate (2) holds if and only if Abadie's CQ is satisfied at every feasible point. Since we can reformulate a constrained minimization problem as an inequality system, weak sharp minimum property may be considered as a weaker form of error estimate (2). From this point of view, we establish two new characterizations of weak sharp minimum property of a convex quadratic program. Finally, a conclusion is included in section 5.

2. Abadie's CQ.

In this section, we review CQs for (1), especially Abadie's CQ. First, Abadie's CQ is a representation of the tangent cone by the gradients of active constraints, which can also be described by a representation of the normal cone by the gradients of active constraints. For differentiable convex optimization problems, Abadie's CQ is the weakest condition that ensures the characterization of an optimal solution by Karush–Kuhn–Tucker (KKT) conditions.

For a point x in a convex set S, the normal cone of S at x [32, 3, 9] is defined by

$$N_S(x) := \{z \in \mathbb{R}^n : z^T(y-x) \le 0 \text{ for } y \in S\}.$$

The tangent cone $T_S(x)$ of S at x is the polar of the normal cone $N_S(x)$. That is, $y \in T_S(x)$ if and only if $y^Tz \le 0$ for every $z \in N_S(x)$. The tangent cone $T_S(x)$ can also be defined as the closed convex cone generated by the elements in $S - x$.

Definition 2.1. *We say that the system* (1) *satisfies Abadie's CQ at $x \in S$* [1, 3] *if*

$$T_S(x) = \{y \in \mathbb{R}^n : g_i'(x)^T y \le 0 \;\; for \;\; i \in I\},$$

where $I := \{i : g_i(x) = 0\}$ is the set of indices of active constraints at x. If Abadie's CQ holds at every point in S, then we say that (1) *satisfies Abadie's CQ.*

Note that we always have $T_S(x) \subset \{y \in \mathbb{R}^n : g_i'(x)^T y \le 0 \text{ for } i \in I\}$ [9, Lemma 2.1.3]. By duality, we can also use the normal cone to describe Abadie's CQ (cf. the proof of Theorem 2.1.4 in [9]).

Lemma 2.2. *For the inequality system* (1), *Abadie's CQ is satisfied at a point $x \in S$ if and only if*

$$N_S(x) = \left\{ \sum_{i \in I} \lambda_i g_i'(x) : \lambda_i \ge 0 \;\; for \;\; i \in I \right\}, \tag{4}$$

where $I := \{i : g_i(x) = 0\}$ is the index set of active constraints at x.

Note that (4) is also called the basic CQ (BCQ) condition (cf. (2.2.1) in [9]). Thus, BCQ is equivalent to Abadie's CQ. The following result about various CQs is well-known, which implies that Abadie's CQ is the weakest one among them (cf. Figure 2.4.2 on p. 317 of [9]).

Lemma 2.3. *Consider the following CQs at a point $x \in S$:*
(LICQ): *$\{g_i'(x) : i \in I\}$ is linearly independent,*
(SCQ): *there exists $\bar{x}$ such that $g_i(\bar{x}) < 0$ for $i = 1, \ldots, m$,*
(MFCQ): *there exists a vector u such that $g_i'(x)^T u < 0$ for $i \in I$,*
(ACQ): *$T_S(x) = \{y \in \mathbb{R}^n : g_i'(x)^T y \le 0 \;\; for \;\; i \in I\}$,*
where $I := \{i : g_i(x) = 0\}$ is the index set of active constraints at x. Then

$$\text{(LICQ)} \Rightarrow \text{(SCQ)} \Leftrightarrow \text{(MFCQ)} \Rightarrow \text{(ACQ)}.$$

In general, (ACQ) $\not\Rightarrow$ (MFCQ) and (SCQ) $\not\Rightarrow$ (LICQ). Any CQ condition weaker than Abadie's CQ is not very useful since Abadie's CQ is the weakest condition that ensures the KKT characterization for an optimal solution of a differentiable convex optimization problem (cf. Lemma 2.4).

Consider the minimization of a differentiable convex function $f(x)$ on $\mathbb{R}^n$ subject to inequality constraints (1):

$$\min f(x) \quad \text{subject to } g_i(x) \leq 0 \text{ for } i = 1, \dots, m. \tag{5}$$

We say that $\bar{x}$ is a KKT point of (5) [32, 3, 9] if there exist nonnegative scalars λ_i such that

$$f'(\bar{x}) + \sum_{i \in I} \lambda_i g_i'(\bar{x}) = 0,$$

where $I := \{i : g_i(\bar{x}) = 0\}$ is the index set of active constraints at $\bar{x}$. Then one can use the KKT characterization for solutions of (5) to describe Abadie's CQ [9, Proposition 2.2.1].

LEMMA 2.4. *The following two statements are equivalent.*

(2.4.1) *The system* (1) *satisfies Abadie's CQ.*

(2.4.2) *For any differentiable convex function $f(x)$ on $\mathbb{R}^n$, $\bar{x}$ is an optimal solution of* (5) *if and only if $\bar{x}$ is a KKT point of* (5).

Finally we list a commonly used Slater-type CQ that implies Abadie's CQ (cf. section 2 of Chapter VII or Figure 2.4.2 of [9]).

LEMMA 2.5. *Suppose that there exists a point $\bar{x}$ such that $g_i(\bar{x}) \leq 0$ for $i = 1, \dots, m$ and $g_i(\bar{x}) < 0$ if $g_i(x)$ is not an affine function. Then* (1) *satisfies Abadie's CQ.*

3. Metric regularity and Abadie's CQ.

It is well known that metric regularity is related to Slater's condition and MFCQ [30, 31, 6]. In this section we prove that Abadie's CQ is equivalent to metric regularity for a convex differentiable inequality system.

Following the definition of metric regularity for set-valued mappings (or multifunctions) (cf. [6, Definition 2.1] or [11, Definition 1.1]) we give a definition of metric regularity for (1).

DEFINITION 3.1. *We say that the system* (1) *is metrically regular at a point $\bar{x} \in S$ if there exist positive constants γ and δ such that*

$$\operatorname{dist}(x, S) \leq \gamma \cdot \sum_{i=1}^{m} (g_i(x))_+ \quad \textit{when } \|x - \bar{x}\| \leq \delta.$$

We say that the system (1) *is metrically regular if it is metrically regular at every point in S.*

Note that we are interested in metric regularity of (1) at every point in S. In general, one needs Slater's condition to ensure such a metric regularity as shown in the following lemma that follows from a more general result by Robinson (cf. section 3 of [30]).

LEMMA 3.2. *If there exists $\bar{x} \in \mathbb{R}^n$ such that $g_i(\bar{x}) < 0$ for $i = 1, \dots, m$, then* (1) *is metrically regular.*

Remark. In fact, in section 3 of [30], Robinson proved the following inequality:

$$\operatorname{dist}(x, S) \leq \gamma \|x - \bar{x}\| \cdot \sum_{i=1}^{m} (g_i(x))_+ \text{ for } x \in \mathbb{R}^n, \tag{6}$$

where γ is a positive scalar. Note that (6) implies the metric regularity of (1). From (6) we obtain error bounds for infeasible solutions of (1) on bounded subsets of $\mathbb{R}^n$:

$$\operatorname{dist}(x, S) \leq \gamma_r \cdot \sum_{i=1}^{m} (g_i(x))_+ \quad \text{when } \|x\| \leq r, \tag{7}$$

where γ_r is a positive scalar depending on r. This shows that metric regularity is closely related to error bounds. In fact, metric regularity of (1) is equivalent to error bounds for infeasible solutions of (1) on bounded subsets of $\mathbb{R}^n$.

THEOREM 3.3. *The system* (1) *is metrically regular if and only if for any scalar* $r > 0$ *there exists a positive constant* γ_r *such that*

$$\operatorname{dist}(x, S) \leq \gamma_r \cdot \sum_{i=1}^{m} (g_i(x))_+ \quad \textit{when } \|x\| \leq r. \tag{8}$$

Proof. Obviously, (8) implies metric regularity of (1). Now assume that (1) is metrically regular. Then, for each $\bar{x} \in S$, there exist positive scalars $\delta_{\bar{x}}$ and $\gamma_{\bar{x}}$ such that

$$\operatorname{dist}(x, S) \leq \gamma_{\bar{x}} \cdot \sum_{i=1}^{m} (g_i(x))_+ \quad \text{when } \|x - \bar{x}\| \leq \delta_{\bar{x}}. \tag{9}$$

Let $x^* \in S$ and $S_r := \{x \in S : \|x\| \leq 2r + \|x^*\|\}$. Then S_r is compact. Moreover,

$$S_r \subset \bigcup_{\bar{x} \in S_r} B(\bar{x}, \delta_{\bar{x}}), \tag{10}$$

where $B(\bar{x}, \delta_{\bar{x}}) := \{x \in \mathbb{R}^n : \|x - \bar{x}\| < \delta_{\bar{x}}\}$ is the open ball in $\mathbb{R}^n$ with center $\bar{x}$ and radius $\delta_{\bar{x}}$. By the compactness of S_r and (10), there exist points $\{x_1, \ldots, x_k\} \subset S_r$ such that

$$S_r \subset \bigcup_{j=1}^{k} B(x_j, \delta_{x_j}). \tag{11}$$

For $x \in \mathbb{R}^n$ with $\|x\| \leq r$, let $P_S(x)$ be the projection of x onto S; i.e., $P_S(x) \in S$ with $\operatorname{dist}(x, S) = \|x - P_S(x)\|$. Then

$$\|P_S(x)\| \leq \|x\| + \|x - P_S(x)\| \leq r + \|x - x^*\| \leq r + \|x\| + \|x^*\| \leq 2r + \|x^*\|.$$

Thus, $P_S(x) \in S_r$. By (11), $P_S(x) \in B(x_j, \delta_{x_j})$ for some x_j. Since $B(x_j, \delta_{x_j})$ is open, there exists $0 < \theta < 1$ such that

$$x_\theta := \theta x + (1 - \theta) P_S(x) \in B(x_j, \delta_{x_j}).$$

By (9), we obtain

$$\operatorname{dist}(x_\theta, S) \leq \gamma_{x_j} \cdot \sum_{i=1}^{m} (g_i(x_\theta))_+. \tag{12}$$

By the convexity of g_i and $g_i(P_S(x)) \leq 0$, we get

$$g_i(x_\theta) \leq \theta g_i(x) + (1 - \theta) g_i(P_S(x)) \leq \theta g_i(x),$$

which implies

$$(g_i(x_\theta))_+ \le (\theta g_i(x))_+ = \theta (g_i(x))_+. \tag{13}$$

By the definition of x_θ, we have

$$\begin{aligned}\|x - P_S(x)\| &\le \|x - P_S(x_\theta)\| \\ &\le \|x - x_\theta\| + \|x_\theta - P_S(x_\theta)\| \\ &= (1-\theta)\|x - P_S(x)\| + \|x_\theta - P_S(x_\theta)\|,\end{aligned}$$

which implies

$$\theta \mathrm{dist}(x, S) \le \mathrm{dist}(x_\theta, S). \tag{14}$$

It follows from (12), (13), and (14) that

$$\mathrm{dist}(x, S) \le \frac{1}{\theta}\mathrm{dist}(x_\theta, S) \le \frac{\gamma_r}{\theta} \cdot \sum_{i=1}^{m} (g_i(x_\theta))_+ \le \gamma_r \cdot \sum_{i=1}^{m} (g_i(x))_+,$$

where $\gamma_r := \max\{\gamma_{x_j} : 1 \le j \le k\}$. □

Before we prove the equivalence of metric regularity and Abadie's CQ for (1), we need the following simple fact about nonnegative linear combination of vectors in $\mathbb{R}^n$ [32, Corollary 17.1.2].

LEMMA 3.4. *Suppose that $y, u_i \in \mathbb{R}^n$ and $y = \sum_{i=1}^{r} \alpha_i u^i \ne 0$ for some nonnegative scalars α_i. Then there exist nonnegative scalars $\lambda_1, \ldots, \lambda_r$ such that*

(3.4.1) *$y = \sum_{i=1}^{r} \lambda_i u^i$,*

(3.4.2) *$\{u^i : \lambda_i \ne 0\}$ are linearly independent.*

Now we are ready to prove the main theorem in this section.

THEOREM 3.5. *The system* (1) *is metrically regular if and only if* (1) *satisfies Abadie's CQ.*

Proof. First we show that metric regularity of (1) implies Abadie's CQ.

Let $S := \{x \in \mathbb{R}^n : g(x) \le 0\}$ be the set of all feasible points of (1). If there is $\bar{x} \in \mathbb{R}^n$ such that $g(\bar{x}) < 0$, then (1) satisfies the Slater condition; hence, it also satisfies Abadie's CQ (cf. Lemma 2.3). Otherwise, for any point $\bar{x} \in S$, consider the set

$$\bar{S} := \{x \in \mathbb{R}^n : g_i'(\bar{x})^T (x - \bar{x}) \le 0 \text{ for } i \in I\},$$

where $I := \{i : g_i(\bar{x}) = 0\}$. Since $\bar{S}$ is a polyhedral set, by Proposition 2.2.2 in [9],

$$N_{\bar{S}}(\bar{x}) = \left\{ \sum_{i \in I} \lambda_i g_i'(\bar{x}) : \lambda_i \ge 0 \right\}.$$

Since $S \subset \bar{S}$, the normal cone $N_{\bar{S}}(\bar{x})$ is a subset of the normal cone $N_S(\bar{x})$. In view of Lemma 2.2, our goal is to show that $N_S(\bar{x}) = N_{\bar{S}}(\bar{x})$. We prove this by contradiction. If there is $u \in N_S(\bar{x}) \setminus N_{\bar{S}}(\bar{x})$, then there exists $z \in \bar{S}$ such that

$$u^T (z - \bar{x}) > 0, \tag{15}$$

while

$$u^T (x - \bar{x}) \le 0 \quad \text{for } x \in S. \tag{16}$$

Since $z \in \bar{S}$, we have

$$g_i'(\bar{x})^T(z-\bar{x}) \le 0 \text{ for } i \in I. \tag{17}$$

Let $0 < \alpha < 1$ and $x(\alpha) = \alpha z + (1-\alpha)\bar{x}$. Then, for any $x \in S$,

$$\begin{aligned} &\|x(\alpha)-\bar{x}\| = \alpha\|z-\bar{x}\| \\ &= \alpha\gamma u^T(z-\bar{x}) = \gamma u^T(x(\alpha)-\bar{x}) \\ &= \gamma\left(u^T(x(\alpha)-x) + u^T(x-\bar{x})\right) \\ &\le \gamma u^T(x(\alpha)-x) \le \gamma\|u\|\cdot\|x(\alpha)-x\|, \end{aligned}$$

where $\gamma := \frac{\|z-\bar{x}\|}{u^T(z-\bar{x})} > 0$, the first inequality follows from (16), and the last inequality is by the Cauchy–Schwarz inequality. As a consequence,

$$\operatorname{dist}(x(\alpha), S) \ge \frac{1}{\gamma\|u\|}\|x(\alpha)-\bar{x}\| > 0. \tag{18}$$

By the assumption, we have metric regularity at $\bar{x} \in S$. That is, there exist positive constants λ and ϵ such that

$$\operatorname{dist}(x, S) \le \lambda\sum_{i=1}^{m}(g_i(x))_+ \quad \text{when } \|x-\bar{x}\| \le \epsilon. \tag{19}$$

Since $x(\alpha) \to \bar{x}$ as $\alpha \to 0^+$, it follows from (19) that

$$\limsup_{\alpha\to 0^+} \frac{\sum_{i=1}^m (g_i(x(\alpha)))_+}{\operatorname{dist}(x(\alpha), S)} \ge \frac{1}{\lambda} > 0. \tag{20}$$

Since $\lim_{\alpha\to 0^+} g_i(x(\alpha)) = g_i(\bar{x}) < 0$ when $i \notin I$, we have

$$\lim_{\alpha\to 0^+} \frac{(g_i(x(\alpha)))_+}{\operatorname{dist}(x(\alpha), S)} = 0 \quad \text{for } i \notin I. \tag{21}$$

Let $e_i(\alpha) := g_i(x(\alpha)) - g_i(\bar{x}) - g_i'(\bar{x})^T(x(\alpha)-\bar{x})$. By the differentiability of g_i, we have

$$\lim_{\alpha\to 0^+} \frac{|e_i(\alpha)|}{\|x(\alpha)-\bar{x}\|} = 0.$$

From (18) and the above limit, we get

$$\limsup_{\alpha\to 0^+} \frac{|e_i(\alpha)|}{\operatorname{dist}(x(\alpha), S)} \le \lim_{\alpha\to 0^+} \gamma\|u\|\frac{|e_i(\alpha)|}{\|x(\alpha)-\bar{x}\|} = 0. \tag{22}$$

However, for $i \in I$, it follows from (17) that $g_i(x(\alpha)) = g_i(x(\alpha)) - g_i(\bar{x}) \le e_i(\alpha)$ and, as a consequence, $(g_i(x(\alpha)))_+ \le |e_i(\alpha)|$. By (22), we have

$$\lim_{\alpha\to 0^+} \frac{(g_i(x(\alpha)))_+}{\operatorname{dist}(x(\alpha), S)} = 0 \quad \text{for } i \in I. \tag{23}$$

The limits (21) and (23) imply that

$$\limsup_{\alpha\to 0^+} \frac{\sum_{i=1}^m (g_i(x(\alpha)))_+}{\operatorname{dist}(x(\alpha), S)} = 0,$$

which contradicts the inequality (20). Thus, by contradiction, we have proved that the system (1) satisfies Abadie's CQ.

Now we prove that Abadie's CQ implies metric regularity of (1).

Let $\bar{x} \in S$ and $I := \{i : g_i(\bar{x}) = 0\}$. Since $g_i(\bar{x}) < 0$ for each $i \notin I$ and g_i are continuous, there exists a positive constant δ_0 such that

$$g_i(x) < 0 \quad \text{when } i \notin I, \|x - \bar{x}\| \le 2\delta_0. \tag{24}$$

Let $\mathcal{I}$ be the collection of all nonempty index sets $J (\subset I)$ such that the inequality system $g_i(x) \le 0$ for $i \in J$ satisfies the Slater condition.

Assume that $\mathcal{I}$ is not empty. Let $J \in \mathcal{I}$. By Lemma 3.2, there exist positive constants $\gamma(J)$ and $\delta(J)$ such that

$$\operatorname{dist}(x, S(J)) \le \gamma(J) \cdot \sum_{i \in J} (g_i(x))_+ \quad \text{when } \|x - \bar{x}\| \le \delta(J),$$

where $S(J) := \{x : g_i(x) \le 0 \text{ for } i \in J\}$. Let $\delta := \min\{\delta_0, \delta(J) : J \in \mathcal{I}\}$ and $\gamma := \max\{\gamma(J) : J \in \mathcal{I}\}$. Since $\mathcal{I}$ is a finite set, $\delta > 0$ and $\gamma > 0$. Moreover,

$$\operatorname{dist}(x, S(J)) \le \gamma \cdot \sum_{i \in J} (g_i(x))_+ \quad \text{when } \|x - \bar{x}\| \le \delta, J \in \mathcal{I}. \tag{25}$$

For any point $x \notin S$ with $\|x - \bar{x}\| \le \delta$, there exists a unique point $x^* \in S$ such that

$$\frac{1}{2}\|x - x^*\|^2 = \frac{1}{2}\operatorname{dist}(x, S)^2 = \min_{z \in S} \frac{1}{2}\|x - z\|^2 > 0.$$

Since (1) satisfies Abadie's CQ, by Lemma 2.4 there exist nonnegative scalars α_i such that

$$x - x^* = \sum_{i \in I^*} \alpha_i g'(x^*) \ne 0,$$

where $I^* := \{i : g_i(x^*) = 0\}$ is the set of indices of active constraints at x^*. By Lemma 3.4, there exists an index set $J \subset I^*$ and nonnegative scalars λ_i such that

$$\begin{aligned} &x - x^* = \textstyle\sum_{i \in J} \lambda_i g_i'(x^*) \ne 0, \\ &\{g_i'(x^*) : i \in J\} \text{ are linearly independent.} \end{aligned} \tag{26}$$

By Lemma 2.3, the above linear independence CQ implies the Slater condition for the inequality system $g_i(x) \le 0$ for $i \in J$.

Since

$$\|x^* - \bar{x}\| \le \|x^* - x\| + \|x - \bar{x}\| \le 2\|x - \bar{x}\| \le 2\delta \le 2\delta_0,$$

by (24), $g_i(x^*) < 0$ for $i \notin I$. Thus, $J \subset I$. As a consequence, $J \in \mathcal{I}$ and $\mathcal{I}$ is not empty. By (25),

$$\operatorname{dist}(x, S(J)) \le \gamma \sum_{i \in J} (g_i(x))_+. \tag{27}$$

It follows from the first equation in (26), $x^* \in S(J)$, and Lemma 2.4 that

$$\operatorname{dist}(x, S(J)) = \|x - x^*\| = \operatorname{dist}(x, S).$$

Therefore, we derive from (27) that

$$\operatorname{dist}(x, S) \leq \gamma \sum_{i=1}^{m} (g_i(x))_+ \quad \text{when } \|x - \bar{x}\| \leq \delta, x \notin S,$$

which implies

$$\operatorname{dist}(x, S) \leq \gamma \sum_{i=1}^{m} (g_i(x))_+ \quad \text{when } \|x - \bar{x}\| \leq \delta. \tag{28}$$

Note that if $\mathcal{I}$ is empty, then the above proof shows

$$\{x \in \mathbb{R}^n : \|x - \bar{x}\| \leq \delta_0\} \subset S.$$

Thus, (28) also holds with $\gamma = 1$ and $\delta = \delta_0$.

Since (28) holds for any $\bar{x} \in S$, the system (1) is metrically regular at every point $\bar{x} \in S$. This proves the metric regularity of (1). □

4. Error bounds. We want to apply the main theorem in the previous section (Theorem 3.5) to a special case: (1) with convex linear/quadratic functions $g_i(x)$. In this case, the metric regularity is equivalent to the existence of a global error bound for infeasible solutions of (1). As a consequence, we obtain that Abadie's CQ is a necessary and sufficient condition for a global error bound given in (2). Our result complements the study done by Luo and Luo [20] as well as Wang and Pang [33] on error bounds for convex quadratic inequalities.

Consider the following system of convex quadratic inequalities:

$$g_i(x) \leq 0 \quad \text{for } i = 1, \ldots, m, \tag{29}$$

where $g_i(x)$ are either affine or convex quadratic functions on $\mathbb{R}^n$.

The essence of our proof is to reduce the problem to the case that Slater's condition holds. Then we can use the following result by Luo and Luo [20, Lemma 3.5] to get (2).

LEMMA 4.1. *If the system* (29) *satisfies the Slater condition, then there exists a positive constant* γ *such that*

$$\left\| \sum_{i \in I(x)} \lambda_i g_i'(x) \right\| \geq \gamma \sum_{i \in I(x)} \lambda_i \quad for\ x \in \mathbb{R}^n, \lambda_i \geq 0, \tag{30}$$

where $I(x) := \{i : g_i(x) = 0\}$.

It is obvious that (2) implies metric regularity. Therefore, the main effort in proving the equivalence of (2) and metric regularity is to show that metric regularity implies (2) for convex quadratic inequalities.

THEOREM 4.2. *The convex quadratic inequality system* (29) *satisfies Abadie's CQ if and only if there exists a positive constant* γ *such that*

$$\operatorname{dist}(x, S) \leq \gamma \sum_{i=1}^{m} (g_i(x))_+ \quad for\ x \in \mathbb{R}^n, \tag{31}$$

where $S := \{x \in \mathbb{R}^n : g_i(x) \leq 0\ for\ i = 1, \ldots, m\}$.

Proof. Since (31) implies the metric regularity of (29), by Theorem 3.5 (29) satisfies Abadie's CQ. On the other hand, if (29) satisfies Abadie's CQ, then for any $x \in \mathbb{R}^n$ the KKT conditions hold for the projection x^* from x onto S (cf. Lemma 2.4):

$$x^* - x + \sum_{i \in I} \lambda_i g_i'(x^*) = 0, \tag{32}$$

where $x^* \in S$ with $\|x^* - x\| = \operatorname{dist}(x, S)$, λ_i are nonnegative scalars, and $I := \{i : g_i(x^*) = 0\}$. By Lemma 3.4, we may assume that $\{g_i'(x^*) : i \in I, \lambda_i \neq 0\}$ are linearly independent. Let $\bar{I} := \{i \in I : \lambda_i \neq 0\}$. It follows from Lemma 2.3 that the system $g_i(x) \leq 0$ for $i \in \bar{I}$ satisfies the Slater CQ. By Lemma 4.1, we have

$$\operatorname{dist}(x, S) = \|x^* - x\| = \left\| \sum_{i \in \bar{I}} \lambda_i g_i'(x^*) \right\| \geq \gamma(\bar{I}) \sum_{i \in \bar{I}} \lambda_i, \tag{33}$$

where $\gamma(\bar{I})$ is a positive constant depending only on $\{g_i : i \in \bar{I}\}$. As a consequence,

$$\operatorname{dist}(x, S)^2 = (x - x^*)^T (x - x^*) = \left(\sum_{i \in \bar{I}} \lambda_i g_i'(x^*) \right)^T (x - x^*)$$

$$= \sum_{i \in \bar{I}} \lambda_i g_i'(x^*)^T (x - x^*) \leq \sum_{i \in \bar{I}} \lambda_i g_i(x) \leq \left(\sum_{i \in \bar{I}} \lambda_i \right) \sum_{i=1}^{m} (g_i(x))_+,$$

where the second equality is from (32), the first inequality follows from convexity of g_i, and the second inequality is derived from $\lambda_i \geq 0$ and $g_i(x) \leq (g_i(x))_+$. The above estimate of $\operatorname{dist}(x, S)$, along with (33), yields

$$\operatorname{dist}(x, S) \leq \frac{1}{\gamma(\bar{I})} \sum_{i=1}^{m} (g_i(x))_+.$$

Since there are only finitely many different $\bar{I}$, (31) holds with

$$\gamma := \max \left\{ \frac{1}{\gamma(\bar{I})} \right\} < \infty.$$

This completes the proof of Theorem 4.2. □

Note that Luo and Luo [20, Theorem 3.1] proved a special case of Theorem 4.2: (31) holds if there exists a vector $\bar{x}$ such that $g(\bar{x}) \leq 0$ and $g_i(\bar{x}) < 0$ whenever $g_i(x)$ is not an affine function (cf. Lemma 2.5).

Theorem 4.2 not only gives a characterization of the existence of global error bound (2) for convex quadratic inequalities but also reveals why there exist weak sharp minima for convex quadratic programming problems [8], as shown in the following theorem.

Theorem 4.3. *Assume that $f(x)$ is a convex quadratic function bounded below on $\{x \in \mathbb{R}^n : Ax \leq b\}$. Let $f_{\min} := \min_{Ax \leq b} f(x)$ and $S := \{x \in \mathbb{R}^n : Ax \leq b, f(x) = f_{\min}\}$. Then the following statements are equivalent.*

(4.3.1) *Abadie's CQ is satisfied at every feasible point of the following inequality system:*

$$f(x) - f_{\min} \leq 0 \textit{ and } Ax - b \leq 0. \tag{34}$$

(4.3.2) *The convex quadratic programming problem* $\min_{Ax \leq b} f(x)$ *has weak sharp minima. That is, there exists a positive constant* γ *such that*

$$f(x) \geq f_{\min} + \gamma \cdot \mathrm{dist}(x, S) \textit{ when } Ax \leq b. \tag{35}$$

(4.3.3) *There exists a positive constant* λ *such that*

$$\mathrm{dist}(x, S) \leq \lambda \left((f(x) - f_{\min})_+ + \|(Ax - b)_+\| \right) \quad \textit{for } x \in \mathbb{R}^n. \tag{36}$$

Proof. By Theorem 4.2, (4.3.1) ⇔ (4.3.3). Obviously, (4.3.3) ⇒ (4.3.2). It suffices to prove that (4.3.2) ⇒ (4.3.1).

Now assume that (35) holds for some $\gamma > 0$. For any $\bar{x} \in S$, let

$$\bar{S} := \{x \in \mathbb{R}^n : f'(\bar{x})^T (x - \bar{x}) \leq 0, Ax \leq b\}$$

and $I := \{i : A_i \bar{x} - b_i = 0\}$, where A_i is the ith row of A and b_i is the ith component of b. Then, by Proposition 2.2.2 in [9],

$$N_{\bar{S}}(\bar{x}) = \left\{ \lambda_0 f'(\bar{x}) + \sum_{i \in I} \lambda_i A_i^T : \lambda_0, \lambda_i \geq 0 \right\}.$$

Since $f(\bar{x}) - f_{\min} = 0$, by Lemma 2.2 (34) satisfies Abadie's CQ at $\bar{x}$ if and only if

$$N_S(\bar{x}) = N_{\bar{S}}(\bar{x}). \tag{37}$$

However, by Ferris and Mangasarian's characterization of weak sharp minima for convex quadratic programs, (35) implies $S = \bar{S}$ [8, Theorem 6]. Hence, (37) holds. This proves the implication (4.3.2) ⇒ (4.3.1). □

Note that various characterizations of weak sharp minima of a convex quadratic programming problem were given by Ferris and Mangasarian [8, Theorem 6]. Theorem 4.2 leads us to two new characterizations (4.3.1) and (4.3.3) in Theorem 4.3. Weak sharp minimum inequality (35) estimates how far away a feasible solution is from the solution set. The inequality (36) actually provides an estimate of the distance from any approximate solution of the quadratic programming problem to its solution set, which is more desirable when infeasible approximate solutions are involved. Even though (36) fails to be true if (34) does not satisfy Abadie's CQ, one could still have the following inequality [18, Corollary 2.8]:

$$\mathrm{dist}(x, S) \leq \gamma \left(f(x) - f_{\min} + \sqrt{f(x) - f_{\min}} \right) \quad \text{for } Ax - b \leq 0,$$

where $f(x)$ is a convex piecewise quadratic function.

5. Conclusion. We have shown that the concepts of metric regularity, error bounds, and weak sharp minimum are closely related. The essence of these concepts is to estimate the distance from an approximate solution to the solution set of the underlying problem, locally or globally. For metric regularity of parametric systems, MFCQ was proven to be a necessary and sufficient condition [31, Theorem 1] (cf.

also Corollary 2.2 and the comments after it in [6]). However, for the nonparametric version of metric regularity defined in this paper, Abadie's CQ is a necessary and sufficient condition. As applications, we prove that Abadie's CQ is a characterization for the existence of a global error bound (2) for convex quadratic inequalities, which leads to a global error bound (36) for approximate solutions of a convex quadratic programming problem with weak sharp minima.

Acknowledgment. The author would like to thank Ivan Singer for helpful comments that simplify the presentation of section 2.

REFERENCES

[1] J. Abadie, *On the Kuhn–Tucker theorem*, in Nonlinear Programming, J. Abadie, ed., North–Holland, Amsterdam, 1967, pp. 19–36.

[2] A. A. Auslender and J.-P. Crouzeix, *Global regularity theorems*, Math. Oper. Res., 13 (1988), pp. 243–253.

[3] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming, Theory and Algorithms*, 2nd ed., John Wiley & Sons, Inc., New York, 1993.

[4] C. Bergthaller and I. Singer, *The distance to a polyhedron*, Linear Algebra Appl., 169 (1992), pp. 111–129.

[5] J. V. Burke and P. Tseng, *A unified analysis of Hoffman's bound via Fenchel duality*, SIAM J. Optim., 6 (1996), pp. 265–282.

[6] R. Cominetti, *Metric regularity, tangent sets, and second-order optimality conditions*, Appl. Math. Optim., 21 (1990), pp. 265–287.

[7] W. Cook, A. M. H. Gerards, A. Schrijver, and É. Tardos, *Sensitivity theorems in integer linear programming*, Math. Programming, 34 (1986), pp. 251–264.

[8] M. C. Ferris and O. L. Mangasarian, *Minimum principle sufficiency*, Math. Programming, 57 (1992), pp. 1–14.

[9] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms* (I), Springer-Verlag, New York, 1993.

[10] A. J. Hoffman, *On approximate solutions of systems of linear inequalities*, J. Res. Natl. Bur. Standards, 49 (1952), pp. 263–265.

[11] A. Jourani, *Regularity and strong sufficient optimality conditions in differentiable optimization problems*, Numer. Funct. Anal. Optim., 14 (1993), pp. 69–87.

[12] D. Klatte and G. Thiere, *A note on Lipschitz constants for solutions of linear inequalities and equations*, Linear Algebra Appl., 236 (1996), pp. 365–374.

[13] D. Klatte and G. Thiere, *Error bounds for solutions of linear equations and inequalities*, Math. Methods Oper. Res., 41 (1995), pp. 191–214.

[14] W. Li, *Remarks on convergence of the matrix splitting algorithm for the symmetric linear complementarity problem*, SIAM J. Optim., 3 (1993), pp. 155–163.

[15] W. Li, *The sharp Lipschitz constants for feasible and optimal solutions of a perturbed linear program*, Linear Algebra Appl., 187 (1993), pp. 15–40.

[16] W. Li, *Linearly convergent descent methods for unconstrained minimization of a convex quadratic spline*, J. Optim. Theory Appl., 86 (1995), pp. 145–172.

[17] W. Li, *Error bounds for piecewise quadratic programs and applications*, SIAM J. Control Optim., 33 (1995), pp. 1510–1529.

[18] W. Li, *A conjugate gradient method for strictly convex quadratic programs with simple bound constraints*, Math. Programming, 72 (1996), pp. 17–32.

[19] W. Li, P. Pardalos, and C. G. Han, *Gauss-Seidel method for least distance problems*, J. Optim. Theory Appl., 75 (1992), pp. 487–500.

[20] X.-D. Luo and Z.-Q. Luo, *Extension of Hoffman's error bound to polynomial systems*, SIAM J. Optim., 4 (1994), pp. 383–392.

[21] Z.-Q. Luo and P. Tseng, *On the convergence of a matrix splitting algorithm for the symmetric monotone linear complementarity problem*, SIAM J. Control Optim., 29 (1991), pp. 1037–1060.

[22] Z.-Q. Luo and P. Tseng, *Error bound and convergence analysis of matrix splitting algorithms for the affine variational inequality problem*, SIAM J. Optim., 2 (1992), pp. 43–54.

[23] Z.-Q. Luo and P. Tseng, *On the convergence of the coordinate descent method for convex differentiable minimization*, J. Optim. Theory Appl., 72 (1992), pp. 7–35.

[24] Z.-Q. Luo and P. Tseng, *On the linear convergence of descent methods for convex essentially smooth minimization*, SIAM J. Control Optim., 30 (1992), pp. 408–425.

[25] Z.-Q. Luo and P. Tseng, *Error bounds and convergence analysis of feasible descent methods: A general approach*, Ann. Oper. Res., 46-47 (1993), pp. 157–178.

[26] Z.-Q. Luo and P. Tseng, *On the convergence rate of dual ascent methods for strictly convex minimization*, Math. Oper. Res., 18 (1993), pp. 846–867.

[27] O. L. Mangasarian, *A condition number for differentiable convex inequalities*, Math. Oper. Res., 10 (1985), pp. 175–179.

[28] O. L. Mangasarian, *Convergence of iterates of an inexact matrix splitting algorithm for the symmetric monotone linear complementarity problem*, SIAM J. Optim., 1 (1991), pp. 114–122.

[29] O. L. Mangasarian and T. H. Shiau, *Lipschitz continuity of solutions of linear inequalities, programs and complementarity problems*, SIAM J. Control Optim., 25 (1987), pp. 583–595.

[30] S. M. Robinson, *An application for error bounds for convex programming in a linear space*, SIAM J. Control Optim., 13 (1975), pp. 271–273.

[31] S. M. Robinson, *Stability theorems for systems of inequalities, part* II*: Differentiable nonlinear systems*, SIAM J. Numer. Anal., 13 (1976), pp. 497–513.

[32] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[33] T. Wang and J.-S. Pang, *Global error bounds for convex quadratic inequality systems*, Optimization, 31 (1994), pp. 1–12.

SIAM J. Optim.
Vol. 7, No. 4, pp. 979–990, November 1997

006

MATHEMATICAL STUDY OF VERY HIGH VOLTAGE POWER NETWORKS I: THE OPTIMAL DC POWER FLOW PROBLEM*

J. FRÉDÉRIC BONNANS[†]

Abstract. The optimal power flow problem involves setting the voltage and power delivered at the nodes of an electrical network in order to minimize the loss of power over the lines. This paper is the first of a series dedicated to the mathematical study of this problem. We use an asymptotic analysis in which the small parameter is the inverse of the reference voltage of the network. We call this scheme the very high voltage approximation. Here we deal with the case of direct current. We obtain an analytic expansion for the optimal value and the solution.

Key words. electrical networks, nonlinear optimization, asymptotic analysis, sensitivity analysis, expansion of solutions, directional constraint qualification

AMS subject classifications. 49K40, 90C31, 58C15

PII. S1052623494278025

Notation.

$F(\mathcal{P})$, set of feasible points of an optimization problem $(\mathcal{P})$,
$S(\mathcal{P})$, set of solutions of an optimization problem $(\mathcal{P})$,
$v(\mathcal{P})$, optimal value of an optimization problem $(\mathcal{P})$,
$\mathcal{S}$, set of nodes of the network, numbered from 1 to n, excluding the reference node 0,
n, cardinality of $\mathcal{S}$, $(n = |\mathcal{S}|)$,
$Y_{k\ell}$, admittance between nodes k and ℓ,
$I_{k\ell}$, current from node k to node ℓ,
V_k, voltage at node k,
V^R, voltage at reference node,
J_k, input of current at node k,
P_k, input of power at node k; $P_k := V_k J_k$,
Z, admittance matrix.

1. Introduction.

The optimal power flow problem is an important issue, which involves setting in an optimal way the voltage and power delivered at the nodes of an alternating current (AC) network. The distribution of voltage and power is subject to certain bounds and must comply with Kirchhoff's and Ohm's laws. A typical criterion is to minimize the loss of energy over the network (see, e.g., Blanchon, Bonnans, and Dodu [3]).

This paper is the first of a series dedicated to the mathematical study of this problem. We use an asymptotic analysis in which the small parameter is, roughly speaking, the inverse of the square root of the nominal voltage of the network. We call this scheme the high voltage approximation. This choice of a small parameter is natural, as industrial networks use very high values for the voltage. The approximation scheme gives considerable insight into the problem, as the limit problem (which, after a proper scaling, is well defined) has its active and reactive parts decoupled. This means that for a sufficiently high voltage the coupling between active and reactive parts is weak, a property that can be very useful for numerical purposes. Indeed, on the basis of our perturbation analysis, one can prove the rapid convergence of some algorithms with decoupled equations (this was the original motivation of our

* Received by the editors December 1, 1994; accepted for publication (in revised form) June 20, 1996. This study was supported by Electricité de France.

http://www.siam.org/journals/siopt/7-4/27802.html

[†] INRIA-Rocquencourt, Domaine de Voluceau, B.P. 105, 78153 Rocquencourt, France (Frederic.Bonnans@inria.fr).

study). These algorithms can be used for solving the power flow problem (without optimization) as well as for the optimal power flow problem.

The main mathematical tool of these papers is the perturbation theory for nonlinear programming, a subject in which significant progress has been made in the last few years, e.g., [2], [6], [10], [15], and the review [7]. Indeed, this study can be viewed as a real-world application of the above-mentioned theory.

While the application deals with AC, it is useful to consider first the analogous problem with direct current. This allows us to study a problem with much simpler equations that nevertheless retains some of the flavor of the real application. Even for readers whose primary interest lies only in real world applications, it is advisable to read this paper first in order to get accustomed to some basic tools of the perturbation theory for nonlinear programming, whereas mathematicians will be pleased, as always, to deal with a simplified model that allows a complete mathematical discussion.

The other parts of this study are devoted to AC networks. Part II [16] discusses the *power flow problem* (without optimization) for which an early reference is Aubin and Raviart [1]. There the high voltage approximation is combined with the hypothesis of small real part of impedances. In part III [17] we obtain our final results, namely the expansion of solutions, by applying the perturbation theory for nonlinear programming to the *AC optimal power flow problem* in the framework of the high voltage approximation.

The present paper is structured as follows. In section 2, we review the equation of the direct current power network and state the problem of minimizing the loss of power over the network. Section 3 introduces the high voltage approximation. We show that, after a proper scaling, the limit problem is well posed, and we exhibit its solution. In section 4, we review the mathematical tools (Auslender and Cominetti [2], Bonnans and Sulem [8]) from the perturbation theory for nonlinear programming that are needed. We combine these two results in order to state a third. Section 5 is devoted to the analysis of the limit problem. We note that the limiting problem has nonqualified constraints, although there exist multipliers associated with the solution. In section 6, we obtain the analytic expansion for the optimal value and the solution. We ultimately give physical interpretations of the expansion of the solution in section 7.

2. Presentation of the optimal direct current power flow problem. We consider a network composed of passive elements, namely resistances, with a possible injection of current at the nodes. The voltages and currents are subject to Kirchhoff's and Ohm's laws. The problem is to minimize the energy losses while respecting some bound constraints on the voltages and injection of power at the nodes. In the discussion below, we use some of the definitions given in the notation section. We may write Ohm's law as

$$I_{k\ell} = Y_{k\ell}(V_k - V_\ell), \quad 0 \le k \ne \ell \le n. \tag{1}$$

The matrix Y is symmetric with a zero diagonal and nonnegative elements. We may interpret a zero value of $Y_{k\ell}$ as the absence of a line between nodes k and ℓ. We assume that the network is *connected* in the sense that any two nodes can be linked by a path consisting of lines with positive values of Y_k. The current J_k injected at node k satisfies Kirchhoff's law

$$J_k = \sum_{\ell=0}^{n} I_{k\ell} = \sum_{\ell=0}^{n} Y_{k\ell}(V_k - V_\ell). \tag{2}$$

Thus the power injected at node k is

$$P_k := J_k V_k = V_k \sum_{\ell=0}^{n} Y_{k\ell}(V_k - V_\ell). \tag{3}$$

Whenever $V_k \neq 0$, equation (3) is equivalent to what we call the *power equation at node k*

$$\sum_{\ell=0}^{n} Y_{k\ell}(V_k - V_\ell) - \frac{P_k}{V_k} = 0. \tag{4}$$

Let Z be the $(n+1) \times (n+1)$ impedance matrix, defined by

$$Z_{kk} := \sum_{\ell=0}^{n} Y_{k\ell}; \quad Z_{k\ell} := -Y_{k\ell}, \quad 0 \leq k \neq \ell \leq n.$$

We note that Z is positive semidefinite since

$$V^t Z V = \sum_{0 \leq k \neq \ell \leq n} Y_{k\ell}(V_k - V_\ell)^2 \geq 0$$

is the sum of the loss of power over all lines linking nodes of $\mathcal{S} \cup \{0\}$. We may write the power equation over nodes of $\mathcal{S}$ as[1]

$$ZV - \frac{P}{V} = 0 \text{ over } \mathcal{S}. \tag{5}$$

(By "over $\mathcal{S}$" we mean that we take the restriction of the $(n+1)$-dimensional equality to nodes of $\mathcal{S}$.)

We will refer to (5) as the *power equation.* Reference [4] studies (5) for the case where the value of either V or P is given at each node. One of the results of [4] is that (5) may have multiple solutions. For instance, denoting by $\mathcal{S}_P$ the set of nodes over which P is fixed when $P > 0$ over $\mathcal{S}_P$ and $\mathcal{S}_P \overset{\neq}{\subset} \mathcal{S}$ then there exist exactly $2^{|\mathcal{S}_P|}$ solutions, each of them being associated with a convention of sign of the components of V over $\mathcal{S}_P$. In the sequel, we limit ourselves to the study of the solutions close to a certain nominal value.

We consider the problem of minimizing the losses in which $(Z, V^R, V^\flat, V^\sharp, P^\flat, P^\sharp)$ are given parameters:

$$(\mathcal{P}) \qquad \begin{array}{l} \underset{V,P}{\text{Min}} \, \frac{1}{2} V^t Z V; \quad ZV - \frac{P}{V} = 0 \text{ over } \mathcal{S}; \;\; V_0 = V^R; \\ \qquad\qquad V^\flat \leq V \leq V^\sharp; \;\; P^\flat \leq P \leq P^\sharp. \end{array}$$

In this problem $V^R \in \mathbb{R}_{+*}$ (the set of positive real numbers) is the reference value for the voltage and $V^\flat$, $V^\sharp$, $P^\flat$, and $P^\sharp$ are given n-dimensional vectors that satisfy $V^\flat \leq V^\sharp$ and $P^\flat \leq P^\sharp$. The bound constraints for V and P are understood for indices 1 to n. The components of the lower bounds $V^\flat$ and $P^\flat$ have values in $\mathbb{R} \cup \{-\infty\}$, whereas those of the upper bounds $P^\sharp$ and $V^\sharp$ have values in $\mathbb{R} \cup \{\infty\}$. An infinite value simply means an absence of lower or upper bound constraint.

[1] The division of vectors, as well as their multiplication, is to be understood componentwise.

It seems difficult to determine if the above problem is well posed and stable with respect to perturbations. First, the power equation is not itself well posed, unless we make specific assumptions about the data. In addition, some conditions on the bounds should be added in order to make them compatible with the power equation. We conclude that, to be able to conduct an analysis of this problem, we have to make some assumptions. Because some of the real-world networks have very high voltage values, a natural possibility is to consider the square root of the inverse of the reference value of the voltage as a small parameter and to let this small parameter go to zero.

3. The very high voltage approximation. We introduce the *very high voltage approximation* by embedding $(\mathcal{P})$ in the family of problems

$$(\mathcal{P}_\varepsilon) \qquad \begin{array}{l} \displaystyle\operatorname*{Min}_{V,P} \frac{1}{2} V^t Z V; \quad ZV - \frac{P}{V} = 0 \text{ over } \mathcal{S}; \quad V_0 = \frac{1}{\sqrt{\varepsilon}}; \\ \displaystyle\frac{V^\flat}{\sqrt{\varepsilon}} \leq V \leq \frac{V^\sharp}{\sqrt{\varepsilon}}; \quad P^\flat \leq P \leq P^\sharp, \end{array}$$

where $\varepsilon > 0$ is a small parameter. Our aim is to compute an asymptotic expansion of the solution of the above problem when $\varepsilon \to 0$ (we will see in a moment that for technical reasons it is more convenient to introduce the square root of the inverse of the nominal value as a small parameter rather than the inverse of the nominal value). It is also convenient to make the following change of variables:

$$\tilde{V} := \sqrt{\varepsilon} V; \; \tilde{P} := \varepsilon P.$$

Since the cost function is nonnegative and positively homogeneous of degree 2, an equivalent problem obtained after this change of variables is

$$(\tilde{\mathcal{P}}_\varepsilon) \qquad \begin{array}{l} \displaystyle\operatorname*{Min}_{\tilde{V},\tilde{P}} \frac{1}{2} \tilde{V}^t Z \tilde{V}; \quad Z\tilde{V} - \frac{\tilde{P}}{\tilde{V}} = 0 \text{ over } \mathcal{S}; \quad \tilde{V}_0 = 1; \\ V^\flat \leq \tilde{V} \leq V^\sharp; \quad \varepsilon P^\flat \leq \tilde{P} \leq \varepsilon P^\sharp. \end{array}$$

We call $(\tilde{V}^\varepsilon, \tilde{P}^\varepsilon)$ a possible solution of $(\tilde{\mathcal{P}}_\varepsilon)$. We observe that, except for the bound constraints on $\tilde{P}$ in which ε appears, this new problem is identical to $(\mathcal{P})$. From a mathematical point of view, it is equivalent to either making the voltage go to infinity with a fixed range of delivered power or making the power go to zero with a fixed range of voltage. In other words, the high voltage approximation is nothing but a small power approximation.

By *elementary case* we mean the following situation:[2] $V^\flat = -\infty$ and $V^\sharp = +\infty$ over the network; i.e., there are no bound constraints on the voltage. This is a simple situation for which various hypotheses can be easily checked.

The limit problem, obtained for $\varepsilon = 0$, is

$$(\tilde{\mathcal{P}}_0) \qquad \begin{array}{l} \displaystyle\operatorname*{Min}_{\tilde{V},\tilde{P}} \frac{1}{2} \tilde{V}^t Z \tilde{V}; \quad Z\tilde{V} - \frac{\tilde{P}}{\tilde{V}} = 0 \text{ over } \mathcal{S}; \quad \tilde{V}_0 = 1; \\ V^\flat \leq \tilde{V} \leq V^\sharp; \quad 0P^\flat \leq \tilde{P} \leq 0P^\sharp, \end{array}$$

where for writing the bound constraints on $\tilde{P}$, we use the convention

$$0 \times (-\infty) = -\infty, \;\; 0 \times (+\infty) = +\infty,$$

[2] Sometimes we write an equality between a vector and a scalar value: this means that each component of the vector is equal to the scalar value.

and $0P^\flat$ (resp., $0P^\sharp$) is the n-dimensional vector with kth component $0 \times P_k^\flat$ (resp., $0 \times P_k^\sharp$). The physical motivation for designing high voltage networks is to reduce the currents, hence the power lost along the lines. In order to achieve this, the bound constraints on the voltage must not forbid values that are close to the reference value, so that we assume that $V^\flat \leq \mathbf{1} \leq V^\sharp$, where $\mathbf{1}$ is a vector of ones whose dimension is determined by the context. Also, if $k \in \mathcal{S}$ is such that $V_k^\flat = 1 = V_k^\sharp$, then we may identify node k with node 0. Consequently, we assume that

$$(H1) \qquad V^\flat \leq \mathbf{1} \leq V^\sharp \quad \text{and} \quad V^\flat < V^\sharp,$$

where the strict inequality between vectors means strict inequality between all components.

Lemma 3.1. *Assume that* $(H1)$ *holds. Then the limit problem has a unique solution* $(\tilde{V}^0, \tilde{P}^0)$ *defined as follows:*

$$\tilde{V}^0 = \mathbf{1} \textit{ and } \tilde{P}^0 = 0.$$

Proof. We first check that $(\tilde{V}^0, \tilde{P}^0)$ defined as above is feasible for $(\tilde{\mathcal{P}}_0)$. As $\tilde{V}^0$ is constant over the network, we have $Z\tilde{V}^0 = 0$. Because $\tilde{V}^0 = \mathbf{1} > 0$, the term $\tilde{P}^0/\tilde{V}^0$ is well defined and has value 0. Therefore the power equation is satisfied. The bound constraints are also satisfied, thanks to $(H1)$.

Now $(\tilde{V}^0, \tilde{P}^0)$ is associated with a zero cost. The cost function being nonnegative, $(\tilde{V}^0, \tilde{P}^0)$ is a solution of $(\tilde{\mathcal{P}}_0)$. Any other solution (V, P) is also associated with a zero cost and hence must satisfy $ZV = 0$. It follows that $V = \tilde{V}^0$, and we deduce from the power equation that $P = \tilde{P}^0$ as well. □

4. Mathematical tools.

This section presents some mathematical tools that we need from the perturbation theory for nonlinear programming. It is devoted to the presentation of two known results and to the derivation of a third one. The first result gives the second-order expansion of the cost and the first-order expansion of the solution under weak hypotheses. The second one needs much stronger hypotheses but gives the analytic expansion of the cost and solution. Combining these two results, we obtain the analytic expansion of the cost and solution under weaker hypotheses than for the second result.

We consider an abstract finite-dimensional nonlinear optimization problem

$$(P_\varepsilon) \qquad \operatorname*{Min}_{x \in \mathbb{R}^n} f(x, \varepsilon); \quad g_i(x, \varepsilon) = 0, \ i = 1, \ldots, q; \quad g_i(x, \varepsilon) \leq 0, \ i = q + 2, \ldots, p,$$

where $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^p$ are C^2 mappings. We consider (P_0) as the unperturbed problem and $\varepsilon \in \mathbb{R}_+$ as the perturbation parameter. Our aim is to compute the expansion of the cost and solution of the problem.

We assume in this section that (P_0) has a unique solution x^0 and that for $\varepsilon > 0$ small enough, the set of solutions of (P_ε) is nonempty and uniformly bounded. This kind of condition can be checked by ad hoc conditions on specific examples, as is the case in the optimal power flow problem (see Lemma 6.1).

We denote the set of active inequality constraints, the Lagrangian function associated with (P_ε), and the set of Lagrange multipliers associated with x^0 as, respectively,

$$\begin{aligned} I(x, \varepsilon) &:= \{j = q + 1, \ldots, p; \quad g_j(x, \varepsilon) = 0\}, \\ L(x, \lambda, \varepsilon) &:= f(x, \varepsilon) + \lambda^t g(x, \varepsilon), \\ \Lambda &:= \{\lambda \in \mathbb{R}^p; \quad L'_x(x^0, \lambda, 0) = 0; \quad \lambda_j \geq 0, \ \lambda_j g_j(x^0, 0) = 0, \ j > q\}. \end{aligned}$$

An important constraint qualification, due to Mangasarian and Fromovitz [12], is the following:

$$(MF) \qquad \begin{cases} \text{(i)} \quad \{\nabla_x g_i(x,0)\},\ i=1,\dots,q \text{ are linearly independent,} \\[2ex] \text{(ii)} \quad \exists d \in \mathbb{R}^n \ \begin{cases} g_i'(x,0)(d,0)=0,\ i=1,\dots,q; \\ g_i'(x,0)(d,0)<0,\ i\in I(x,0). \end{cases} \end{cases}$$

This constraint qualification holds if and only if the set of Lagrange multipliers is nonempty and bounded (Gauvin [9]) and is also equivalent to a certain stability property of the feasible set (Robinson [13]). It therefore seems natural to assume that the constraint qualification holds in order to conduct a perturbation analysis. However, in the specific application we have in mind, (MF) is not satisfied. Consequently we will rely on the *directional constraint qualification* due to Gollan [11]

$$(DQ) \qquad \begin{cases} \text{(i)} \quad \{\nabla_x g_i(x,0)\},\ i=1,\dots,q \text{ are linearly independent,} \\[2ex] \text{(ii)} \quad \exists d \in \mathbb{R}^n \ \begin{cases} g_i'(x,0)(d,1)=0,\ i=1,\dots,q; \\ g_i'(x,0)(d,1)<0,\ i\in I(x,0). \end{cases} \end{cases}$$

It is easily checked that (MF) implies (DQ).

Now consider the linearization of the data with respect to (x,ε) at $(x^0,0)$. If $(d,1)$ denotes the direction, we get the *linearized problem*

$$(\hat{\mathcal{L}}) \qquad \begin{cases} \operatorname*{Min}_{d\in\mathbb{R}^n} f'(x^0,0)(d,1); \quad & g_i'(x,0)(d,1)=0,\ i=1,\dots,q; \\ & g_i'(x,0)(d,1)\le 0,\ i\in I(x,0). \end{cases}$$

If (DQ) holds, then $(\hat{\mathcal{L}})$ is feasible. A nice interpretation of condition (DQ) is that it is equivalent to the condition of Mangasarian and Fromovitz for the linearized problem. It follows from (DQ) that the (standard) dual of $(\hat{\mathcal{L}})$, whose expression is

$$(\mathcal{D}) \qquad \operatorname*{Max}_{\lambda} L_\varepsilon'(x^0,\lambda,0)\ ;\ \lambda\in\Lambda,$$

has, if Λ is nonempty, a nonempty and bounded set of solutions $S(\mathcal{D})$. In this case $S(\hat{\mathcal{L}})$ is nonempty. In short, if (DQ) holds and Λ is nonempty, then both $S(\hat{\mathcal{L}})$ and $S(\mathcal{D})$ are nonempty and $S(\mathcal{D})$ is bounded.

We need some second-order analysis. The *critical cone* is defined as

$$C(x) := \quad \{d\in\mathbb{R}^n;\ f_x'(x,0)d\le 0;\quad \begin{array}{l} g_i'(x,0)(d,0)=0,\ i=1,\dots,q; \\ g_i'(x,0)(d,0)\le 0,\ i\in I(x,0)\}. \end{array}$$

The *directional second-order condition*, due to Shapiro [15], is as follows:

$$\sup_{\lambda\in S(\mathcal{D})} L_{x^2}''(x^0,\lambda,0)dd>0\ \forall d\in C(x^0)\backslash\{0\}.$$

Note that, the supremum over an empty set being $-\infty$, the directional second-order condition implies that the set of multipliers is nonempty (except in the special case $C(x^0)=\{0\}$, but it is also true in this case that the set of multipliers is nonempty).

We say that a map $\mathbb{R}_+\to\mathbb{R}^n$, $\varepsilon\to x^\varepsilon$, is a *path* if $x^\varepsilon\to x^0$ when $\varepsilon\downarrow 0$ (we do not require continuity of $\varepsilon\to x^\varepsilon$). A path of $o(\varepsilon^2)$ solutions is a path x^ε such that, for $\varepsilon > 0$ small enough, x^ε is feasible for (P_ε) and $f(x^\varepsilon, \varepsilon) \leq v(P_\varepsilon) + o(\varepsilon^2)$. We say that $d \in \mathbb{R}^n$ is a *first-order term* associated with a path x^ε if d is a limit point of $(x^\varepsilon - x^0)/\varepsilon$. We define the following subproblem

$$(Q) \qquad \operatorname*{Min}_{d \in S(\hat{\mathcal{L}})} \max_{\lambda \in S(\mathcal{D})} L''(x^0, \lambda, 0)(d, 1)(d, 1).$$

As $S(\hat{\mathcal{L}})$ is a polyhedron, if $S(\mathcal{D})$ is a singleton then (Q) is a quadratic optimization problem (a problem with quadratic cost and linear constraints).

The result below, due to Bonnans, Ioffe, and Shapiro [6], has its origins in the work of Shapiro [15] and Auslender and Cominetti [2].

THEOREM 4.1. *Assume that the directional constraint qualification hypothesis is satisfied, as well as the directional second-order condition. Then*

(i) *(Stability) Any path x^ε of $o(\varepsilon^2)$ solutions satisfies $x(\varepsilon) = x^0 + O(\varepsilon)$.*

(ii) *(Expansion of solutions) The union of all first-order terms associated with $o(\varepsilon^2)$ solutions is equal to $S(Q)$. In particular, if $S(Q) = \{\overline{d}\}$, then any path of* exact *solutions x^ε satisfies $x^\varepsilon = x^0 + \varepsilon \overline{d} + o(\varepsilon)$.*

(iii) *With any solution of (P_ε), there is associated, for ε small enough, a nonempty and uniformly bounded set of multipliers. The set of limit points of these multipliers (when $\varepsilon \downarrow 0$) is included in $S(\mathcal{D})$.*

We now state a second abstract result based on stronger hypotheses. The condition of *linear independence* (of gradients of active constraints at x^0) is

$$(LI) \qquad \{\nabla_x g_i(x^0, 0);\ i \in \{1, \dots, q\} \cup I(x^0, 0)\} \text{ is linearly independent.}$$

This condition implies that x^0 is associated with a unique multiplier λ^0. We define the *enlarged critical cone* (which is a vector subspace) as

$$C^\sharp(x^0, \lambda^0) := \{d \in \mathbb{R}^n;\ g_i'(x^0, 0)(d, 0) = 0,\ i \in \{1, \dots, q\} \cup \{j \in I(x^0, 0);\ \lambda_j^0 > 0\}\}.$$

It is easily checked that $C(x^0) \subset C^\sharp(x^0)$. Both sets coincide if (x^0, λ^0) is a strictly complementary pair in the sense that $\lambda_j^0 > 0$ whenever $j \in I(x^0, 0)$. The *strong second-order condition*, due to Robinson [14], assumes that a unique multiplier λ^0 is associated with x^0 such that

$$(6) \qquad L''_{x^2}(x^0, \lambda^0, 0)dd > 0 \;\forall d \in C^\sharp(x^0, \lambda^0) \backslash \{0\}.$$

The theorem below is due to Bonnans and Sulem [8].

THEOREM 4.2. *Assume that (x^0, λ^0) satisfy the condition of linear independence as well as the strong second-order condition. Then in a certain neighborhood of x^0 for $\varepsilon > 0$ small enough, (P_ε) has a unique solution x^ε and the mapping $\varepsilon \to (x^\varepsilon, \lambda^\varepsilon, v(P_\varepsilon))$ is (real) analytic. The coefficients of the expansion of x^ε and λ^ε can be computed by expanding the optimality system as in* [8]. *The first-order expansion is $x^\varepsilon = x^0 + \varepsilon \overline{d} + O(\varepsilon^2)$, where $\overline{d}$ is the unique solution of $S(Q)$.*

We now combine the above two results in order to deduce a third. We say that the *directional linear independence qualification condition* holds if $(\hat{\mathcal{L}})$ is feasible and satisfies the hypothesis of linear independence at each solution of (Q). We say that the *strong directional second-order condition* holds if $(\mathcal{D})$ has a unique solution λ^0, such that (6) holds.

THEOREM 4.3. *Assume that x^0 is a local solution of (P_0) satisfying*

(i) *the directional linear independence qualification condition, and*

(ii) *the strong directional second-order condition.*
Then the conclusion of Theorem 4.2 *holds.*

Proof. By (i), the linearized problem is qualified, i.e., (DQ) holds. Condition (ii) implies the directional second-order condition (which therefore involves only the multiplier λ^0). We may apply Theorem 4.1. Problem (Q) consists of minimizing a quadratic cost over $S(\hat{\mathcal{L}})$. By (ii), the Hessian of the cost function is positive definite over the vector space parallel to the affine hull of $S(\hat{\mathcal{L}})$ (note that as λ^0 solves the dual of $(\hat{\mathcal{L}})$, the constraints associated with nonzero components of λ^0 are active at any solution of $(\hat{\mathcal{L}})$). Therefore problem (Q) has a unique solution $\overline{d}$. Let us denote by I^* the set of active inequality constraints for $(\hat{\mathcal{L}})$ associated with $\overline{d}$. Let $\{x^\varepsilon\}$ be a path of solutions. The inequality constraints in $I\backslash I^*$ are not active for $\varepsilon > 0$ small enough. Therefore, x^ε is (for positive ε) a local solution of

$$(\hat{P}_\varepsilon) \qquad \operatorname*{Min}_{x\in\mathbb{R}^n} f(x,\varepsilon);\ g_i(x,\varepsilon)=0,\ i=1,\dots,q;\ g_i(x,\varepsilon)\le 0, i\in I^*.$$

Now x^0 is a local solution of $(\hat{P}_0)$ associated with a unique multiplier λ^0 (as can be checked with the second-order sufficient conditions), and we may apply Theorem 4.2 to problem $(\hat{P}_0)$ in order to get the conclusion. □

5. Study of the limit problem.

Let us return to the optimal power flow problem. We observe that the limit problem is in general not qualified in the sense that (MF) does not hold. Indeed, if there exists $k \in \mathcal{S}$ such that both $P_k^\flat$ and $P_k^\sharp$ have finite but different values, then we get the constraint $0 \le \tilde{P}_k \le 0$, and these two inequalities cannot be strictly satisfied.

We need the following notation for the active constraints: $\bar{V}^\flat$, $\bar{V}^\sharp$ are n-dimensional vectors related to the active constraints such that for all $k \in \mathcal{S}$,

$$\bar{V}_k^\flat = \begin{cases} V_k^\flat & \text{if } V_k^\flat = 1 \\ -\infty & \text{otherwise.} \end{cases} \quad \text{and} \quad \bar{V}_k^\sharp = \begin{cases} V_k^\sharp & \text{if } V_k^\sharp = 1 \\ +\infty & \text{otherwise.} \end{cases}$$

(Recall that $\tilde{V}^0 = \mathbf{1}$.)

Linearizing the data of $(\tilde{\mathcal{P}}_\varepsilon)$ with respect to $(\tilde{V}, \tilde{P}, \varepsilon)$ at $(\tilde{V}^0, \tilde{P}^0, 0)$ we obtain the *linearized problem*

$$(\mathcal{L}) \qquad \left\{ \begin{array}{l} \operatorname*{Min}_{dV,dP} (\tilde{V}^0)^t Z dV; \quad ZdV = dP/\tilde{V}^0 \text{ over } \mathcal{S};\ dV_0 = 0; \\ \\ \qquad\qquad\qquad \bar{V}^\flat \le \mathbf{1} + dV \le \bar{V}^\sharp;\ \ P^\flat \le dP \le P^\sharp. \end{array} \right.$$

As $\tilde{V}^0$ is constant, we have $(\tilde{V}^0)^t Z dV = (Z\tilde{V}^0)^t dV = 0$. The equality $S(\mathcal{L}) = F(\mathcal{L})$ follows. In the lemma below, we use hypothesis $(H1)$ defined just before Lemma 3.1. Denote by $\mathcal{S}_{\tilde{P}}^{=}$ the set of nodes over which the lower and upper bound on $\tilde{P}$ are equal. In order to apply the theoretical result, we view the bound constraints on $\tilde{P}$ over $\mathcal{S}_{\tilde{P}}^{=}$ as equalities over $\mathcal{S}_{\tilde{P}}^{=}$.

LEMMA 5.1. *Assume that* $(H1)$ *holds. Then*

(i) *the gradients of the equality constraints are linearly independent*;

(ii) *the limit problem is directionally qualified if and only if there exists* (dV, dP) *in* $F(\mathcal{L})$ *with each component strictly between the bounds whenever they are not equal*;

(iii) *if the limit problem is directionally qualified, then* $(\mathcal{D})$ *has a unique solution, namely the zero multiplier.*

Proof. (i) Checking (i) amounts to checking that the linearized equality constraints are onto, i.e., checking that the system

$$dV_0 = W^0;\ ZdV - \frac{dP}{V} = W^1 \text{ over } \mathcal{S};\ dP = W^2 \text{ over } \mathcal{S}^{=}_{\bar{P}},$$

has at least one solution for any (W^0, W^1, W^2). Set dP to 0 over $\mathcal{S} \backslash \mathcal{S}^{=}_{\bar{P}}$. With respect to V, it remains to solve the system

$$dV_0 = W^0;\ ZdV = \frac{dP}{V} + W^1 \ \text{over}\, \mathcal{S},$$

which is the equation of a linear DC network with given voltage at node 0 and given inputs of current at other nodes. This problem has a unique solution.

(ii) This is an immediate consequence of part (i) and the definition of (DQ).

(iii) As the gradient of the cost of problem $\tilde{\mathcal{P}}_0$ is zero, the set of Lagrange multipliers is a cone. Problem $(\mathcal{D})$ consists of maximizing a linear cost over this cone. It follows that $S(\mathcal{D})$ is itself a cone. We know that if directional qualification holds, then the set of solutions is nonempty and bounded. Being a cone, this set can be nothing else than $\{0\}$. □

Remark. In the elementary case (i.e., when the voltage is unconstrained) Lemma 5.1(i) is satisfied. Therefore, the limit problem is directionally qualified. Another situation where directional qualification can be checked is when the upper bounds on the voltage are strictly greater than 1, and there is no upper bound on the power. It follows that problem $(\mathcal{L})$ has no upper bound. Now we can take $dP = \alpha(1, \dots, 1)^t$ with $\alpha > 0$ large enough to obtain $dP > P^\flat$. Then dV, the solution of the linearized power equation, is *positive*; hence, (dV, dP) satisfies the condition for directional qualification.

We may have a better insight into these qualification conditions by making an analogy with the optimal control theory. See dP as the control and dV as the state. What we call the elementary case occurs when there is no state constraint. In the case of one-sided constraints for the control and state, where we have taken advantage of the positivity of the mapping "control → state," the discussion parallels the one for the control of nonlinear elliptic equations (see, e.g., Bonnans and Casas [5]).

We assume in the sequel that the directional qualification hypothesis holds. As $S(\mathcal{D}) = \{0\}$, the cost of the quadratic subproblem reduces in our application to the Hessian of the cost. Using $S(\mathcal{L}) = F(\mathcal{L})$, we can reformulate this problem as

$$(SP) \qquad \operatorname*{Min}_{dV, dP} \frac{1}{2}(dV)^t Z dV;\ (dV, dP) \in F(\mathcal{L}).$$

Let Z^R be obtained by deleting the first row and column of Z. As the network is connected, Z^R is invertible. Because dV_0 is set to 0, the system $ZdV = dP/\tilde{V}^0$ has a unique solution $dV = (Z^R)^{-1}(dP/\tilde{V}^0)$. Introducing the auxiliary variable $h := dP/\tilde{V}^0$, and substituting in (SP), we obtain the equivalent problem

$$\operatorname*{Min}_{h} \frac{1}{2} h^t (Z^R)^{-1} h;\ dV_0 = 0;\ \bar{V}^\flat \leq (Z^R)^{-1} h \leq \bar{V}^\sharp;\ P^\flat \leq h \leq P^\sharp.$$

This problem has a unique solution, namely the projection, along the norm associated with $(Z^R)^{-1}$, of the origin over the feasible set. We call $(\overline{dV}, \overline{dP})$ the solution of (SP).

The following lemma allows us to check the second-order conditions.

Lemma 5.2. *Consider the limit problem* $(\tilde{\mathcal{P}}_0)$*. Then the Hessian of the cost is positive definite over the space tangent to the equality constraints.*

Proof. The quadratic form $(dV)^t Z dV$ is positive semidefinite, and its null space is the kernel of Z, which is known to be the space of vectors with all equal components. As $dV_0 = 0$, it follows that $(dV)^t Z dV > 0$ unless dV is zero. In the latter case, set dP to zero whenever the linearized power equation is satisfied. As a consequence, if (dV, dP) is tangent to the linearized equality constraints, then the quadratic form associated with the Hessian of the cost is positive unless $(dV, dP) = 0$, as was to be proved. □

6. Expansion of the solution and multipliers. We start with a technical lemma allowing us to check some of the hypotheses of the results of section 4.

Lemma 6.1. *The set of solutions of* $(\tilde{\mathcal{P}}_\varepsilon)$ *for* ε *small enough is nonempty and uniformly bounded.*

Proof. By directional qualification, we know that

$$v(\tilde{\mathcal{P}}_\varepsilon) \leq v(\tilde{\mathcal{P}}_0) + \varepsilon v(\mathcal{L}) + o(\varepsilon) \leq O(\varepsilon).$$

Due to the condition $\tilde{V}_0 = \mathbf{1}$, the cost function of $v(\tilde{\mathcal{P}}_\varepsilon)$ satisfies for some $\alpha > 0$ and any feasible $\tilde{V}$ the relation $\frac{1}{2}\tilde{V}^t Z \tilde{V} \geq \alpha \|\tilde{V} - \mathbf{1}\|^2$. Denoting by "." the componentwise multiplication of vectors, we have $\|P\| = \|\tilde{V}.Z\tilde{V}\| = O(\|\tilde{V} - \mathbf{1}\|^2)$, where from these relations we deduce for ε small enough the uniform boudedness of minimizing sequences. In addition, if $(\tilde{V}^k, \tilde{P}^k)$ is a minimizing sequence, then from $\|\tilde{V}^k - \mathbf{1}\|^2 \leq O(\varepsilon)$ we deduce (for k large enough and ε small enough) that the limit points of $\tilde{V}^k$ have positive values. Consequently, these limit points are solutions of $(\tilde{\mathcal{P}}_\varepsilon)$. As the same estimates hold for the limit points, the conclusion follows. □

Theorem 6.2. *Assume that* $(H1)$ *and* $(\tilde{\mathcal{P}}_0)$ *are directionally qualified. Then, for* ε *small enough,* $(\tilde{\mathcal{P}}_\varepsilon)$ *has a unique solution* $(\tilde{V}^\varepsilon, \tilde{P}^\varepsilon)$ *and the following expansions hold:*

$$v(\tilde{\mathcal{P}}_\varepsilon) = v(\tilde{\mathcal{P}}_0) + \varepsilon^2 v(SP) + o(\varepsilon^2),$$

$$(\tilde{V}^\varepsilon, \tilde{P}^\varepsilon) = (\tilde{V}^0, \tilde{P}^0) + \varepsilon(\overline{dV}, \overline{dP}) + o(\varepsilon).$$

Associated with $(\tilde{V}^\varepsilon, \tilde{P}^\varepsilon)$ *is a nonempty and uniformly bounded set of multipliers* Λ_ε*. This set of multipliers converges to* 0 *in the sense that if* $\lambda^\varepsilon \in \Lambda_\varepsilon$ *and* $\varepsilon \downarrow 0$ *then* $\lambda^\varepsilon \to 0$*.*

Proof. Apply Theorem 4.1, using Lemmas 5.1, 5.2, and 6.1. □

We now study the higher-order expansion of the solution of $(\tilde{\mathcal{P}}_\varepsilon)$ and its associated multipliers. We observe that the result of Bonnans and Sulem [8] cannot be used directly because problem $(\tilde{\mathcal{P}}_0)$ is not in general qualified, as already observed. However, we may apply Theorem 4.3, if the hypothesis below holds.

$(H2)$ Linear independence of gradients of active constraint for $(\mathcal{L})$ at $(\overline{dV}, \overline{dP})$.

Let us state first a natural sufficient condition for $(H2)$.

Lemma 6.3. *Assume that the bound constraints on* $\overline{dV}$ *and* $\overline{dP}$ *are never simultaneously active at a node of the network. Then* $(H2)$ *holds.*

Proof. Let the bound constraints on the voltage (resp., power) be active on $\mathcal{S}_V^a$ (resp., $\mathcal{S}_P^a$). By hypothesis, $\mathcal{S}_V^a \cap \mathcal{S}_P^a = \phi$. Set dP to 0 over $\mathcal{S} \backslash (\mathcal{S}_V^a \cup \mathcal{S}_P^a)$. We must check that it is possible to solve a linear DC problem with voltage fixed over $\mathcal{S}_V^a$ as well as the reference node and the injection of current fixed over the other nodes. This problem is known to have a unique solution. □

THEOREM 6.4. *Assume that* $(H1)$ *and* $(H2)$ *hold. Then the mapping* $\varepsilon \to (\tilde{V}^\varepsilon, \tilde{P}^\varepsilon, \lambda^\varepsilon, v(\tilde{\mathcal{P}}_\varepsilon))$, *where* $(\tilde{V}^\varepsilon, \tilde{P}^\varepsilon) \in S(\tilde{\mathcal{P}}_\varepsilon)$ *and* λ^ε *is the multiplier associated with the power equation, is real analytic for small enough* ε.

Proof. By $(H2)$, Lemmas 5.1, 5.2, and 6.1, the hypotheses of Theorem 4.3 are satisfied. The conclusion follows. □

7. Back to the physical problem.

In this section we return to the physical variables $(V^\varepsilon, P^\varepsilon)$ and give simple interpretations of the above results. Restating Theorem 6.2, we obtain the following theorem.

THEOREM 7.1. *Assume that* $(\tilde{\mathcal{P}}_0)$ *is directionally qualified. Then, for* ε *small enough,* $(\mathcal{P}_\varepsilon)$ *has a unique solution* $(V^\varepsilon, P^\varepsilon)$, *and the following expansions hold:*

$$v(\mathcal{P}_\varepsilon) = \varepsilon v(SP) + o(\varepsilon),$$

$$V^\varepsilon = \frac{1}{\sqrt{\varepsilon}}\mathbf{1} + \sqrt{\varepsilon}\,\overline{dV} + o(\sqrt{\varepsilon}),$$

$$P^\varepsilon = \overline{dP} + o(\varepsilon).$$

In particular, we deduce that if the nominal value is very high and if the bound constraints are compatible, in the sense that $(H1)$ holds, then

(a) the loss of power is of the order of the square of the inverse of the average value of the network,

(b) the difference between the average value and the actual value of the voltage is of the order of the inverse of the average value, and

(c) the distribution of power over the network has a limit.

Concluding remark. There are two possible uses of our technical results. The first one is the above set of *qualitative* remarks (a) to (c), which are of interest by themselves. The second possibility is to use the first-order expansion as the starting point of a numerical algorithm, dedicated to a *quantitative* resolution of the optimal power flow problem. Still, the most important aspect of the result is that it suggests a possible extension to the AC power flow problem, whose importance was stressed in the introduction.

Acknowledgments. The author would like to thank the referees for their useful remarks. Thanks are also due to G. Blanchon and J.C. Dodu, from Direction des Etudes et Recherches, EDF, for the numerous discussions I have had with them for many years.

REFERENCES

[1] J. P. AUBIN AND P. A. RAVIART, *On the resolution of equations arising in load flow problems*, in Proc. IEEE Power Industry Computer Applications Conf., 1965, pp. 119–132.

[2] A. AUSLENDER AND R. COMINETTI, *First and second order sensitivity analysis of nonlinear programs under directional constraint qualification conditions*, Optimization, 21 (1990), pp. 351–363.

[3] G. BLANCHON, J. F. BONNANS, AND J. C. DODU, *Optimisation de réseaux électriques de grande taille*, Lect. Notes Inf. Cont. Sci. 144, Springer-Verlag, Berlin, 1990, pp. 423–431.

[4] J. F. BONNANS, *Intégration de la méthode CRIC dans l'optimiseur OPSYC*, Rapport de Contrat EDF-INRIA, INRIA, Rocquencourt, France, 1992.

[5] J. F. Bonnans and E. Casas, *Contrôle de systèmes elliptiques semilinéaires comportant des contraintes sur l'état*, in Nonlinear Partial Differential Equations and Their Applications, Collège de France seminar Vol. VIII, H. Brézis and J. L. Lions, eds., Pitman Research Notes in Mathematics Series 166, Longman Scientific and Technical, New York, 1988, pp. 69–86.

[6] J. F. Bonnans, A. D. Ioffe, and A. Shapiro, *Développement de solutions exactes et approchées en programmation non linéaire*, Comptes Rendus Acad. Sci. Paris, t. 315, Série I, (1992), pp. 119–123.

[7] J. F. Bonnans and A. Shapiro, *Optimization Problems with Perturbations, A Guided Tour*, Rapport de Recherche INRIA 2872, INRIA, Rocquencourt, France, April 1996.

[8] J. F. Bonnans and A. Sulem, *Pseudopower expansion of solutions of generalized equations and constrained optimization problems*, Math. Programming, 70 (1995), pp. 123–148.

[9] J. Gauvin, *A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming*, Math. Programming, 12 (1977), pp. 136–138.

[10] J. Gauvin and R. Janin, *Directional behavior of optimal solutions in nonlinear mathematical programming*, Math. Oper. Res., 13 (1988), pp. 629–649.

[11] B. Gollan, *On the marginal function in nonlinear programming*, Math. Oper. Res., 9 (1984), pp. 208–221.

[12] O. L. Mangasarian and S. Fromovitz, *The Fritz John necessary optimality conditions in the presence of equality and inequality constraints*, J. Math. Anal. Appl., 17 (1967), pp. 37–47.

[13] S. M. Robinson, *Stability theory for systems of inequalities, part* II*: Differentiable nonlinear systems*, SIAM J. Numer. Anal., 13 (1976), pp. 497–513.

[14] S. M. Robinson, *Strongly regular generalized equations*, Math. Oper. Res., 5 (1980), pp. 43–62.

[15] A. Shapiro, *Sensitivity analysis of nonlinear programs and differentiability properties of metric projections*, SIAM J. Control Optim., 26 (1988), pp. 628–645.

[16] J. F. Bonnans, *Mathematical study of very high voltage power networks* II*: The AC power flow problem*, SIAM J. Appl. Math, to appear.

[17] J. F. Bonnans, *Mathematical Study of Very High Voltage Power Networks* III: *The AC Optimal Power Flow Problem,* Rapport de Recherche INRIA, INRIA, Rocquencourt, France, 1997.

SIAM J. OPTIM.
Vol. 7, No. 4, pp. 991–1016, November 1997

007

ROBUST TRUSS TOPOLOGY DESIGN VIA SEMIDEFINITE PROGRAMMING[*]

A. BEN-TAL[†] AND A. NEMIROVSKI[†]

Abstract. We present and motivate a new model of the truss topology design problem, where the rigidity of the resulting truss with respect both to given loading scenarios and small "occasional" loads is optimized. It is shown that the resulting optimization problem is a semidefinite program. We derive and analyze several equivalent reformulations of the problem and present illustrative numerical examples.

Key words. structural optimization, truss topology design, robustness, semidefinite programming, interior point methods

AMS subject classifications. 19C25, 19C30, 19C50, 73K40

PII. S1052623495291951

1. Introduction.

Truss topology design (TTD) deals with the selection of optimal configuration for structural systems (mechanical, civil engineering, aerospace) and constitutes one of the newest and most rapidly growing fields of structural design (see the excellent survey paper by Rozvany, Bendsøe, and Kirsch [12]). The TTD problem was studied extensively, both mathematically and algorithmically, in [1, 2, 3, 4, 5].

In this paper we bring forth the issue of the *robustness* of the truss; here we say that a truss is *robust* if it is reasonably rigid with respect both to the given set of loading scenarios and to all small uncertain (in size and direction) loads, which may act at any of the *active* nodes of the truss, i.e., those which are linked at least by one bar. In the engineering literature, rigidity is modeled by considering different *loading scenarios* on the structure (the multiload TTD problem) or by imposing upper and lower bounds on nodal displacements. The first approach depends on the engineer's ability to "guess right" the relevant scenarios, while the second approach leads to a mathematical problem which is not tractable computationally. Here we suggest a new modeling approach, which circumvents both of the above mentioned difficulties.

The paper is organized as follows. Section 2 describes the modeling approach in question. The preliminary section 2.1 presents the basic notions related to the TTD problem and the traditional formulations of the problem. We demonstrate by simple example (section 2.2) that robustness restrictions (which are basically ignored in the traditional formulations) are critical to obtain reasonable designs; this observation motivates our modeling approach presented in section 2.3. Its computational tractability is demonstrated in section 2.4, where we show that the TTD problem in our new formulation can be equivalently cast as a *semidefinite program.* This brings the problem into the realm of convex programming for which efficient (polynomial time) interior point algorithms can be employed. Sections 3–5 are devoted to mathematical processing of the semidefinite program of section 2.4; the goal is to get a program better

[*]Received by the editors September 15, 1995; accepted for publication (in revised form) June 28, 1996.

http://www.siam.org/journals/siopt/7-4/29195.html

[†]Faculty of Industrial Engineering and Management at Technion Israel Institute of Technology, Technion City, Haifa 32000, Israel (morbt@ie.technion.ac.il, nemirovs@ie.technion.ac.il). This research was partly supported by the Israeli Academy of Sciences and by The Fund for Promotion of Research at the Technion.

suited for interior point algorithms. Possibilities for robust truss topology design by these algorithms are discussed in section 6. We end up (section 7) with illustrating usefulness of our approach by considering several examples of optimal trusses with and without robustness considerations. We show that at least for these examples *robustness* can be gained without sacrificing much in the optimality of the resulting trusses. Concluding section 8 contains remarks on the possibility to extend the idea of "robust reformulation" of an optimization program from the particular case of the TTD problem to other problems of mathematical programming.

2. Truss topology design with robustness constraints.

2.1. Trusses, loads, compliances. Informally, a *truss* is a 2D or 3D construction composed of thin elastic *bars* linked with each other at *nodes*—points from finite *nodal set* $\mathcal{V}$ given in advance in 2D plane (respectively, 3D) space. When subjected to a given *load*—distribution of external forces applied at the nodes—the construction deformates until the reaction forces caused by deformations of the bars compensate the external load. The deformated truss capacitates certain potential energy, and this energy, the *compliance*, measures stiffness of the truss, its ability to withstand the load; the less is compliance, the more rigid is the truss with respect to the load.

In the usual TTD problem we are given the nodal set and one (*single-load* TTD) or several (*multi-load* TTD) loads along with total volume of the bars. The displacements of some of the nodes are completely or partially fixed, so that the space R_v of virtual displacements of node v is certain linear subspace and the problem is to distribute the given volume of the truss between the bars in order to get the most rigid construction, i.e., the one which minimizes the maximal compliance over the given set of loads. Some of the bars can get zero volume, i.e., be eliminated from the resulting construction, so that in fact the topology of the construction is optimized as well (this is the origin of the term "topology design").

The mathematical formulation of the problem, in its simplest form, is as follows.

Given are

(i) graph $(\mathcal{V}, \mathcal{B})$ (ground structure) with the nodal set $\mathcal{V} \subset R^D$ $(D = 2, 3)$ composed of $\widehat{n}$ nodes and with arc set $\mathcal{B}$ of m tentative bars;

(ii) collection of linear subspaces $R_v \subset R^D$, $v \in \mathcal{V}$—the spaces of virtual displacements of the nodes.

We refer to the quantity $n = \sum_{v \in \mathcal{V}} \dim R_v$ as the number of degrees of freedom of the nodal set and call the space $R^n = \prod_{v \in \mathcal{V}} R_v$ the *space of nodal displacements.* A vector $x \in R^n$ can be naturally interpreted as collection of virtual displacements of the nodes. Similarly, a load—collection of external forces applied at the nodes – can be interpreted as a vector from R^n. (One can ignore the components of the forces orthogonal to the subspaces of virtual nodal displacements, since these components are compensated by supports restricting virtual displacements of nodes; the remaining components of the forces can be naturally assembled in a vector from R^n.)

(iii) When designing the truss, we are given a finite set $F \subset R^n$ of *loading scenarios*; the truss should be able to carry the load for each of the scenarios.

(iv) The design variables in the problem are *bar volumes* t_i, $i = 1, ..., m$; along with the nodal set $\mathcal{V}$, they completely determine the truss. We allow ourselves, for the sake of brevity, *truss* t. We are given the total volume $V > 0$ of the bars, so that the set of all admissible vectors of bar volumes is the simplex

$$T = \left\{ t \in R^m | \, t \geq 0, \ \sum_{i=1}^{m} t_i = V \right\}.$$

With the elastic model of the bars, deformation of truss accompanied by displacement $x \in R^n$ of the nodes results in the vector of reaction forces $A(t)x$, where t is the vector of bar volumes and

$$A(t) = \sum_{i=1}^{m} t_i A_i$$

is the $n \times n$ *bar-stiffness matrix* of the truss. The *bar-stiffness matrix* A_i of the ith bar is readily given by the geometry of the nodal set and involves the Young modulus of the material. What is crucial for us is that for all i,

$$A_i = b_i b_i^T \tag{2.1}$$

is a rank 1 positive semidefinite symmetric matrix (for explanations and details, see, e.g., [1, 2, 3]).

Given $t \in T$ and a load $f \in F$, one can associate with this pair the equilibrium equation

$$A(t)x = f. \tag{2.2}$$

(As was explained, x is the vector of nodal displacements caused by the load f, provided that the vector of bar volumes is t.) Solvability of this equation means that the truss is capable of carrying the load f, and if this is the case, then the *compliance*[1]

$$c_f(t) \equiv f^T x = \sup_{u \in R^n} \left[2f^T u - u^T A(t) u \right] \tag{2.3}$$

is regarded as a measure of internal work done by the truss with respect to the load f; the smaller is the compliance, the larger is the stiffness of the truss. If the equilibrium equation (2.2) for a given t is unsolvable, then it is convenient to define the compliance $c_f(t)$ as $+\infty$, which is compatible with the second equality in (2.3).

The problem of optimal minmax TTD is to find the vector of bar volumes which results in the smallest possible worst-case compliance:

($\text{TD}_{\text{minmax}}$) : *find* $t \in T$ *which minimizes the worst-case compliance* $c^F(t) = \sup_{f \in F} c_f(t)$.

From now on we assume that the problem is *well posed*, i.e., that

A. The matrix $\sum_{i=1}^m A_i$ is positive definite.

(This actually means that the supports prevent rigid body motion of the truss.)

2.2. Robustness constraint: Motivation. The "standard" case of problem ($\text{TD}_{\text{minmax}}$) is the one when F is a singleton (*single-load TTD problem*) or a finite set composed of small number (3-5) of loads (*multiload TTD problem*). An evident shortcoming of both these settings is that they do not take "full" care of the robustness of the resulting truss. The associated optimal design ensures reasonable (in fact the best possible) behavior of the truss under the loads from the list of scenarios F; it may happen, however, that a load not from this set, even a "small" one, will cause an inappropriately large deformation of the truss. Consider, e.g., the following toy example. Figure 2.1 represents a six-element nodal set with two fixed nodes ($R_v = \{0\}$) and four free nodes ($R_v = R^2$), the "ground structure"—the set of all tentative bars and the load f which is the unique element of F.

[1] The "true" compliance, as defined in mechanics, is one half of the quantity given by (2.3); we rescale the compliance in order to avoid multiple fractions $\frac{1}{2}$.

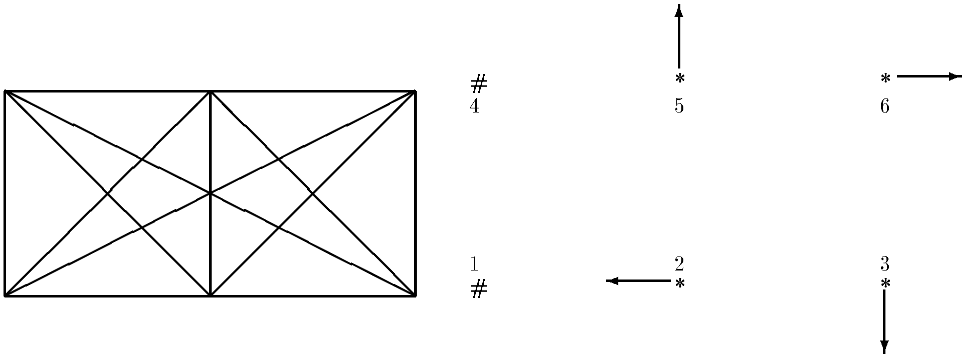


FIG. 2.1. *Ground structure and loading scenario* * *– free nodes;* # *– fixed nodes; arrows – forces.*

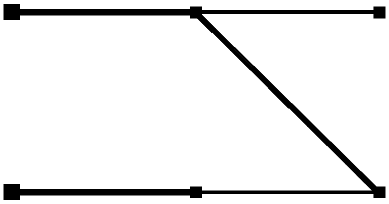


FIG. 2.2. *Optimal single-load design.*

Figure 2.2 shows the results of the usual single-load design, which results in the optimal compliance 16.000. Note that the resulting truss is completely unstable; e.g., the bar linking nodes 5 and 6 can rotate around node 5, so that arbitrarily small nonhorizontal force applied at node 6 will cause infinite compliance.

It seems that a "good" design should ensure reasonable compliances under *all* tentative loads of reasonable magnitude acting at the nodes of the resulting truss, not only "the best possible" compliance under the small list of loads in F of primary interest.

The indicated requirement can be modeled as follows. When formulating the problem, the engineer embeds a small finite set of loads $F = \{f_1, ..., f_q\}$ he is especially interested in ("primary" loads) into a "more massive" set M containing F, but also "occasional loads" of perhaps much smaller magnitude ("secondary" loads), and looks for the truss $t \in T$ which minimizes the worst-case compliance $c^M(t)$ taken with respect to this extended set M of loading scenarios.

In order to get a computationally tractable problem, in what follows we restrict ourselves to the case where M is an ellipsoid centered at the origin.[2]

$$M = QW_q \equiv \{Qe|\, e \in R^q,\, e^T e \leq 1\}.$$

Here Q is a given $n \times q$ "scale" matrix and W_q is the unit Euclidean ball in R^q. Note that we allow the case $q < n$ as well, where M is a "flat" q-dimensional ellipsoid.

The corresponding modification of $(\mathrm{TD}_{\mathrm{minmax}})$ is as follows:

[2] The only other case when the indicated problem is computationally tractable seems to be that one of a polytope M given by the list of its vertices. This case hardly deserves a special consideration, since it leads to the standard multiload TTD problem.

$(\mathrm{TD}_{\mathrm{robust}})$: *find $t \in T$ which minimizes the compliance*

$$c^M(t) = \max_{e^T e \leq 1} \max_{x \in R^n} \left[2(Qe)^T x - x^T A(t) x \right].$$

2.3. Selection of scale matrix Q. Problem $(\mathrm{TD}_{\mathrm{robust}})$ takes care of all loads $f \in M$, M being the image of the unit q-dimensional Euclidean ball under the mapping $e \mapsto Qe$. It follows that if a load $f \in M$ has a nonzero force acting at certain node l, then this node will for sure be present in the resulting construction. This observation means that we should be very careful when forming Q; otherwise we enforce incorporating into the final construction the nodes which in fact are redundant. There are two ways to meet the latter requirement.

A. We could use the indicated approach as a postoptimality analysis; after we have found the solution to the usual multiload TTD problem, given the resulting nodal structure, we can improve the robustness of the solution by solving $(\mathrm{TD}_{\mathrm{robust}})$ associated with this nodal structure.

B. We know in advance some nodes which for sure will be present in the solution (certainly the nodes where the forces from the given loading scenarios are applied) and it seems to be natural to require rigidity with respect to all properly scaled forces acting at these "active" nodes.

Let us discuss in more detail the latter possibility. Let $F = \{f_1, ..., f_k\}$ be the given set of loading scenarios. We say that a node $v \in \mathcal{V}$ is *active* with respect to F if the projection of certain load f_j on the space R_v of virtual displacements of the node is nonzero. Let $\mathcal{V}^*$ be the set of all active nodes. Our goal is to embed F into a "reasonably chosen" ellipsoid M in the space $R^q = \prod_{v \in \mathcal{V}^*} R_v$ (which for sure will be the part of the displacement space in the final construction). According to our motivation, M should contain

(i) the set F of given loads;

(ii) the ball $B = \{f \in R^q | \, f^T f \leq r^2\}$ of all "occasional" loads of prescribed magnitude r.

The setup $M = F \cup B$ most adequate to our motivation is inappropriate; as it was explained, we need M to be an ellipsoid in order to get a computationally tractable problem, so that we should look for "the smallest possible" ellipsoid M containing $F \cup B$. The simplest interpretation of "the smallest possible" here is in terms of q-dimensional volume. Thus, it is natural to choose as M the ellipsoid in R^q centered at the origin and containing $F \cup B$ of the minimum q-dimensional volume. To form the indicated *ellipsoidal envelope* M of F and B is a convex problem; since normally q is not large, there is no difficulty to solve the problem numerically. Note, however, that there exists an "easy case" where M can be pointed out explicitly. Namely, let $L(F) \subset R^k$ be the linear span of F. Assume that

1. the loads $f_1, ..., f_k$ are linearly independent;
2. the convex hull $\hat{F}$ of the set $F \cup (-F)$ contains the k-dimensional ball $B' = B \cap L(F)$.

Note that in actual design both these assumptions normally are satisfied.

Lemma 2.1. *Under the indicated assumptions the ellipsoidal envelope of F and B is*

$$M = QW_q, \quad Q = [f_1; ...; f_k; re_1; ...; re_{q-k}], \tag{2.4}$$

where $e_1, ..., e_{q-k}$ is an orthonormal basis in the orthogonal complement to $L(F)$ in R^q.

Proof. We can choose an orthonormal basis in R^q in such a way that the first k vectors of the basis span $L(F)$ and the rest $q-k$ vectors span the orthogonal complement $L^{\perp}(F)$ to $L(F)$ in R^q. Let $x=(u,v)$ be the coordinates of a vector in this basis (u are the first k and v are the rest $q-k$ coordinates). A centered at the origin ellipsoid E in R^q can be parameterized by a positive definite symmetric $q\times q$ matrix A:

$$E=\{x|\, x^T A x\le 1\};$$

the squared volume of E is inversely proportional to $\det A$. The matrix A_* corresponding to the minimum volume centered at the origin ellipsoid containing F and B is therefore an optimal solution to the following convex program:

$$\ln\det A\to\max\,|\, A=A^T>0,\; x^T A x\le 1\ \forall x\in B\cup\hat{F}. \tag{2.5}$$

The problem clearly is solvable, and since its objective is strictly concave on the cone of positive definite symmetric $q\times q$ matrices, the solution is unique. On the other hand, let J be the matrix of the mapping $(u,v)\mapsto(u,-v)$; then the mapping $A\mapsto J^T A J$ clearly is a symmetry of (2.5). This mapping preserves feasibility and does not vary the value of the objective. We conclude that the optimal solution is invariant with respect to the indicated mapping: $A_*=JA_*J$, whence A_* is block diagonal with $k\times k$ diagonal block U_* and $(q-k)\times(q-k)$ diagonal block V_*. Since the ellipsoid $\{x|\, x^T A_* x\le 1\}$ contains $B\cup\hat{F}$, the k-dimensional ellipsoid $M'=\{u|\, u^T U_* u\le 1\}$ in $L(F)$ contains $\hat{F}$, while the $(q-k)$-dimensional ellipsoid $M''=\{v|\, v^T V_* v\le 1\}$ in $L^{\perp}(F)$ contains the ball B'' centered at the origin of the radius r in $L^{\perp}(F)$.

Now let $U=U^T>0$ and $V=V^T>0$ be $k\times k$ and $(q-k)\times(q-k)$ matrices such that the ellipsoids $E'=\{u|\, u^T U u\le 1\}$ and $E''=\{v|\, v^T V v\le 1\}$ contain $\hat{F}$ and B'', respectively. We claim that then the ellipsoid $\{x|\, x^T A x\le 1\}$, $A=\mathrm{Diag}(U,V)$, contains $B\cup\hat{F}$. Indeed, the ellipsoid clearly contains $\hat{F}$, and all we need is to verify that if $x=(u,v)\in B$, i.e., $u^T u+v^T v\le r^2$, then $u^T U u+v^T V v\le 1$. This is immediate: since $E'\supset\hat{F}\supset B'$, we have $u^T U u\le 1$ whenever $u^T u\le r^2$, or, which is the same, $u^T U u\le r^{-2}u^T u$ for all u. Similarly, $E''\supset B''$ implies that $v^T V v\le r^{-2}v^T v$, so that $u^T u+v^T v\le r^2$ indeed implies $u^T U u+v^T V v\le 1$.

The above observations combined with the identity $\ln\det A=\ln\det U+\ln\det V$ for positive definite symmetric $A=\mathrm{Diag}(U,V)$ demonstrate that the block U_* of the optimal solution to (2.5) corresponds to the minimum volume ellipsoid in $L(F)$ containing $\hat{F}$, and similarly for V_*, $L^{\perp}(F)$ and B''. In other words, M is the "ellipsoidal product" of the ellipsoid M' of the minimum volume in $L(F)$ containing $F\cup(-F)$ and the ball B'' in $L^{\perp}(F)$. If $M'=Q'W_k$, then

$$M=[Q';re_1;...;re_{q-k}]\,W_q.$$

To conclude the proof, it suffices to verify that one can choose, as Q', the matrix $[f_1;...;f_k]$, which is immediate. Indeed, let $s_1,...,s_k$ be an orthonormal basis in $L(F)$, and let D be the linear transformation of $L(F)$ which maps s_i onto f_i, $i=1,...,k$. Since the ratio of k-dimensional volumes of solids in $L(F)$ remains invariant under the transformation D, $M'=DN'$, where N' is the minimum volume ellipsoid centered at the origin in $L(F)$ containing $s_1,...,s_k$. The latter ellipsoid is clearly $[s_1;...;s_k]\,W_k$, whence

$$M'=DN'=\left\{D\left(\sum_{i=1}^k\lambda_i s_i\right)|\,\lambda\in W_k\right\}=\left\{\sum_{i=1}^k\lambda_i f_i|\,\lambda\in W_k\right\}=[f_1;...;f_k]\,W_k. \qquad \Box$$

Remark 2.1. Evident modification of the proof of Lemma 2.1 demonstrates that the minimum volume ellipsoid in R^q centered at the origin and containing $F \cup B$ always is the "ellipsoidal product" of the minimum volume ellipsoid M' in $L(F)$ containing $F \cup (-F) \cup B'$ and the ball B'' in $L^{\perp}(F)$. If $M' = Q'W_{\widehat{k}}$, $\widehat{k} = \dim L(F)$, then $M = [Q'; re_1, ..., re_{q-\widehat{k}}]W_q$, $e_1, ..., e_{q-\widehat{k}}$ being an orthonormal basis in $L^{\perp}(F)$. Thus, to find M is, basically, the same as to find M', and this latter convex problem is normally of quite a small dimension, since $\widehat{k} \le k$ and typically $k \le 5$.

The outlined way of modeling the robustness constraint is, perhaps, more reasonable than the usual multiload setting of the TTD problem. Indeed, the new model enforces certain level of rigidity of the resulting construction with respect not only to the primary loads, but also to loads associated with "active" nodes. At the same time, it turns out, as we are about to demonstrate, that the resulting problem ($\mathrm{TD_{robust}}$) is basically not more computationally demanding than the usual multiload TTD problem of the same size (i.e., with the same ground structure and the number of scenario loads equal to the dimension of the loading ellipsoid used in ($\mathrm{TD_{robust}}$)).

2.4. Semidefinite reformulation of ($\mathbf{TD_{robust}}$) . Our goal now is to rewrite ($\mathrm{TD_{robust}}$) equivalently as a so-called *semidefinite program.* To this end we start with the following simple result.

LEMMA 2.2. *Let A be a positive semidefinite $n \times n$ matrix, and let*

$$c = \max_{x \in R^n; e \in R^q : e^T e \le 1} \left[2(Qe)^T x - x^T A x \right] . \tag{2.6}$$

Then the inequality $c \le \tau$ is equivalent to positive semidefiniteness of the matrix

$$\mathcal{A} = \begin{pmatrix} \tau I_q & Q^T \\ Q & A \end{pmatrix},$$

I_q being the unit $q \times q$ matrix.

Proof. We have

$$\begin{aligned}
& c \le \tau \Leftrightarrow \forall (x \in R^n, e \in R^q, e^T e \le 1) : \quad \tau - 2(Qe)^T x + x^T A x \ge 0 \Leftrightarrow \\
& \qquad \text{[by homogeneity reasons]} \\
& \forall (\lambda > 0, x \in R^n, e \in R^q, e^T e \le 1) : \quad \tau \lambda^2 - 2(Q\lambda e)^T (\lambda x) + (\lambda x)^T A (\lambda x) \ge 0 \Leftrightarrow \\
& \qquad \text{[set } \lambda e = f, \lambda x = y] \\
& \forall (\lambda > 0, y \in R^n, f \in R^q, f^T f \le \lambda^2) : \quad \tau \lambda^2 - 2(Qf)^T y + y^T A y \ge 0 \Rightarrow \\
& \forall \left(\begin{pmatrix} f \\ y \end{pmatrix} \in R^{q+n} \right) : \quad \begin{pmatrix} f \\ y \end{pmatrix}^T \begin{pmatrix} \tau I_q & Q^T \\ Q & A \end{pmatrix} \begin{pmatrix} f \\ y \end{pmatrix} \equiv \tau f^T f - 2(Qf)^T y + y^T A y \ge 0.
\end{aligned}$$

Thus, $\tau \ge c \Rightarrow \mathcal{A} \ge 0$. Vice versa, if $\mathcal{A} \ge 0$, then clearly $\tau \ge 0$, and, therefore, the implication $\Rightarrow$ in the above chain can be inverted. □

Remark 2.2. It is well known that a symmetric matrix $\left(\begin{smallmatrix} U & Q^T \\ Q & A \end{smallmatrix} \right)$ with positive definite U is positive semidefinite if and only if $A \ge QU^{-1}Q^T$. Applying this observation to the case of $U = \tau I_q$, we can reformulate the result of Lemma 2.2 as follows:

The compliance c of a truss t, with respect to the ellipsoid of loads $M = QW_q$ is $\le \tau$ if and only if $A(t) \ge \tau^{-1} QQ^T$.

In the particular case when QQ^T is the orthoprojector P onto the linear span L of the columns of Q, the above observation can be reformulated as follows:

$c \le \tau$ if and only if the minimum eigenvalue of the restriction of $A(t)$ onto L is $\ge \tau^{-1}$.

(In the general case, the interpretation is similar, but instead of the usual minimum eigenvalue of the restriction we should speak about minimum eigenvalue of the matrix pencil $(A|_L, QQ^T|_L)$ on L.)

In view of Lemma 2.2, problem $(\mathrm{TD}_{\mathrm{robust}})$ can be rewritten equivalently as the following *semidefinite program*:

$$(\mathrm{TD}_{\mathrm{sd}}) \qquad \min_{t\in R^m, \tau\in R} \tau$$

$$\text{subject to} \qquad \begin{array}{rcl} \begin{pmatrix} \tau I_q & Q^T \\ Q & A(t) \end{pmatrix} & \geq & 0, \\ t & \geq & 0, \\ \sum_{i=1}^{m} t_i & = & V. \end{array}$$

(Here and in what follows the inequality $A \geq B$ between symmetric matrices means that the matrix $A - B$ is positive semidefinite.)

3. Deriving a dual problem to $(\mathbf{TD}_{\mathbf{sd}})$.

Here we derive the Fenchel–Rockafellar [11] dual to the problem $(\mathrm{TD}_{\mathrm{sd}})$. The latter problem is of the form

$$\min\{\tau : \; \mathcal{A}(\tau, t) + B \in \mathbf{S}_+, \, t \in T\},$$

where

$$\mathcal{A}(\tau, t) = \begin{pmatrix} \tau I_q & 0 \\ 0 & A(t) \end{pmatrix}$$

is a linear mapping from $R \times R^n$ to the space $\mathbf{S}$ of symmetric $(n+q)\times(n+q)$ matrices equipped with the standard Frobenius Euclidean structure $\langle X, Y\rangle = \mathrm{Tr}(XY)$, $\mathbf{S}_+$ is the cone of positive semidefinite matrices from $\mathbf{S}$ and

$$B = \begin{pmatrix} 0 & Q^T \\ Q & 0 \end{pmatrix} \in \mathbf{S}.$$

We write the problem in the Fenchel–Rockafellar primal scheme:
(P) $\quad \min\left\{f(\tau, t) - g(\mathcal{A}(\tau, t))\right\}$,
where

$$f(\tau, t) = \tau + \delta(t|T), \quad g(X) = -\delta(X + B|\mathbf{S}_+)$$

and $\delta(x|W)$ is the indicator function of a set W. To derive the dual to (P), we need to compute the conjugates f^* and g_* of the convex function f and the concave function g, which is quite straightforward:

$$\begin{array}{rcl} f^*(\sigma, s) & = & \sup_{\tau,t}\{\sigma\tau + s^T t - \tau | \, t \in T\} = \begin{cases} V \max_{1\leq i\leq n} s_i, & \sigma = 1, \\ +\infty & \text{otherwise}, \end{cases} \\ g_*(R) & = & \inf_S\{\mathrm{Tr}(SR) | S + B \in \mathbf{S}_+\} = \inf\{\mathrm{Tr}((Z - B)R) | Z \in \mathbf{S}_+\} \\ & = & \begin{cases} -\,\mathrm{Tr}(BR), & R \in \mathbf{S}_+, \\ -\infty & \text{otherwise}. \end{cases} \end{array}$$

(We have used the well-known fact that the cone of positive semidefinite matrices is self-conjugate with respect to the Frobenius Euclidean structure.)

The Fenchel–Rockafellar dual to (P) is

$$\text{(D)} \qquad \sup_{R\in\mathbf{S}} \{g_*(R) - f^*(\mathcal{A}^*R)\},$$

where $\mathcal{A}^* : \mathbf{S} \to R \times R^n$ is the adjoint to $\mathcal{A}$.

Representing $R \in \mathbf{S}$ in the block form

$$R = \begin{pmatrix} \Lambda & X^T \\ X & Y \end{pmatrix}$$

(Λ is $q \times q$, Y is $n \times n$), we get

$$\mathcal{A}^*R = \begin{pmatrix} \tau = \operatorname{Tr}\Lambda \\ t_1 = \operatorname{Tr}(A_1 Y) \\ \dots \\ t_n = \operatorname{Tr}(A_n Y) \end{pmatrix}.$$

Substituting the resulting expressions for f^*, g_*, and A^*, we come to the following explicit formulation of the dual problem (D):

$$\text{(D)} \qquad \max\left[-2\operatorname{Tr}(QX^T) - V\max_{i=1,\dots,m}[\operatorname{Tr}(A_i Y)]\right],$$

s.t.

$$\begin{pmatrix} \Lambda & X^T \\ X & Y \end{pmatrix} \geq 0, \ \operatorname{Tr}\Lambda = 1,$$

the design variables being symmetric $q \times q$ and $n \times n$ matrices Λ, Y, respectively, and $n \times q$ matrix X.

Note that the functions f and g in (P) are clearly closed convex and concave, respectively. Moreover, from the well-posedness assumption **A**, it immediately follows that (P) is strictly feasible (i.e., the relative interiors of the domains of $f(\tau, t)$ and $\phi(\tau, t) = g(\mathcal{A}(\tau, t))$ have nonempty intersection, and the image of the mapping $\mathcal{A}$ intersects the interior of the domain of g); to see this, choose arbitrary positive $t \in T$ and enforce τ to be large enough. Of course (P) is bounded below (the compliance always is nonnegative); thus, all requirements of the Fenchel–Rockafellar duality theorem are satisfied, and we come to the following.

PROPOSITION 3.1. (D) *is solvable, and the optimal values in* (P) *and* (D) *are equal to each other.*

Remark 3.1. Until now, we dealt with the TTD problem with *simple constraints* on the bar volumes:

$$t \in T = \left\{ t \in R^n \,|\, t \geq 0, \sum_{i=1}^n t_i = V \right\}.$$

In the case when there are also lower and upper bounds on the bar volumes so that the constraints on t are

$$t \in T^+ = \{t \in T \,|\, L \leq t \leq U\}$$

($U > L \geq 0$ are given n-dimensional vectors), the above derivation results in a dual problem as follows:

$$(\mathrm{D_b}) \quad \max\left[-2\operatorname{Tr}(QX^T) - \lambda V - \textstyle\sum_{i=1}^n \max\left[(\operatorname{Tr}(YA_i) - \lambda)L_i; (\operatorname{Tr}(YA_i) - \lambda)U_i\right]\right]$$

s.t.

$$\begin{pmatrix} \Lambda & X^T \\ X & Y \end{pmatrix} \geq 0, \ \operatorname{Tr}\Lambda = 1,$$

the design variables being real λ, symmetric $q \times q$ matrix Λ, symmetric $n \times n$ matrix Y, and $n \times q$ matrix X.

4. A simplification of the dual problem (D). Our next goal is to simplify problem (D), derived in the previous section, by eliminating the matrix variable Y. To this end it suffices to note that (D) can be rewritten as

$$
\begin{array}{llcl}
(\mathrm{TD}_{\mathrm{dl}}) & & & \\
& \multicolumn{3}{c}{\min\limits_{X\in R^{n\times q},\Lambda=\Lambda^T\in R^{q\times q},Y=Y^T\in R^{n\times n},\rho\in R} [2\,\mathrm{Tr}(QX^T)+V\rho]} \\
\text{s.t.} & & & \\
(\alpha) & \mathrm{Tr}(YA_i) & \leq & \rho,\ i=1,\dots,m, \\
(\beta) & \begin{pmatrix}\Lambda & X^T\\ X & Y\end{pmatrix} & \geq & 0, \\
(\gamma) & \mathrm{Tr}(\Lambda) & = & 1.
\end{array}
$$

(We have replaced the maximization problem (D) by an equivalent minimization one.) Note that $(\mathrm{TD}_{\mathrm{dl}})$ is strictly feasible—there exists a feasible solution where all scalar inequality constraints and the matrix inequality one are strict (take $\Lambda = q^{-1}I_q$, $Y = I_n$, and enforce ρ to be large enough).

The matrix inequality (β) clearly implies that Λ is positive semidefinite. Thus, we do not vary $(\mathrm{TD}_{\mathrm{dl}})$ when adding (in fact, redundant) inequality $\Lambda \geq 0$. Now let us strengthen, for a moment, the latter inequality to

$$\Lambda > 0, \tag{4.1}$$

i.e., to positive definiteness of Λ; it is immediately seen from strict feasibility of $(\mathrm{TD}_{\mathrm{dl}})$ that the transformation does not violate the optimal value of the problem, although it may cut off the optimal solution (anyhow, from the computational viewpoint the exact solution is nothing but a fiction). Thus, we may focus on the problem $(\mathrm{TD}'_{\mathrm{dl}})$ obtained from $(\mathrm{TD}_{\mathrm{dl}})$ by adding to the list of constraints inequality (4.1).

The pair of matrix inequalities (β), (4.1), which are present among the constraints of $(\mathrm{TD}'_{\mathrm{dl}})$, is equivalent to the pair of matrix inequalities

$$\Lambda > 0; \quad Y \geq Y^*(\Lambda, X) = X\Lambda^{-1}X^T.$$

Now let (Λ, X, Y, ρ) be a feasible solution to $(\mathrm{TD}'_{\mathrm{dl}})$; then, as we have just mentioned, $Y \geq Y^*(\Lambda, X)$ and the collection $(\Lambda, X, Y^* = Y^*(\Lambda, X), \rho)$ satisfies (β), (γ) and (4.1). Moreover, since A_i are symmetric positive semidefinite and $Y \geq Y^*$, we have $\mathrm{Tr}(YA_i) \geq \mathrm{Tr}(Y^*A_i)$ so that the updated collection satisfies (α) as well, and (Λ, X, Y^*, ρ) is feasible for $(\mathrm{TD}'_{\mathrm{dl}})$. Note that the transformation $(\Lambda, X, Y, \rho) \mapsto (\Lambda, X, Y^*(\Lambda, X), \rho)$ does not affect the objective function of the problem. We conclude that $(\mathrm{TD}'_{\mathrm{dl}})$ can be equivalently rewritten as

$$
\begin{array}{ll}
& \min_{X\in R^{n\times q},\Lambda=\Lambda^T\in R^{q\times q},\rho\in R}\ 2\,\mathrm{Tr}(QX^T)+V\rho \\
\text{s.t.} & \\
& \Lambda > 0,\ \mathrm{Tr}(\Lambda)=1,\ \rho \geq \mathrm{Tr}(X\Lambda^{-1}X^TA_i),\ i=1,...,m.
\end{array}
$$

Substituting $A_i = b_ib_i^T$ (see (2.1)), we can rewrite the constraints

$$\rho \geq \mathrm{Tr}(X\Lambda^{-1}X^TA_i)$$

as

$$\rho \geq (X^Tb_i)^T\Lambda^{-1}(X^Tb_i),$$

which is the same (since $\Lambda = \Lambda^T > 0$) as

$$\begin{pmatrix} \Lambda & X^T b_i \\ b_i^T X & \rho \end{pmatrix} \geq 0.$$

With this substitution, the problem $(\mathrm{TD}'_{\mathrm{dl}})$ becomes

$$\begin{array}{ll} & \min_{X \in R^{n \times q}, \Lambda = \Lambda^T \in R^{q \times q}, \rho \in R} \; 2\,\mathrm{Tr}(QX^T) + V\rho \\ \text{s.t.} & \\ & \Lambda > 0,\ \mathrm{Tr}(\Lambda) = 1,\ \begin{pmatrix} \Lambda & X^T b_i \\ b_i^T X & \rho \end{pmatrix} \geq 0,\ i = 1, ..., m. \end{array}$$

When replacing the strict inequality $\Lambda > 0$ in the latter problem with the nonstrict one $\Lambda \geq 0$, we clearly do not vary the optimal value of the problem; in the modified problem, the inequality $\Lambda \geq 0$ is in fact redundant (it follows from positive semidefiniteness of any of the matrices $\begin{pmatrix} \Lambda & X^T b_i \\ b_i^T X & \rho \end{pmatrix}$). With these modifications, we come to the final formulation of the problem dual to $(\mathrm{TD}_{\mathrm{robust}})$:

$$\begin{array}{llcl} (\mathrm{TD}_{\mathrm{fn}}) & & & \\ & \min_{\Lambda = \Lambda^T \in R^{q \times q}, X \in R^{n \times q}, \rho \in R} & & 2\,\mathrm{Tr}(QX^T) + V\rho \\ \text{s.t.} & & & \\ & \begin{pmatrix} \Lambda & X^T b_i \\ b_i^T X & \rho \end{pmatrix} & \geq & 0,\quad i = 1, ..., m, \\ & \mathrm{Tr}(\Lambda) & = & 1. \end{array}$$

Note that $(\mathrm{TD}_{\mathrm{fn}})$ is very similar to the standard multiload TTD problem in dual setting [5]; the only difference is that in the latter problem Λ is further restricted to be diagonal.

5. Recovering the bar volumes. Until now, the only relation between the initial primal problem $(\mathrm{TD}_{\mathrm{robust}})$ and the dual one $(\mathrm{TD}_{\mathrm{fn}})$ is that their optimal values are negations of each other (note that when coming to $(\mathrm{TD}_{\mathrm{fn}})$ from the maximization problem $(\mathrm{TD}_{\mathrm{dl}})$, which has the same optimal value as $(\mathrm{TD}_{\mathrm{sd}})$, we have changed the sign of the objective and have replaced maximization with minimization). Thus, the problem arises: how to restore good approximate solutions to $(\mathrm{TD}_{\mathrm{robust}})$ via good approximate solutions to $(\mathrm{TD}_{\mathrm{fn}})$. To resolve this problem, we first derive the Fenchel–Rockafellar dual $(\mathrm{TD}^*_{\mathrm{fn}})$ to $(\mathrm{TD}_{\mathrm{fn}})$ and recognize in it the initial problem $(\mathrm{TD}_{\mathrm{robust}})$, and then use the well-known relation in interior point theory between "central path" approximate solutions to $(\mathrm{TD}_{\mathrm{fn}})$ and approximate solutions to $(\mathrm{TD}^*_{\mathrm{fn}})$.

5.1. A dual problem to $(\mathbf{TD}_{\mathbf{fn}})$. Similar to the above, we represent problem $(\mathrm{TD}_{\mathrm{fn}})$ in the Fenchel–Rockafellar scheme:
(PI) $\qquad \min \left\{ f(\Lambda, X, \rho) - g(\mathcal{A}(\Lambda, X, \rho)) \right\},$
where

$$f(\Lambda, X, \rho) = 2\,\mathrm{Tr}(QX^T) + V\rho + \delta(\mathrm{Tr}(\Lambda)|\{1\}),$$

$$\mathcal{A}(\Lambda, X, \rho) = \mathrm{Diag}\left\{ \begin{pmatrix} \Lambda & X^T b_i \\ b_i^T X & \rho \end{pmatrix}, i = 1, ..., m \right\}$$

is the linear mapping from the space of design variables of $(\mathrm{TD}_{\mathrm{fn}})$ to the space $\mathbf{S}$ of block-diagonal symmetric matrices with m diagonal blocks of the sizes $(q+1)\times(q+1)$ each, and

$$g(W) = -\delta(W|\mathbf{S}_+),$$

$\mathbf{S}_+$ being the cone of positive semidefinite matrices from $\mathbf{S}$.

The dual to (P) is

$$\text{(DI)} \qquad \max_{R\in\mathbf{S}} \left\{ g_*(R) - f^*(\mathcal{A}^*R) \right\},$$

where $\mathcal{A}^*$ is the operator adjoint to $\mathcal{A}$. Here

$$\begin{array}{rcl} f^*(L,\Xi,r) & = & \sup_{\Lambda,X,\rho}\left[\mathrm{Tr}(\Lambda L) + \mathrm{Tr}(\Xi X^T) + r\rho - f(\Lambda,X,\rho)\right] \\ & = & \sup_{\Lambda}\left[\mathrm{Tr}(\Lambda L) - \delta(\mathrm{Tr}(\Lambda)|\{1\})\right] + \sup_X\left[\mathrm{Tr}(\Xi X^T) - 2\,\mathrm{Tr}(QX^T)\right] \\ & & + \sup_\rho\left[r\rho - V\rho\right] \\ & = & \dfrac{1}{q}\,\mathrm{Tr}(L) + \delta((L,\Xi,r)|\{(L=\lambda I_q, 2Q, V)|\,\lambda\in R\}) \\ & = & \begin{cases} \lambda & \text{if } L=\lambda I_q \text{ for some } \lambda\in R \text{ and } \Xi = 2Q,\ r=V, \\ \infty & \text{otherwise} \end{cases} \end{array}$$

and

$$g_*(R) = \inf_S\left[\mathrm{Tr}(SR) + \delta(S|\mathbf{S}_+)\right] = -\delta(R|\mathbf{S}_+).$$

(Here we again used the fact that the cone $\mathbf{S}_+$ is self-dual with respect to the Frobenius Euclidean structure of $\mathbf{S}$.)

Denoting a generic element of $\mathbf{S}$ as

$$R = \mathrm{Diag}\left\{ \begin{pmatrix} L_i & d_i \\ d_i^T & t_i \end{pmatrix},\ i=1,\dots,m \right\}$$

(L_i are symmetric $q\times q$ matrices, d_i are q-dimensional vectors, t_i are reals) it can be seen that

$$\mathcal{A}^*R = \left(L = \sum_{i=1}^m L_i, \Xi = 2\sum_{i=1}^m b_i d_i^T, r = \sum_{i=1}^m t_i \right).$$

With these relations, the dual (DI) to (PI) becomes

$$\begin{array}{llcl} (\mathrm{TD}^*_{\mathrm{fn}}) & & & \\ & \min\limits_{\lambda\in R, L_i = L_i^T\in R^{q\times q}, d_i\in R^q, t_i\in R} & & \lambda \\ \text{s.t.} & & & \\ (\alpha) & \sum\limits_{i=1}^m L_i & = & \lambda I_q, \\ (\beta) & \sum\limits_{i=1}^m b_i d_i^T & = & Q, \\ (\gamma) & \sum\limits_{i=1}^m t_i & = & V, \\ (\delta) & \begin{pmatrix} L_i & d_i \\ d_i^T & t_i \end{pmatrix} & \geq & 0,\ i=1,\dots,m. \end{array}$$

(We again have replaced a maximization problem with the equivalent minimization one.)

Problem $(\mathrm{TD_{fn}})$ clearly satisfies the assumption of the Fenchel–Rockafellar duality theorem, and this together with Proposition 3.1 proves the following.

PROPOSITION 5.1. *Problem* $(\mathrm{TD}^*_{\mathrm{fn}})$ *is solvable, and its optimal value* λ^* *is equal to the optimal value* c^* *of the initial problem* $(\mathrm{TD_{robust}})$.

It is not difficult to guess that the variables t_i involved into $(\mathrm{TD}^*_{\mathrm{fn}})$ can be interpreted as our initial bar volumes t_i. The exact statement is given by the following theorem.

THEOREM 5.2. *Let* $R = \{\lambda; L_i, d_i, t_i,\ i = 1, ..., m\}$ *be a feasible solution to* $(\mathrm{TD}^*_{\mathrm{fn}})$ *. Then the vector* $t = (t_1, ..., t_m)$ *is a feasible solution to* $(\mathrm{TD_{robust}})$ *, and the value of the objective of the latter problem at* t *is less than or equal to* λ*. In particular, if* R *is an* ϵ*-solution to* $(\mathrm{TD}^*_{\mathrm{fn}})$ *(i.e.,* $\lambda - \lambda^* \leq \epsilon$*), then* t *is an* ϵ*-solution to* $(\mathrm{TD_{robust}})$ *(i.e.,* $c^M(t) - c^* \leq \epsilon$*).*

Proof. The "in particular" part of the statement follows from its first part due to Proposition 5.1, and all we need is to prove the first part. From the positive semidefiniteness constraints (δ) in $(\mathrm{TD}^*_{\mathrm{fn}})$ it follows that $t \geq 0$, which combined with (γ) implies the inclusion $t \in T$. To complete the proof, we should verify that $c^M(t) \leq \lambda$.

Let $e \in R^q$, $e^T e \leq 1$. From (β) we have

$$Qe = \sum_{i=1}^{m} (d_i^T e) b_i.$$

Let $x \in R^n$. Due to $A_i = b_i b_i^T$, we have

$$\begin{array}{rcl}
\phi_e(x) & \equiv & 2(Qe)^T x - x^T A(t) x \\
& = & \sum\limits_{i=1}^{m} 2(d_i^T e)(b_i^T x) - t_i \sum\limits_{i=1}^{m} (b_i^T x)^2 \\
& = & \sum\limits_{i=1}^{m} \left[2(d_i^T e)(b_i^T x) - t_i (b_i^T x)^2 \right] \\
& & \quad [\text{denoting } s_i = -b_i^T x] \\
& = & -\sum\limits_{i=1}^{m} \left[e^T L_i e + 2(d_i^T e) s_i + t_i s_i^2 \right] + \sum\limits_{i=1}^{m} e^T L_i e \\
& = & -\sum\limits_{i=1}^{m} \begin{pmatrix} e \\ s_i \end{pmatrix}^T \begin{pmatrix} L_i & d_i \\ d_i^T & t_i \end{pmatrix} \begin{pmatrix} e \\ s_i \end{pmatrix} + \sum\limits_{i=1}^{m} e^T L_i e \\
& & \quad [\text{by } (\delta)] \\
& \leq & \sum\limits_{i=1}^{m} e^T L_i e \\
& & \quad [\text{by } (\alpha)] \\
& = & \lambda.
\end{array}$$

Thus, $\phi_e(x) \leq \lambda$ for all x. By definition, $c^M(t)$ is the upper bound of $\phi_e(x)$ over x, and the inequality $c^M(t) \leq \lambda$ then follows. □

Remark 5.1. Note that $(\mathrm{TD}^*_{\mathrm{fn}})$ is a natural modification of the "bar-forces" formulation of the usual multiload TTD problem; see [5].

6. Solving ($\mathrm{TD_{fn}}$) and ($\mathrm{TD^*_{fn}}$) via interior point methods. Among numerical methods available for solving semidefinite programs like ($\mathrm{TD_{fn}}$) and ($\mathrm{TD^*_{fn}}$) , the most attractive (and, in fact, the only meaningful in the large scale case) are the recent interior point algorithms (for relevant general theory, see [10]). Here we discuss the corresponding possibilities. In what follows we restrict ourselves to outlining the main elements of the construction, since our goal now is not to present detailed description of the algorithms, but to demonstrate the following.

(i) From the above semidefinite programs related to truss topology design with robustness constraints, the most convenient for numerical processing by interior point methods is the problem ($\mathrm{TD_{fn}}$) .

(ii) Solving ($\mathrm{TD_{fn}}$) by *interior point path-following methods*, one has the possibility of generating, as a byproduct, good approximate solutions to the problem of interest ($\mathrm{TD^*_{fn}}$), i.e., of recovering the primal design variables (bar volumes).

When solving a generic semidefinite program

(SP) $\quad \sigma^T\xi \to \min \,|\, \mathcal{A}(\xi) \in \mathbf{S}_+,$

$\xi \in R^N$ being the design vector, $\mathcal{A}(\xi)$ being an affine mapping from R^N to the space $\mathbf{S}$ of symmetric matrices of certain fixed block-diagonal structure, and $\mathbf{S}_+$ being the cone of positive semidefinite matrices from $\mathbf{S}$, by a path-following interior point method, one defines the family of barrier-type functions

$$F_s(\xi) = s\sigma^T\xi + \Phi(\mathcal{A}(\xi)), \ \ \Phi(\Xi) = -\ln \mathrm{Det}\, \Xi,$$

and traces the *central path*—the path of minimizers

$$\xi^*(s) = \underset{\xi \in \mathrm{Dom}\, F_s}{\mathrm{argmin}}\ F_s(\xi).$$

If (SP) is strictly feasible (i.e., $\mathcal{A}(\xi)$ is positive definite for certain ξ) and the level sets

$$\{\xi \in R^N \,|\, \mathcal{A}(\xi) \in \mathbf{S}_+, \sigma^T\xi \le a\},$$

$a \in R$, are bounded, then the path ξ^* is well defined and converges, as $s \to \infty$, to the optimal set of the problem. In the path-following scheme, one generates close (in certain exact sense) approximations ξ_i to the points $\xi^*(s_i)$ along certain sequence $\{s_i\}$ of penalty parameters "diverging to ∞ fast enough," thus generating a sequence of strictly feasible approximate solutions converging to the optimal set. Updating $(s_i, \xi_i) \mapsto (s_{i+1}, \xi_{i+1})$ is as follows: first, we increase, according to certain rule, the current value s_i to a larger value s_{i+1}. Second, we restore closeness to the path of the new point $\xi^*(s_{i+1})$ by running the *damped Newton method*—the recurrence

$$\begin{array}{rcl} y & \mapsto & y^+ = y - (1+\lambda(F_s, y))^{-1}[\nabla^2_y F_s(y)]^{-1}\nabla_y F_s(y), \\ \lambda(F_s, y) & = & \sqrt{\nabla^T_y F_s(y)[\nabla^2_y F_s(y)]^{-1}\nabla_y F_s(y)}, \end{array} \tag{6.1}$$

with s set to s_{i+1}. The recurrence is started at $y = \xi_i$ and is terminated when, for the first time, it turns out that $\lambda(F_{s_{i+1}}, y) \le \kappa$, $\kappa \in (0, 1)$ being a once forever fixed threshold. (Thus, the exact meaning of "closeness of a point ξ to the point $\xi^*(s)$" is given by the inequality $\lambda(F_s, \xi) \le \kappa$. In what follows, for the sake of definiteness, it is assumed that $\kappa = 0.1$.) The resulting y is chosen as ξ_{i+1}, and the process is iterated.

The following is known:

(i) it is possible to trace the path "quickly": with reasonable policy of updating the values of the penalty parameter, it takes, for any $T > 2$, no more than

$$M = M(T) = O(1)\sqrt{\mu}\ln T$$

Newton steps (6.1) to come from a point ξ_0 close to $\xi^*(s_0)$ to a point ξ_M close to $\xi^*(s_M)$ with $s_M \geq Ts_0$; here μ is the total row size of the matrices from $\mathbf{S}$ and $O(1)$ is an absolute constant;

(ii) if ξ is close to $\xi(s)$, then the quality of ξ as an approximate solution to (SP) can be expressed via the value of s alone:

$$\sigma^T\xi - \sigma^* \leq \frac{2\mu}{s}, \tag{6.2}$$

σ^* being the optimal value in (SP);

(iii) being close to the path, it is easy to come "very close" to it; if $\lambda \equiv \lambda(F_s, y) \leq 0.1$, then (6.1) results in

$$\lambda^+ \equiv \lambda(F_s, y^+) \leq 2.5\lambda^2. \tag{6.3}$$

Although the indicated remarks deal with the path-following scheme only, the conclusions related to the number of "elementary steps" required to solve a semidefinite program to a given accuracy and to the complexity of a step (dominated by the computational cost of the Newton direction; see (6.1)) are valid for other interior point methods for semidefinite programming. The "integrated" complexity characteristic of an interior point method for (SP) is the quantity

$$\mathcal{C} = \sqrt{\mu}\mathcal{C}_{\mathrm{Nwt}},$$

where $\mathcal{C}_{\mathrm{Nwt}}$ is the arithmetic cost of computing the Newton direction. Indeed, according to the above remarks, it takes $O(1)\sqrt{\mu}$ Newton steps to increase the value of the penalty by an absolute constant factor, or, which is the same, to reduce by the same factor the (natural upper bound for) inaccuracy of the current approximate solution.

Now let us look at the complexity characteristic $\mathcal{C}$ for the semidefinite programs related to $(\mathrm{TD}_{\mathrm{robust}})$. In the table below we write down the principal terms of the corresponding quantities (omitting absolute constant factors); it is assumed (as it is normally the case for TTD) that

$$m = O(n^2); \quad q << n.$$

The expression for $\mathcal{C}_{\mathrm{Nwt}}$ corresponds to the "explicit" policy when we first assemble, in the natural manner, the Hessian matrix $\nabla^2_\xi F_s(\cdot)$ and then solve the resulting Newton system by traditional direct linear algebra routines like Choleski decomposition. It turns out that the specific structure of matrix inequalities in our problems[3] allows us to assemble the Hessians at a relatively low cost, so that the cost of a single Newton step is dominated by the complexity of Choleski factorization of the Hessian, i.e., by cube of the design dimension of the corresponding problem. With this remark, we come to the results as follows:

[3] In particular, the fact that in TTD design each of the vectors b_i has $O(1)$ nonzero entries—at most four in the case of 2D and at most six in the case of 3D trusses.

Model	μ	$\mathcal{C}_{\rm Nwt}$	$\mathcal{C}$
$(\mathrm{TD}_{\rm sd})$	m	m^3	$m^{3.5}\approx n^7$
$(\mathrm{TD}_{\rm dl})$	m	m^3	$m^{3.5}\approx n^7$
$(\mathrm{TD}_{\rm fn})$	qm	q^3n^3	$q^{3.5}n^4$
$(\mathrm{TD}^*_{\rm fn})$	qm	q^6m^3	$q^{6.5}m^{3.5}\approx q^{6.5}n^7$

The reader should be aware that there are "implicit" schemes of computing the Newton direction in $(\mathrm{TD}^*_{\rm fn})$ with arithmetic cost $O(q^3n^3)$ (the same as in $(\mathrm{TD}_{\rm fn})$). Thus, in fact, the primal and dual problems in primal-dual pairs $((\mathrm{TD}_{\rm sd})\,,\ (\mathrm{TD}_{\rm dl})\,)$, $((\mathrm{TD}_{\rm fn})\,,\ (\mathrm{TD}^*_{\rm fn})\,)$ are theoretically equivalent in complexity; moreover, there are "symmetric" primal-dual methods which solve simultaneously the primal-dual pair of the problems at the complexity, respectively, $O(n^7)$ and $O(q^{3.5}n^4)$. Nevertheless, we believe that at the moment practical considerations still are in favor of "purely primal" methods as applied to $(\mathrm{TD}_{\rm sd})$ in the first primal-dual pair and to $(\mathrm{TD}_{\rm fn})$ in the second pair. The reason is that the feasible planes $\mathcal{L}$ in the "unfavorable" problems of the above pairs are given by linear equalities, while in the "favorable" components of the pairs they are parameterized (from the very beginning they are represented as images of affine mappings). Now, the theoretically efficient way to compute the Newton direction for an "unfavorable" problem represents the direction as the difference of a certain "exactly known" vector and its projection on the orthogonal complement to $\mathcal{L}$. Such a computation is relatively unstable—rounding errors make the actually computed Newton directions nonparallel to $\mathcal{L}$, and the iterates eventually become far from the feasible plane. In order to overcome this instability, in the existing software for semidefinite problems, "expensive" linear algebra routines, like QR factorization, are used, at least at the final phase of computations. In contrast to this, in the "favorable" problems the Newton direction is computed in the space of parameters identifying a point on the feasible plane, so that there is no danger of being kicked off this plane.

With the above remarks, it is clear that among the semidefinite programs we introduced, the most convenient for numerical processing by interior point methods is $(\mathrm{TD}_{\rm fn})$, as it was claimed in **I**. There is, however, an a priori drawback of this approach; what we need are the bar volumes, and they "are not seen" at all in $(\mathrm{TD}_{\rm fn})$. We are about to demonstrate that in order to overcome this difficulty it suffices to solve $(\mathrm{TD}_{\rm fn})$ not by an arbitrary interior point method, but with a path-following one.

Assume that we are applying a path-following method to $(\mathrm{TD}_{\rm fn})$ and have computed a point $\xi=(\Lambda,X,\rho)$ close (in the aforementioned sense) to the point $\xi^*(s)$. From (6.3) it follows that a small number of steps of the recurrence (6.1) started at ξ allows to come "very close" to $\xi^*(s)$ (six steps of the recurrence restore $\xi^*(s)$ within machine accuracy). We may, therefore, assume for the sake of simplicity that we can "stand at the path," i.e., operate with $\xi^*(s)$ itself rather than with a tight approximation of the point.[4] It turns out that given $\xi^*(s)$, one can explicitly generate a feasible solution to $(\mathrm{TD}^*_{\rm fn})$ of inaccuracy $\leq O(1/s)$. The exact statement is as follows.

PROPOSITION 6.1. *Let $s>0$, and let $\xi^*(s)=(\Lambda_s,X_s,\rho_s)$ be the minimizer of the function*

$$F_s(\Lambda,X,\rho)=s\left[2\,\mathrm{Tr}(QX^T)+V\rho\right]+\Phi(\mathcal{A}(\Lambda,X,\rho)) \tag{6.4}$$

[4] This is an idealization, of course, but it is as well motivated as the standard model of precise real arithmetic. We could replace in the forthcoming considerations $\xi^*(s)$ by its tight approximation, with minor modification of the construction, but we do not think it makes sense.

over the set of strictly feasible solutions to (TD$_{\rm fn}$) *. Here*

$$\Phi(S) = -\ln \mathrm{Det}\, S : \mathrm{int}\ \mathbf{S}_+ \to R. \tag{6.5}$$

$\mathbf{S}$ *is the space of block-diagonal symmetric matrices with* m *diagonal blocks of the size* $(q+1)\times(q+1)$ *each, and*

$$\mathcal{A}(\Lambda, X, \rho) = \mathrm{Diag}\left\{ \begin{pmatrix} \Lambda & X^T b_i \\ b_i^T X & \rho \end{pmatrix},\ i = 1, ..., m \right\}. \tag{6.6}$$

Then the matrix

$$\begin{array}{rcl} R(s) & \equiv & \mathrm{Diag}\left\{ \begin{pmatrix} L_i & d_i \\ d_i^T & t_i \end{pmatrix}\ i = 1, ..., m \right\} \\ & = & s^{-1}\mathcal{A}^{-1}(\Lambda_s, X_s, \rho_s) \quad \left[= -s^{-1}\nabla_S|_{S=\mathcal{A}(\Lambda_s, X_s, \rho_s)}\Phi(S)\right] \end{array} \tag{6.7}$$

is such that $\sum_{i=1}^m L_i = \lambda_s I_q$ *for some real* λ_s*, and*

$(R(s), \lambda_s)$ *is a feasible solution to* (TD$^*_{\rm fn}$) *with the value of the objective*

$$\lambda_s \le c^* + \frac{\mu}{s}, \tag{6.8}$$

where c^* *is the optimal value in* (TD$^*_{\rm fn}$) *and* $\mu = m(q+1)$ *is the total row size of the matrices from* $\mathbf{S}$*.*

The proposition is an immediate consequence of general results of [10]; to make the paper self-contained, below we present a direct proof.

Let us set $Y = \mathcal{A}(\Lambda_s, X_s, \rho_s)$, $Z = Y^{-1}$, so that

$$R(s) = s^{-1}Z; \quad \nabla\Phi(Y) = -Z.$$

The set G of strictly feasible solutions to (TD$_{\rm fn}$) is comprised of all triples $\xi = (\Lambda, X, \rho)$, which correspond to positive definite $\mathcal{A}(\xi)$ and are such that $\mathrm{Tr}\,\Lambda = 1$; this is an open convex subset in the hyperplane given by the equation $\mathrm{Tr}\,\Lambda = 1$. Since $\xi^*(s) = (\Lambda_s, X_s, \rho_s)$ is the minimizer of F_s over G, we have, for certain real p,

$$\nabla_\Lambda F_s(\xi^*(s)) = pI_q; \quad \nabla_X F_s(\xi^*(s)) = 0; \quad \nabla_\rho F_s(\xi^*(s)) = 0.$$

Substituting the expression for F_s and $\mathcal{A}$, we obtain

$$\begin{array}{rcccccl} \sum_{i=1}^m L_i & \equiv & [\mathcal{A}^* R(s)]_\Lambda & \equiv & -s^{-1}[\mathcal{A}^*\nabla\Phi(Y)]_\Lambda & = & -s^{-1}pI_q, \\ 2\sum_{i=1}^m b_i d_i^T & \equiv & [\mathcal{A}^* R(s)]_X & \equiv & -s^{-1}[\mathcal{A}^*\nabla\Phi(Y)]_X & = & 2Q, \\ \sum_{i=1}^m t_i & \equiv & [\mathcal{A}^* R(s)]_\rho & \equiv & -s^{-1}[\mathcal{A}^*\nabla\Phi(Y)]_\rho & = & V. \end{array}$$

(Here $[\cdot]_\Lambda$, $[\cdot]_X$ and $[\cdot]_\rho$ denote, respectively, the Λ-, the X-, and the ρ-component of the design vector of (TD$_{\rm fn}$).) Note also that Y (and therefore Z) is positive definite. We see that $(R(s), \lambda \equiv -s^{-1}p)$ indeed is a feasible solution of (TD$^*_{\rm fn}$) .

Now, if (Λ, X, ρ) is a feasible solution to (TD$_{\rm fn}$) , and

$$\left(R \equiv \mathrm{Diag}\left\{ \begin{pmatrix} M_i & c_i \\ c_i^T & r_i \end{pmatrix},\ i = 1, ..., m \right\}, \lambda \right)$$

is a feasible solution to $(\mathrm{TD}^*_{\mathrm{fn}})$, then

$$\begin{array}{rcl} 2\,\mathrm{Tr}(QX^T)+V\rho &=& \big[\mathrm{Tr}([\mathcal{A}^*R]_X X^T)+[\mathcal{A}^*R]_\rho\rho+\mathrm{Tr}([\mathcal{A}^*R]_\Lambda\Lambda)\big]-\lambda \\ && \quad [\text{since } [\mathcal{A}^*R]_\Lambda=\lambda I_q,\ [\mathcal{A}^*R]_X=2Q,\ [\mathcal{A}^*R]_\rho=V \text{ by the} \\ && \qquad \text{constraints of } (\mathrm{TD}^*_{\mathrm{fn}}) \text{ and } \mathrm{Tr}\,\Lambda=1 \\ && \qquad \text{by the constraints of } (\mathrm{TD}_{\mathrm{fn}})\,] \\ &=& \mathrm{Tr}(R\mathcal{A}(\Lambda,X,\rho))-\lambda, \end{array}$$

whence

$$[2\,\mathrm{Tr}(QX^T)+V\rho]+\lambda=\mathrm{Tr}(R\mathcal{A}(\Lambda,X,\rho)).$$

Since the optimal values in $(\mathrm{TD}_{\mathrm{fn}})$ and $(\mathrm{TD}^*_{\mathrm{fn}})$, by the Fenchel–Rockafellar duality theorem, are negations of each other, we come to

$$\epsilon[\Lambda,X,\rho]+\epsilon^*[R,\lambda]=\mathrm{Tr}(R\mathcal{A}(\Lambda,X,\rho)); \tag{6.9}$$

here $\epsilon[\Lambda,X,\rho]$ is the accuracy of the feasible solution (Λ,X,ρ) of $(\mathrm{TD}_{\mathrm{fn}})$ (i.e., the value of the objective of $(\mathrm{TD}_{\mathrm{fn}})$ at (Λ,X,ρ) minus the optimal value of the problem), and $\epsilon^*[\cdot]$ is similar accuracy in $(\mathrm{TD}^*_{\mathrm{fn}})$.

Specifying (Λ,X,ρ) as (Λ_s,X_s,ρ_s) and (R,λ) as $(R(s),\lambda_s)$, we make the right-hand side of (6.9) equal to

$$\mathrm{Tr}(R(s)Y)=s^{-1}\,\mathrm{Tr}(ZY)=s^{-1}\,\mathrm{Tr}(Y^{-1}Y)=s^{-1}\mu,$$

and with this equality (6.9) implies (6.8). □

7. Numerical examples. Let us illustrate the developed approach by a few examples.

Example 1. Our first example deals with the toy problem presented in Fig. 2.1; as was explained in section 2.2, here the single-load optimal design results in an unstable truss capable of carrying only very specific loads; the compliance of the truss with respect to the given load is 16.000. Now let us apply approach **B** from section 2.3, where the robustness constraint is imposed before solving the problem and corresponds to "active" nodes—those where the given load is applied. When imposing robustness requirement, we choose Q as explained in section 2.3. Namely, in our case we have 2 fixed and 4 free nodes, so that the dimension n of the space of virtual nodal displacements is $2\times 4=8$. Since all free nodes are active, the ellipsoid of loads in robust setting is full-dimensional ($q=n=8$); this ellipsoid is chosen as explained in section 2.3—one of the half-axes is the given load, and the remaining 7 half-axes are 10 times smaller. The corresponding matrix (rounded to 3 decimal places after the dot) is

$$Q=\begin{pmatrix} 2.000 & 0.014 & -0.026 & 0.117 & -0.063 & 0.170 & -0.264 & -0.054 \\ 0 & 0.235 & 0.216 & 0.125 & -0.032 & -0.161 & -0.070 & 0.104 \\ 0 & -0.040 & -0.107 & 0.099 & 0.311 & -0.158 & -0.117 & -0.035 \\ 2.000 & 0.045 & 0.137 & -0.263 & 0.162 & 0.039 & 0.002 & 0.043 \\ 0 & -0.202 & 0.148 & -0.081 & -0.111 & -0.190 & -0.124 & -0.164 \\ -2.000 & 0.149 & -0.108 & -0.203 & -0.030 & 0.006 & -0.210 & -0.009 \\ -2.000 & -0.089 & 0.219 & 0.057 & 0.129 & 0.203 & -0.052 & -0.003 \\ 0 & 0.173 & 0.028 & 0.020 & 0.042 & 0.035 & 0.098 & -0.341 \end{pmatrix}.$$

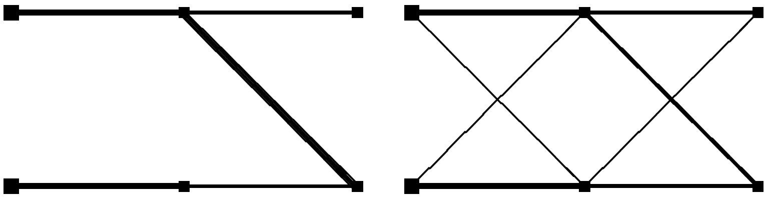


FIG. 7.1. *Optimal design without* (*left*) *and with* (*right*) *robustness constraints.*

TABLE 7.1
Optimal designs for Example 1.

Problem setting	Compliance	Bars, node : node	Bar volumes, %
without robustness constraints	16.000	1 : 2	25.00
		4 : 5	25.00
		3 : 5	25.00
		5 : 6	12.50
		2 : 3	12.50
with robustness constraints	17.400	4 : 5	24.48
		1 : 2	24.48
		3 : 5	23.68
		2 : 3	11.95
		5 : 6	11.95
		2 : 4	1.27
		1 : 5	1.27
		2 : 6	0.92

(To relate Q to the nodal structure presented on Fig. 2.1, note that the coordinates of virtual displacements are ordered as 2X,2Y,3X,3Y,5X,5Y,6X,6Y, where, say, 3X corresponds to the displacement of node #3 along the X-axis.)

The result of "robust" design is presented in Fig. 7.1 and Table 7.1.

Now the maximum over the 8-dimensional loading ellipsoid compliance becomes 17.400 (8.75% growth). But the compliance of the truss with respect to the load f is 16.148; i.e., it is only larger by 0.9% than for the truss given by single-load setting.

Example 2 (Console). The second example deals with approach **A** from section 2.3, where the robustness constraint is used for postoptimality analysis. The left part of Fig. 7.2 represents optimal single-load design for a 9×9 nodal grid on a 2D plane; nodes from the very left column are fixed, the remaining nodes are free, and the load is the unit force acting down and applied at the midnode of the very right column (long arrow). The compliance of the resulting truss with respect to f^*, in appropriate scale, is 1.00. Now note that the compliance of t with respect to very small (of magnitude $0.005\|f^*\|$) "occasional" load (short arrow) applied at properly chosen node is > 8.4 ! Thus, in fact, t is highly unstable.

The right part of Fig. 7.2 represents the truss obtained via postoptimality design with robustness constraint. We marked the nodes incident to the bars of t (there were only 12 of them) and formed a new design problem with the nodal set composed of these marked nodes, and the tentative bars given by all 66 possible pair connections in this nodal set (in the original problem, there were 2040 tentative bars). The truss represented in the right part corresponds to optimal design with robustness constraint imposed at all 10 free nodes of this ground structure in the same way as in the previous example (i.e., the first column in the 20×20 matrix Q is the given load f^*, and the remaining 19 columns formed orthogonal basis in the orthogonal complement to f^* in of 20-dimensional space of virtual displacements of the construction; the Euclidean lengths of these additional columns were set to 0.1 (10% of the magnitude of f^*).

The maximal compliance, over the resulting ellipsoid of loads, of the "robust" truss is now 1.03, and its compliance with respect to f is 1.0024—i.e., it is only larger by 0.24% than the optimal compliance c^* given by the single-load design; at the same time, the compliance of the new truss with respect to all "occasional" loads of magnitude 0.1 is at most by 3% greater than c^*.

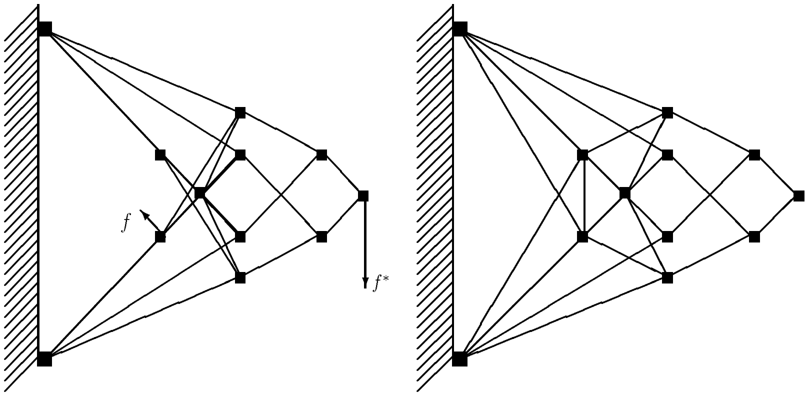


FIG. 7.2. *Single-load optimal design* (*left*) *and its postoptimal "robust correction"* (*right*).

Example 3 ($N\times 2$-truncated pyramids). The examples below deal with simple 3D trusses. The nodal set is composed of $2N$ points. N "ground" nodes are the vertices of equilateral N-polygon in the plane $z = 0$:

$$x_i = \cos(2\pi i/N),\ y_i = \sin(2\pi i/N),\ z_i = 0,\ \ i = 1, \dots, N,$$

and N "top" nodes are the vertices of twice smaller concentric polygon in the plane $z = 2$:

$$x_i = \frac{1}{2}\cos(2\pi i/N),\ y_i = \frac{1}{2}\sin(2\pi i/N),\ z_i = 2,\ \ i = N+1, \dots, 2N.$$

The ground nodes are fixed, and the top ones are free. The ground structure is composed of all pair connections of the nodes, except connections between the ground-fixed ones.

We dealt with two kinds of loading scenarios, referred to, respectively, as "$N\times$2s"- and "$N\times$ 2m"-design data. $N\times$ 2s-data corresponds to a singleton scenario set, where the load is composed of N nearly horizontal forces acting at the top nodes and "rotating" the construction. The force acting at the ith node, $i = N + 1, ..., 2N$, is

$$f_i = \alpha(\sin(2\pi i/N), -\cos(2\pi i/N), -\rho),\ \ i = N+1, \dots, 2N, \tag{7.1}$$

where ρ is a small parameter and α is a normalizing coefficient which makes the Euclidean length of the load equal to 1 (i.e., $\alpha = 1/\sqrt{N(1+\rho^2)}$). $N\times$ 2m-data correspond to N-scenario design where the forces (7.1) act nonsimultaneously (and are renormalized to be of unit length, i.e., $\alpha = 1/\sqrt{1+\rho^2}$).

Along with the traditional "scenario design" (single load in the case of s-data and multiload in the case of m-data), we carried out "robust design" where we minimized

Table 7.2
Compliances in Example 3.

Design data	Scenario design			Robust design	
	Compl(Scen)	Compl(0.1)	Compl(0.3)	Compl(Scen)	Compl(0.3)
3x2s	1.0000	7.5355	67.820	1.0029	1.0029
4x2s	1.0000	12.209	109.88	1.0028	1.0028
5x2s	1.0000	2.7311	24.580	1.0022	1.0022
3x2m	1.0000	1.2679	1.2679	1.0942	1.0943
4x2m	1.0000	4.1914	37.722	1.2903	1.2903
5x2m	1.0000	1.5603	1.6882	1.5604	1.5604

the maximum compliance with respect to a full-dimensional ellipsoid of loads M_θ—the "ellipsoidal envelope" of the unit ball in the linear span $L(F)$ of the scenario loads and the ball of radius θ in the orthogonal complement of $L(F)$ in the $3N$-dimensional space of virtual displacements of the nodal set. In other words, dim $L(F)$ of the principal half-axes of M_θ are of unit length and span $L(F)$, and the remaining principal half-axes are of length θ. In our experiments with robust design, we used $\theta = 0.3$ and measured the worst-case compliance of the resulting trusses, same as those given by the usual scenario design, with respect to three sets of loads:

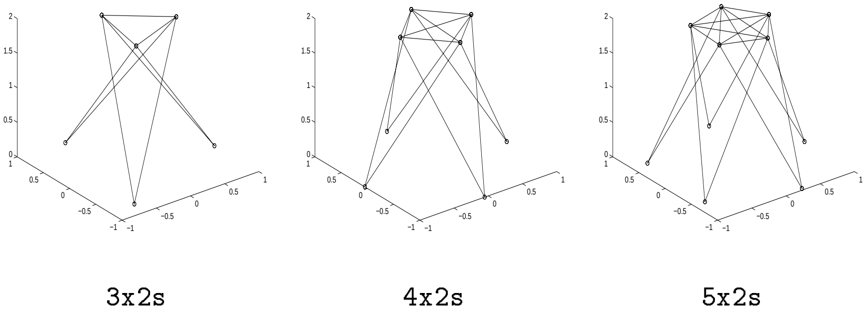
(i) the original set of scenarios,
(ii) the ellipsoid of loads $M_{0.1}$,
(iii) the ellipsoid of loads $M_{0.3}$.

The resulting structures are shown in Fig. 7.3 (data $N \times 2$s) and Fig. 7.4 (data $N \times 2$m), and the corresponding compliances are seen in Table 7.2. In Table 7.2, Compl(Scen) means the maximum compliance of the designed structure with respect to the set of loading scenarios given by the corresponding data, while Compl(θ), $\theta = 0.1, 0.3$ is the maximum compliance with respect to the ellipsoid M_θ. In order to make the comparison more clear, we normalize the data in each row to make the compliance of the truss given by scenario design with respect to the underlying set of scenarios equal to 1.

The summary of the numerical results in question is as follows.

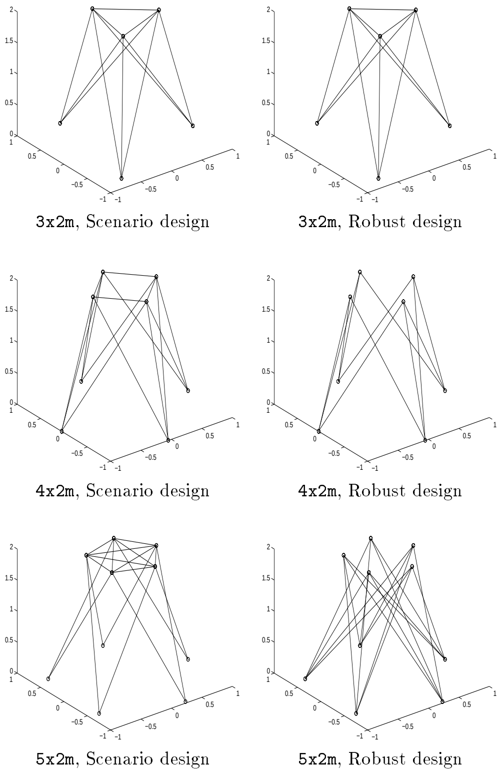
1. $N \times 2$s *design data.* The trusses given by the scenario and the robust designs have the same topology and differ only in bar volumes; the difference basically is in the thickness of the "top" – horizontal – bars (see Fig. 7.3): for the "robust" truss they are approximately 80 times larger in volume than for the "scenario" one (0.1% of the total bar volume instead of 0.0012% for $N = 3$). Although this difference in sizing seems small, it is in fact quite significant. The scenario design results in highly unstable constructions: appropriately chosen "occasional" loads with magnitude only 10% of the scenario load, result in 2.6–13.0 times larger compliance than the "scenario" one. When the occasional load is allowed to be 30% of the scenario one, the ratio in question may become 15–100. Note that bad robustness of the trusses given by the scenario design has very simple origin: in the limiting case of $\rho = 0$ (purely horizontal rotating load—the torque) the top bars disappear at all, and the optimal truss given by the usual single-load design becomes completely unstable.

The robust design associated with the ellipsoid $M_{0.3}$ ("occasional" loads may be as large as 30% of the scenario one) results in trusses nearly optimal with respect to the scenario load ("nonoptimality" is at most 0.3%). Surprisingly enough, for the trusses given by the robust design the maximum compliance with respect to the ellipsoid of loads is the same as their compliance with respect to the scenario load. Thus, in the case in question, the robustness is "almost costless."



3x2s 4x2s 5x2s

FIG. 7.3. *Scenario and robust design, single "rotating" load* ($\rho = 0.001$ *for* 3×2s *and* 4×2s, $\rho = 0.01$ *for* 5×2s).



3x2m, Scenario design 3x2m, Robust design

4x2m, Scenario design 4x2m, Robust design

5x2m, Scenario design 5x2m, Robust design

FIG. 7.4. *Scenario design vs. robust design, multiple "rotating" loads* ($\rho = 0.001$ *for* 3×2m *and* 4×2m, $\rho = 0.01$ *for* 5×2m).

2. $N \times 2$m *design data.* Here the trusses given by the scenario design are of course much more stable than in the case of $N \times 2$s data, and both kinds of design possess their own advantages and drawbacks. On one hand, the maximum compliance, over the ellipsoid $M_{0.3}$ of loads, of the truss given by the scenario design is considerably larger than the optimal value of this quantity (by 27% for $N = 3$, by 3670 % for $N = 4$,[5] and by 69% for $N = 5$). On the other hand, the maximum compliance, over the scenario set, of the truss given by robust design is also considerably larger than the optimal value of this quantity (by 9% for $N = 3$, by 29% for $N = 4$, and by 56% for $N = 5$). Thus, it is difficult to say which design—the scenario or the robust one—results in better construction.

The results in question suggest a seemingly better approach to ensuring robustness than those mentioned in section 2.3, namely, as follows. Given a scenario set F, we embed it into an ellipsoid M (see section 2.3) and solve the resulting problem ($\mathrm{TD}_{\mathrm{robust}}$); let c^*_{robust} be the corresponding optimal value. After this value is found, we increase it in certain fixed proportion $1+\chi$, say, by 10%, and solve the problem

$$\begin{array}{c}\text{find } t \in T \text{ which minimizes the compliance } c^F(t) = \max_{f\in F} c_f(t)\\ \text{s.t.}\quad c^M(t) \equiv \max_{f\in M} c_f(t) \le (1+\chi)c^*_{\mathrm{robust}}.\end{array}$$

Note that the latter problem can be posed as a semidefinite program, which only slightly differs from ($\mathrm{TD}_{\mathrm{sd}}$):

$$\begin{array}{rcl} & \min\limits_{t\in R^m, \tau\in R} \tau & \\ \text{s.t.} & & \\ \begin{pmatrix} \tau & f^T \\ f & \sum_{i=1}^m t_i A_i \end{pmatrix} & \ge & 0,\ \forall f \in F \\ \begin{pmatrix} a & Q^T \\ Q & \sum_{i=1}^m t_i A_i \end{pmatrix} & \ge & 0, \end{array}$$

where

$$a = (1+\chi)c^*_{\mathrm{robust}}.$$

The dual to the latter problem is the computationally more convenient program

$$\begin{array}{rcl} \min \left\{ a\,\mathrm{Tr}(\Lambda) + 2\,\mathrm{Tr}(QX^T) + 2\sum\limits_{f\in F} f^T x_f + V\rho \right\} & & \\ \text{s.t.} & & \\ \begin{pmatrix} \Lambda & X^T b_i \\ b_i^T X & \sigma_i \end{pmatrix} & \ge & 0,\ i = 1,\dots,m, \\ \sigma_i + \sum\limits_{f\in F} \frac{(b_i^T x_f)^2}{\lambda_f} & \le & \rho,\ i = 1,\dots,m, \\ \lambda_f & \ge & 0,\ f \in F, \\ \sum\limits_{f\in F}^{k} \lambda_f & = & 1, \end{array}$$

[5] This huge difference mainly comes not from the difference in the topology of trusses but from different sizing of the bars linking bottom nodes with "the same" top ones; for the robust design these bars are approximately 30 times thicker than for the scenario design (1.5% of the total bar volume vs. 0.05%, resp.).

Table 7.3
Computational performance.

Problem	Scenario design			Robust design		
	$(N_{\rm dsg}, N_{\rm LMI}, N_{\rm img})$	Nwt	CPU	$(N_{\rm dsg}, N_{\rm LMI}, N_{\rm img})$	Nwt	CPU
Example 2	(146,2041,6121)	75	$3'58''$	(611,67,15247)	95	$24'42''$
3x2s	(11,13,37)	14	$0.2''$	(127,13,661)	62	$14.5''$
4x2s	(14,23,67)	16	$0.4''$	(223,23,2003)	77	$1'18''$
5x2s	(17,36,106)	17	$0.6''$	(346,36,4761)	59	$3'13''$
3x2m	(31,13,121)	16	$0.4''$	(127,13,661)	101	$24''$
4x2m	(53,23,331)	23	$1.5''$	(223,23,2003)	65	$1'6''$
5x2m	(81,36,736)	23	$3''$	(346,36,4761)	65	$3'32''$

In the table:

$N_{\rm dsg}$ – number of design variables in $({\rm TD_{fn}})$,
$N_{\rm LMI}$ – number of linear matrix inequalities in $({\rm TD_{fn}})$,
$N_{\rm ing}$ – total image dimension of $({\rm TD_{fn}})$, i.e., the dimension of the corresponding semidefinite cone,
Nwt – number of Newton steps performed by the interior point solver when solving $({\rm TD_{fn}})$,
CPU – solution time (workstation RS 6000).

the design variables being $\Lambda \in \mathbf{S}^k$, $X \in R^{n\times q}$, $\sigma \in R^n$, $\{(\lambda_f, x_f) \in R \times R^n\}_{f\in F}$, and $\rho \in R$.

The reported numerical experiments were carried out with the LMI Control Toolbox [7], the only software for semidefinite programming available to us at the moment. The projective interior point method [10, Chapter 5], implemented in the Toolbox is of the potential reduction rather than of the path-following type, and we were forced to add to the Toolbox solver a "centering" interior point routine which transforms a good approximate solution to $({\rm TD_{fn}})$ into another solution of the same quality belonging to the central path, which enabled us to recover the optimal truss, as is explained in section 6. The time of solving $({\rm TD_{fn}})$ by the Toolbox solver was moderate, as it is seen in Table 7.3.

8. Concluding remarks. Uncertainty of the data is a generic property associated with optimization problems of real world origin. Accordingly, "robust reformulation" of an optimization model as a way to improve applicability of the resulting solution is a very traditional idea in mathematical programming, and different approaches to implement this idea were proposed. One of the best-known approaches is *stochastic programming*, where uncertainty is assumed to be of stochastic nature. Another approach is *robust optimization* (see [9] and references therein); here, roughly speaking, the "robust solution" should not necessarily be feasible for all "allowed" data, and the "optimal robust solution" minimizes the sum of the original objective and a penalty for infeasibilities, the infeasibilities being taken over a finite set of scenarios. The approach used in our paper is somewhat different: a solution to the "stabilized" problem should be feasible for all allowed data. This approach is exactly the one used in robust control. The goal of this concluding section is to demonstrate that the approach developed in the paper can be naturally extended to other mathematical programming problems. To this end let us look at what in fact was done in section 2.

(ii) We start with an optimization program in the "conic" form

$$({\rm P}) \qquad c^T u \to \min \mid \ Au \in K, \ u \in E,$$

where u is the design vector, A is $M \times N$ matrix, K is closed convex cone in R^M, and

E is an affine plane in R^N.

This is exactly the form of a single-load TTD problem $\min\{\sigma \mid \sigma \geq c_f(t), t \in T\}$ (see section 2.1): to cast TTD as (P) it suffices to specify (P) as follows:

- $u = (t, \tau, \sigma) \in R^m \times R \times R$;
- $E = \{(t, \tau, \sigma) \mid \tau = 1, \sum_{i=1}^m t_i = V\}$;
- K is the direct product of the cone of positive semidefinite symmetric $(n+1) \times (n+1)$ matrices ("matrix part") and R_+^m ("vector part");
- the "vector" part of the linear mapping $(t, \tau, \sigma) \mapsto A(t, \tau, \sigma)$ is t, and the "matrix" part is $\begin{pmatrix} \sigma & \tau f^T \\ \tau f & A(t) \end{pmatrix}$, f being the load in question.

(ii) We say that the data in (P)(entries in the data matrix A) are inexact (in TTD, these are entries associated with the load vector f). We model the corresponding uncertainty by the assumption that $A \in \mathcal{U}$, where $\mathcal{U}$ is certain ellipsoid in the space of $M \times N$ matrices.[6] Accordingly, we impose on the decision u the requirement to be *robust feasible*, i.e., to satisfy the inclusions $u \in E$ and $Au \in K$ for *all* possible data matrices $A \in \mathcal{U}$. This leads to our *robust reformulation* of (P):

$$(\mathrm{P}_{\mathrm{st}}) \qquad c^T u \to \min \mid u \in E,\ Au \in K\ \forall A \in \mathcal{U}.$$

Note that this is a general form of the approach we have used in section 2; and the goal of the remaining sections was to realize, for the case when (P) is the single load TTD problem, what is $(\mathrm{P}_{\mathrm{st}})$ as a mathematical programming problem and how to solve it efficiently.

Problem (P) is a quite general form of a convex programming problem; the advantage of this conic form is that it allows to separate the "structure" of the problem (c, K, E) and the "data" (A).[7] The data now become a quite tractable entity—simply a matrix. Whenever a program in question can be naturally posed in the conic form, we can apply the above approach to get a "robust reformulation" of (P). Let us look at some concrete examples.

Robust linear programming. Let K in (P) be the nonnegative orthant; this is exactly the case when (P) is a linear programming problem in the canonical form.[8] It is shown in [6] that $(\mathrm{P}_{\mathrm{st}})$ is a conic quadratic program (i.e., a conic program with K being a direct product of the second order cones).

Robust quadratic programming. Let K be a direct product of the second order cones, so that (P) is a conic quadratic program (a natural extension of the usual quadratically constrained convex quadratic program). It can be verified (see [6]) that in this case, under mild restrictions on the structure of the uncertainty ellipsoid $\mathcal{U}$, the problem $(\mathrm{P}_{\mathrm{st}})$ can be equivalently rewritten as a semidefinite program (a conic program with K being the cone of positive semidefinite symmetric matrices).

Note that in these examples $(\mathrm{P}_{\mathrm{st}})$ is quite tractable computationally, in particular, it can be efficiently solved by interior point methods.

[6] Here, as in the main body of the paper, a k-dimensional ellipsoid in R^M is, by definition, the image of the unit Euclidean ball in R^k under an affine embedding of R^k into R^M.

[7] In some applications, the objective c should be treated as a part of the data rather than the structure. One can easily reduce this case to the one in question by evident equivalent reformulation of (P).

[8] Up to the fact that the mapping $u \mapsto Au$ is assumed to be linear rather than affine. This assumption does not restrict generality, since we incorporate into the model the affine constraint $u \in E$; at the same time, the homogeneous form $Au \in K$ of the nonnegativity constraints allows us to handle both uncertainties in the matrix of the linear inequality constraints and those in the right-hand side vector.

A somewhat "arbitrary" element in the outlined general approach is that we model uncertainty as an *ellipsoid*. Note, anyhow, that *in principle* the above scheme can be applied to any other uncertainty set $\mathcal{U}$, and the actual "bottleneck" is our ability to solve efficiently the resulting problem $(\mathrm{P}_{\mathrm{st}})$. Note that the robust problem $(\mathrm{P}_{\mathrm{st}})$ always is convex, so that there is a sufficient condition for its "efficient solvability." The condition, roughly speaking (for the details, see [8]), is that we should be able to equip the feasible domain

$$G = \{u \mid u \in E, Au \in K \ \forall A \in \mathcal{U}\}$$

of $(\mathrm{P}_{\mathrm{st}})$ with a *separation oracle*—a "computationally efficient" routine which, given on input u, reports on output whether $u \in G$, and if it is not the case, returns a linear form which separates G and u. Whether this sufficient condition is satisfied or not depends on the geometry of $\mathcal{U}$ and K, and the "more complicated" $\mathcal{U}$ is, the "simpler" K should be. When $\mathcal{U}$ is very simple (a polytope given as a convex hull of a finite set), K could be an arbitrary "tractable" cone (one which can be equipped with a separation oracle); when $\mathcal{U}$ is an ellipsoid, K could be for sure the nonnegative orthant or a direct product of the second order cones. On the other hand, if K is simple (the nonnegative orthant, as in the linear programming case), $\mathcal{U}$ could be more complicated than an ellipsoid—e.g., it could be an intersection of finitely many ellipsoids. Under mild regularity assumptions, in the latter case $(\mathrm{P}_{\mathrm{st}})$ turns out to be a conic quadratic program [6]. In other words, there is a "tradeoff" between the *flexibility* and the *tractability*, i.e., between the ability to express uncertainties, on one hand and the ability to produce computationally tractable problems $(\mathrm{P}_{\mathrm{st}})$ on the other hand.

We strongly believe that the approach advocated here is promising and is worthy of investigation, and we intend to devote a separate paper to it.

REFERENCES

[1] W. Achtziger, M. P. Bendsøe, A. Ben-Tal, and J. Zowe, *Equivalent displacement-based formulations for maximum strength truss topology design*, Impact Comput. Sci. Eng., 4 (1992), pp. 315–345.

[2] M. P. Bendsøe, A. Ben-Tal, and J. Zowe, *Optimization methods for truss geometry and topology design*, Structural Optimization, 7 (1994), pp. 141–159.

[3] A. Ben-Tal and M. P. Bendsøe, *A new method for optimal truss topology design*, SIAM J. Optim., 3 (1993), pp. 322–358.

[4] A. Ben-Tal, M. Kočvara, and J. Zowe, *Two nonsmooth approaches to simultaneous geometry and topology design of trusses*, Topology Design of Structures, Proc. NATO-ARW, M. P. Bendsøe, ed., Sesimbra, Portugal, 1992.

[5] A. Ben-Tal and A. Nemirovski, *Potential reduction polynomial time method for Truss Topology Design*, SIAM J. Optim., 4 (1994), pp. 596–612.

[6] A. Ben-Tal and A. Nemirovski, *Robust Convex Programming*, manuscript, Optimization Laboratory, Faculty of Industrial Engineering and Management at Technion, Haifa, Israel, 1995.

[7] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, *LMI Control Toolbox*, The Mathworks Partner Series, The MathWorks Inc., Natick, MA, 1995.

[8] M. Grötschel, L. Lovasz, and A. Schrijver, *The Ellipsoid Method and Combinatorial Optimization*, Springer-Verlag, Heidelberg, 1988.

[9] J. M. Mulvey, R. J. Vanderbei, and S. A. Zenios, *Robust optimization of large-scale systems*, Oper. Res., 43 (1995), pp. 264–281.

[10] Yu. Nesterov and A. Nemirovski, *Interior Point Polynomial Methods in Convex Programming*, SIAM Series in Applied Mathematics, Philadelphia, PA, 1994.

[11] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[12] G. Rozvany, M. P. Bendsøe, and U. Kirsch, *Layout optimization of structures*, Appl. Mech. Rev., 48 (1955), pp. 41–119.

SIAM J. OPTIM.
Vol. 7, No. 4, pp. 1017–1036, November 1997

AN EFFICIENT ALGORITHM FOR MINIMIZING A SUM OF EUCLIDEAN NORMS WITH APPLICATIONS[*]

GUOLIANG XUE[†] AND YINYU YE[‡]

Abstract. In recent years rich theories on polynomial-time interior-point algorithms have been developed. These theories and algorithms can be applied to many nonlinear optimization problems to yield better complexity results for various applications. In this paper, the problem of minimizing a sum of Euclidean norms is studied. This problem is convex but not everywhere differentiable. By transforming the problem into a standard convex programming problem in conic form, we show that an ϵ-optimal solution can be computed efficiently using interior-point algorithms. As applications to this problem, polynomial-time algorithms are derived for the Euclidean single facility location problem, the Euclidean multifacility location problem, and the shortest network under a given tree topology. In particular, by solving the Newton equation in linear time using *Gaussian elimination on leaves of a tree*, we present an algorithm which computes an ϵ-optimal solution to the shortest network under a given full Steiner topology interconnecting N regular points, in $O(N\sqrt{N}(\log(\bar{c}/\epsilon)+\log N))$ arithmetic operations where $\bar{c}$ is the largest pairwise distance among the given points. The previous best-known result on this problem is a graphical algorithm which requires $O(N^2)$ arithmetic operations under certain conditions.

Key words. polynomial time, interior-point algorithm, minimizing a sum of Euclidean norms, Euclidean facilities location, shortest networks, Steiner minimum trees

AMS subject classifications. 68Q20, 68Q25, 90C25, 90C35

PII. S1052623495288362

1. Introduction.

The motivation to write this paper was to apply new techniques—polynomial time interior-point algorithms for convex programming—to solve two old problems: the Euclidean facilities location problem and the Steiner minimal tree (SMT) problem. The first problem, studied by researchers in location science, has applications in transportation and logistics. The second problem, studied by researchers in combinatorial optimization, has applications in communication networks. Both problems can be described as the minimization of a sum of Euclidean norms and they both trace back to an ancient problem studied by Fermat in the 17th century.

At the end of his celebrated essay on maxima and minima, in which he presented precalculus rules for finding tangents to a variety of curves, Fermat threw out this challenge: "Let he who does not approve of my method attempt the solution of the following problem: Given three points in the plane, find a fourth point such that the sum of its distances to the three given points is at minimum!" The solution to the original Fermat problem is either the Torricelli point—an interior point which opens an angle of 120° to each of the three sides of the triangle—or one of the given points whose inner angle is no less than 120°. This problem has been generalized into the Euclidean facilities location problem and the SMT problem.

The facilities location problem is one of locating N new facilities with respect

[*] Received by the editors June 28, 1995; accepted for publication April 30, 1996.
http://www.siam.org/journals/siopt/7-4/28836.html

[†] Department of Computer Science and Electrical Engineering, The University of Vermont, Burlington, VT 05405-0156 (xue@cs.uvm.edu). The research of this author was supported in part by the US Army Research Office grant DOD DAAH04-96-1-0233 and by the National Science Foundation grants NSF ASC-9409285 and NSF OSR-9350540.

[‡] Department of Management Sciences, The University of Iowa, Iowa City, IA 52242-1000 (yyye@dollar.biz.uiowa.edu). The research of this author was supported in part by NSF grant DDM-9207347 and the University of Iowa Business School Summer Grant.

to M existing facilities, the locations of which are known. The problem consists of finding locations of new facilities which will minimize a total cost function. This total cost function consists of a sum of costs directly proportional to the distances between the new facilities and costs directly proportional to the distances between new and existing facilities. If there is only one new facility ($N = 1$), the problem is called a Euclidean single facility location (ESFL) problem. If there is more than one new facility ($N \geq 2$), the problem is called a Euclidean multifacility location (EMFL) problem.

For the general ESFL problem, Weiszfeld [30] gave a simple closed form iterative algorithm in 1937. Later, it was proved by numerous authors [18, 24, 29] that the algorithm converges globally and, under certain conditions, linearly. Chandrasekaran and Tamir [5, 6] exhibited a solution to the strong separation problem associated with the ESFL problem which shows that an *ϵ-optimal solution* (i.e., a feasible solution whose absolute error in the objective function is within ϵ to the optimal objective function value) to the ESFL problem can be constructed in polynomial time using the ellipsoid method.

Miehle [21] was the first to propose an extension of the Weiszfeld algorithm for ESFL to solve EMFL problems. Ostresh [24] proved that Miehle's algorithm is a descending one. However, Miehle's algorithm may converge to a nonoptimal point; see [26, 31]. Eyster, White, and Wierwille [9] proposed a hyperboloid approximation procedure (HAP) for solving the perturbed EMFL problem. Rosen and Xue [31, 27] proved that the HAP always converges from any initial point. Calamai and Conn [3, 4] and Overton [25] proposed projected Newton algorithms for minimizing a sum of Euclidean norms and proved that the algorithms have quadratic rate of convergence provided the sequence of points generated by the algorithm converges to a strong minimizer. For more details, see the books by Francis, McGinnis, and White [11] and by Love, Morris, and Wesolowsky [19].

Recently, Xue, Rosen, and Pardalos [32] showed that the dual of the EMFL problem is the minimization of a linear function subject to linear and convex quadratic constraints and can therefore be solved by the interior-point techniques in polynomial time. den Hertog [8] (and see references therein) also presented a polynomial-time interior-point Newton barrier method for solving (2.1). More recently, Andersen [1] used the HAP idea [9] to smooth the objective function by introducing a perturbation $\epsilon > 0$ and applied a Newton barrier method to solving the problem. Andersen and Christiansen [2] and Conn and Overton [7] also proposed a primal–dual method based on the ϵ-perturbation and presented impressive computational results, although no complexity result is established for their method at this moment. None of the above formulations is in conic form.

The SMT problem [12, 20] is concerned with interconnecting a set of given points on the Euclidean plane with a shortest network. The shortest network is always a tree network and may contain some additional points called Steiner points. The SMT problem is NP-hard. Recently, there have been increased interests in the computation of a shortest interconnection network after the connections among the points (which is called a *topology*, to be defined in section 6.3) are specified. Hwang [14] proposed a linear-time algorithm for computing the shortest network under a full Steiner topology when the shortest network is a nondegenerate full Steiner tree. Hwang and Weng [15] proposed an $O(N^2)$ arithmetic operation algorithm for computing the shortest network under a Steiner topology when the shortest network is a tree whose vertex degrees are all less than or equal to 3. Smith [28] used an EMFL approach to compute the shortest network under a given topology. His algorithm is essentially a first-order method.

In this paper, we first transform the basic problem of minimizing a sum of Euclidean norms into a standard convex programming problem in conic form and present an interior-point algorithm that can compute an ϵ-optimal solution in $O(\sqrt{m}(\log(\bar{c}/\epsilon)+\log m))$ iterations, where m is the number of norms in the summation and $\bar{c}$ is a constant that is not less than the Euclidean norm of any of the given vectors $c_i, i = 1, 2, \ldots, m$. We then study several applications of the basic problem and show improved computational complexity results wherever possible. In particular, we show that an ϵ-optimal solution to the shortest network under a given tree topology for a set of N points can be computed in $O(N\sqrt{N}(\log(\bar{c}/\epsilon)+\log N))$ arithmetic operations where $\bar{c}$ is the largest pairwise distance among the given points.

The rest of this paper is organized as follows. In section 2, we describe the basic problem of minimizing a sum of Euclidean norms. In section 3, the basic problem is transformed into a standard convex programming problem in conic form. In section 4, we present a primal–dual potential reduction algorithm for solving the problem. In section 5, we discuss the computational complexity and simplifications of the potential reduction algorithm. In section 6, we present applications to the ESFL problem, the EMFL problem, and the SMT problem. In section 7, we present some computational examples of SMT problems. We conclude this paper in section 8.

2. Minimizing a sum of Euclidean norms.

Let $c_1, c_2, \ldots, c_m \in R^d$ be column vectors in the Euclidean d-space and $A_1, A_2, \ldots, A_m \in R^{n\times d}$ be n-by-d matrices with each having full column rank. We want to find a point $u \in R^n$ such that the following sum of Euclidean norms is minimized:

$$\begin{array}{ll} \min & \sum_{i=1}^m ||c_i - A_i^T u|| \\ \text{s.t.} & u \in R^n. \end{array} \tag{2.1}$$

It is clear that $u = 0$ is an optimal solution to (2.1) when all of the c_i are zero. Therefore, we will assume in the rest of this paper that not all of the c_i are zero. Problem (2.1) is a convex programming problem, but its objective function is not everywhere differentiable. Two special cases of this problem are the Euclidean facilities location problem and the SMT problem under a given topology.

We will call problem (2.1) the *basic problem* in the rest of our paper. This problem can be formulated as the maximization of a linear function subject to affine and convex cone constraints as follows:

$$\begin{array}{ll} \max & -\sum_{i=1}^m t_i \\ \text{s.t.} & t_1 \geq ||c_1 - A_1^T u||, \\ & t_2 \geq ||c_2 - A_2^T u||, \\ & \vdots \\ & t_m \geq ||c_m - A_m^T u||, \end{array} \tag{2.2}$$

where $t_i \in R, \quad i = 1, 2, \ldots, m$.

Problem (2.1) and problem (2.2) are equivalent in the following sense. If $(t_1; t_2; \cdots; t_m; u)$ is the optimal solution to (2.2), then u is the optimal solution to (2.1). If u is the optimal solution to (2.1), then $(t_1; t_2; \cdots; t_m; u)$ is the optimal solution to (2.2), where $t_i = ||c_i - A_i^T u||, \quad i = 1, 2, \ldots, m$ and $(t_1; t_2; \cdots; t_m; u)$ is an $(m+n)$-dimensional column vector whose first m elements are $t_i, \quad i = 1, 2, \ldots, m$ and whose last n elements are the elements of u.

In the rest of this paper, when we represent a large matrix with several small matrices, we will use semicolons ";" for column concatenation and commas "," for row concatenation. This notation also applies to vectors. We will use 0_n to represent an n-dimensional column vector whose elements are all zero. We will also use I_d to represent the d-by-d identity matrix.

3. Conic formulation. In this section, we will transform our basic problem (2.1) into a standard convex programming problem in conic form, where the cone and its associated barrier are *self-scaled* (or *homogeneous and self-dual*); see Nesterov and Nemirovskii [22], Nesterov and Todd [23], and Güler [13]. Because of the special constraints in problem (2.2), the cone of our choice is the second-order cone or the Lorentz cone. For definitions and theory about the second-order cone, self-scaled barriers, and related theory, see [22, 23, 13].

Let the cone be

$$K := \{(t; s) \in R^{d+1} : \ t \geq \|s\|\}.$$

Then its interior is

$$\text{int}K := \{(t; s) \in R^{d+1} : \ t > \|s\|\}.$$

Let

$$\delta(t; s) = \sqrt{t^2 - \|s\|^2},$$

and

$$f(t; s) = -\log \delta^2(t; s).$$

Then, for any $(t; s) \in \text{int}K$ we have

$$f'(t; s) = \frac{2}{\delta^2(t; s)} \begin{pmatrix} -t \\ s \end{pmatrix}$$

and

$$f''(t; s) = \frac{2}{\delta^2(t; s)} \begin{pmatrix} -1 & 0 \\ 0 & I_d \end{pmatrix} + \frac{4}{\delta^4(t; s)} \begin{pmatrix} t^2 & -ts^T \\ -ts & ss^T \end{pmatrix}, \tag{3.1}$$

which is positive definite. Its inverse is

$$(f''(t; s))^{-1} = \frac{\delta^2(t; s)}{2} \begin{pmatrix} -1 & 0 \\ 0 & I_d \end{pmatrix} + \begin{pmatrix} t^2 & ts^T \\ ts & ss^T \end{pmatrix}. \tag{3.2}$$

Also note that

$$(f''(t; s))^{-1} f'(t; s) = -(t; s). \tag{3.3}$$

Now let

$$\mathcal{B} = \begin{pmatrix} -1 \\ -1 \\ \vdots \\ -1 \\ 0_n \end{pmatrix} \in R^{m+n}, \quad \mathcal{C} = \begin{pmatrix} (0; c_1) \\ (0; c_2) \\ \vdots \\ (0; c_m) \end{pmatrix} \in R^{m+md},$$

and

$$\mathcal{A}^T = \begin{pmatrix} -1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & A_1^T \\ 0 & -1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & A_2^T \\ & & \ddots & & \\ 0 & 0 & \cdots & -1 & 0 \\ 0 & 0 & \cdots & 0 & A_m^T \end{pmatrix} \in R^{(m+md)\times(m+n)}.$$

Then, problem (2.1) or (2.2) can be written in the standard (dual) form

$$\begin{array}{ll} \max & \mathcal{B}^T(t_1;t_2;\cdots;t_m;u) \\ \text{s.t.} & \begin{pmatrix} (t_1;s_1) \\ (t_2;s_2) \\ \vdots \\ (t_m;s_m) \end{pmatrix} = \mathcal{C} - \mathcal{A}^T(t_1;t_2;\cdots;t_m;u), \\ & (t_i;s_i) \in K,\ i=1,2,\ldots,m. \end{array} \tag{3.4}$$

Let $(\tau_1;x_1),(\tau_2;x_2),\ldots,(\tau_m;x_m) \in R^{d+1}$. Then its corresponding primal problem is

$$\begin{array}{ll} \min & \mathcal{C}^T((\tau_1;x_1);(\tau_2;x_2);\cdots;(\tau_m;x_m)) \\ \text{s.t.} & \mathcal{A}((\tau_1;x_1);(\tau_2;x_2);\cdots;(\tau_m;x_m)) = \mathcal{B}, \\ & (\tau_i;x_i) \in K,\ i=1,2,\ldots,m. \end{array} \tag{3.5}$$

Thus, using $\mathcal{X} := ((\tau_1;x_1);(\tau_2;x_2);\cdots;(\tau_m;x_m))$, $\mathcal{S} := ((t_1;s_1);(t_2;s_2);\cdots;(t_m;s_m))$, $\mathcal{Y} := (t_1;t_2;\cdots;t_m;u)$, and $\mathcal{K} := K^m := K \times K \times \cdots \times K$, we can write the two problems (3.5) and (3.4) as

$$(P)\qquad \begin{array}{ll} \min & \mathcal{C}^T\mathcal{X} \\ \text{s.t.} & \mathcal{A}\mathcal{X} = \mathcal{B}, \\ & \mathcal{X} \in \mathcal{K} \end{array}$$

and

$$(D)\qquad \begin{array}{ll} \max & \mathcal{B}^T\mathcal{Y} \\ \text{s.t.} & \mathcal{S} = \mathcal{C} - \mathcal{A}^T\mathcal{Y}, \\ & \mathcal{S} \in \mathcal{K}. \end{array}$$

This is the pair of problems (P) and (D) in Nesterov and Nemirovskii [22] and Nesterov and Todd [23]. Since K is a convex self-dual and self-scaled cone with $\nu = 2$, $\mathcal{K}$ is a convex self-dual and self-scaled cone with $\nu = 2m$. Thus, we can use an interior-point algorithm to compute an ϵ-optimal solution of the problem in polynomial time.

Kojima [16] recently pointed out to us that problem (2.2) can be formulated as a positive semidefinite program:

$$\begin{array}{ll} \max & -\sum_{i=1}^m t_i \\ \text{s.t.} & \begin{pmatrix} t_i & (c_i - A_i^T u)^T \\ (c_i - A_i^T u) & t_i I_d \end{pmatrix} \text{ positive semidefinite for} \\ & i=1,2,\ldots,m. \end{array} \tag{3.6}$$

However, as we will illustrate later, the complexity bound for solving positive semidefinite program (3.6) will be $\sqrt{d}$ factor higher than that for solving problem (3.4).

4. A primal–dual potential reduction algorithm. Let

$$F(\mathcal{X}) = \sum_{i=1}^{m} f(\tau_i; x_i) \quad \text{and} \quad F(\mathcal{S}) = \sum_{i=1}^{m} f(t_i; s_i). \tag{4.1}$$

A primal–dual potential function for the pair (P) and (D) is

$$\phi_\rho(\mathcal{X}, \mathcal{S}) := \rho \log(\langle \mathcal{X}, \mathcal{S} \rangle) + F(\mathcal{X}) + F(\mathcal{S}), \tag{4.2}$$

where $\rho = 2m + \gamma\sqrt{2m}$, $\gamma \geq 1$. Note that

$$\langle \mathcal{X}, \mathcal{S} \rangle = \mathcal{X}^T \mathcal{S} = \mathcal{C}^T \mathcal{X} - \mathcal{B}^T \mathcal{Y}$$

and

$$\phi_{2m}(\mathcal{X}, \mathcal{S}) := 2m \log(\langle \mathcal{X}, \mathcal{S} \rangle) + F(\mathcal{X}) + F(\mathcal{S}) \geq 2m \log m. \tag{4.3}$$

The central trajectory for this pair is $\{\mathcal{X}(\mu), \mathcal{Y}(\mu), \mathcal{S}(\mu)\}$, for any given $\mu > 0$, such that $\mathcal{X} = \mathcal{X}(\mu)$ is primal feasible and $(\mathcal{Y}; \mathcal{S}) = (\mathcal{Y}(\mu); \mathcal{S}(\mu))$ is dual feasible, and

$$\begin{pmatrix} \tau_i \\ x_i \end{pmatrix} + \mu f'(t_i; s_i) = 0, \quad i = 1, 2, \ldots, m, \tag{4.4}$$

or

$$\begin{pmatrix} t_i \\ s_i \end{pmatrix} + \mu f'(\tau_i; x_i) = 0, \quad i = 1, 2, \ldots, m. \tag{4.5}$$

The main iteration of a potential reduction algorithm starts with a strictly feasible primal–dual pair $\mathcal{X}$ and $(\mathcal{Y}; \mathcal{S})$; i.e.,

$$\begin{aligned} \mathcal{A}\mathcal{X} &= \mathcal{B}, \qquad \mathcal{S} = \mathcal{C} - \mathcal{A}^T \mathcal{Y}, \\ \mathcal{X} &\in \text{int}\mathcal{K}, \ \text{ and } \ \mathcal{S} \in \text{int}\mathcal{K}. \end{aligned}$$

It computes a search direction $(d_\mathcal{X}, d_\mathcal{Y}, d_\mathcal{S})$ via solving a system of linear equations. After obtaining $(d_\mathcal{X}, d_\mathcal{Y}, d_\mathcal{S})$, a new strictly feasible primal–dual pair $\mathcal{X}^+$ and $(\mathcal{Y}^+; \mathcal{S}^+)$ is generated from

$$\mathcal{X}^+ = \mathcal{X} + \alpha d_\mathcal{X}, \ \mathcal{Y}^+ = \mathcal{Y} + \beta d_\mathcal{Y}, \ \mathcal{S}^+ = \mathcal{S} + \beta d_\mathcal{S},$$

for some step-sizes α and β, and

$$\phi_\rho(\mathcal{X}^+, \mathcal{S}^+) \leq \phi_\rho(\mathcal{X}, \mathcal{S}) - \Omega(1).$$

The search direction $(d_\mathcal{X}, d_\mathcal{Y}, d_\mathcal{S})$ is determined by the following equations.

$$\mathcal{A} d_\mathcal{X} = 0, \quad d_\mathcal{S} = -\mathcal{A}^T d_\mathcal{Y} \quad \text{(feasibility)} \tag{4.6}$$

and

$$d_\mathcal{X} + F''(\mathcal{S}) d_\mathcal{S} = -\frac{\rho}{\mathcal{X}^T \mathcal{S}} \mathcal{X} - F'(\mathcal{S}) \quad \text{(dual scaling)}, \tag{4.7}$$

or

$$d_\mathcal{S} + F''(\mathcal{X}) d_\mathcal{X} = -\frac{\rho}{\mathcal{X}^T \mathcal{S}} \mathcal{S} - F'(\mathcal{X}) \quad \text{(primal scaling)}, \tag{4.8}$$

or

$$d_{\mathcal{S}} + F''(\mathcal{Z})d_{\mathcal{X}} = -\frac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{S} - F'(\mathcal{X}) \quad \text{(joint scaling)}, \tag{4.9}$$

where $\mathcal{Z}$ is chosen to satisfy

$$\mathcal{S} = F''(\mathcal{Z})\mathcal{X}. \tag{4.10}$$

(These directions were presented for linear and quadratic programming in Ye [33].) We will discuss each of these three cases in the next three subsections. The differences among the three algorithms are the computation of the search direction and their theoretical close-form step-sizes. All three generate an ϵ-optimal solution $(\mathcal{X}, \mathcal{Y}, \mathcal{S})$; i.e.,

$$\langle \mathcal{X}, \mathcal{S} \rangle \le \epsilon$$

in a guaranteed $O(\gamma\sqrt{2m}\log(\langle \mathcal{X}^0, \mathcal{S}^0\rangle/\epsilon) + \phi_{2m}(\mathcal{X}^0, \mathcal{S}^0) - 2m\log m)$ iteration. (Note from (4.3) that $\phi_{2m}(\mathcal{X}^0, \mathcal{S}^0) - 2m\log m) \ge 0$.)

In practice, one usually finds the largest step-sizes $\bar{\alpha}$ and $\bar{\beta}$ such that

$$\mathcal{X} + \bar{\alpha}d_{\mathcal{X}} \in \mathcal{K}, \quad \text{and} \quad \mathcal{S} + \bar{\beta}d_{\mathcal{S}} \in \mathcal{K} \tag{4.11}$$

then takes $\alpha \in [0, \bar{\alpha}]$ and $\beta \in [0, \bar{\beta}]$, via a line search, to minimize $\phi_\rho(\mathcal{X}^+, \mathcal{S}^+)$, or simply chooses

$$\alpha = (0.5 \sim 0.99)\bar{\alpha} \quad \text{and} \quad \beta = (0.5 \sim 0.99)\bar{\beta} \tag{4.12}$$

as long as ϕ_ρ is reduced.

4.1. Dual scaling.

The theoretical potential reduction algorithm using dual scaling can be described as follows.

```
ALGORITHM PDD.
{$\gamma$ and $\Delta$ are fixed constants such that $\gamma \ge 1$, $0 < \Delta < 1$, and $\frac{\gamma(\gamma(1-\Delta)-\Delta)}{1+\gamma} > \frac{\Delta^2}{2(1-\Delta)^2}$}.
Step_1 Compute the search direction $(d_{\mathcal{X}}, d_{\mathcal{Y}}, d_{\mathcal{S}})$ using (4.6) and (4.7).
Step_2 Compute $\lambda = \sqrt{d_{\mathcal{S}}^T F''(\mathcal{S}) d_{\mathcal{S}}}$.
    If $\lambda > \Delta$ then
        $\mathcal{X}^+ = \mathcal{X}$,                                  (primal step-size $\alpha = 0$)
        $\mathcal{S}^+ = \mathcal{S} + \frac{1}{1+\lambda}d_{\mathcal{S}}$,      (dual step-size $\beta = \frac{1}{1+\lambda}$)
    else
        $\mathcal{X}^+ = \mathcal{X} + \frac{\langle \mathcal{S}, \mathcal{X}\rangle}{\rho}d_{\mathcal{X}}$,  (primal step-size $\alpha = \frac{\langle \mathcal{S}, \mathcal{X}\rangle}{\rho}$)
        $\mathcal{S}^+ = \mathcal{S}$.                                  (dual step-size $\beta = 0$)
    endif
```

According to Nesterov and Nemirovskii [22], we have the following theorem.

THEOREM 4.1. *Starting from any strictly feasible primal solution $\mathcal{X}^0$ and strictly dual feasible solution $(\mathcal{Y}^0; \mathcal{S}^0)$, an ϵ-optimal solution to problem* (2.2) *can be obtained by repeated application of Algorithm PDD for at most $O(\gamma\sqrt{2m}\log(\langle \mathcal{X}^0, \mathcal{S}^0\rangle/\epsilon) + \phi_{2m}(\mathcal{X}^0, \mathcal{S}^0) - 2m\log m)$ iterations.* □

At first glance, it seems that the dimension of the system of linear equations defined by (4.6) and (4.7) is very large. However, the system is structured and its solution can be simplified.

Consider the dual-scaling form (4.7). Using $d_{\mathcal{S}} = -\mathcal{A}^T d_{\mathcal{Y}}$, we have

$$d_{\mathcal{X}} - F''(\mathcal{S})\mathcal{A}^T d_{\mathcal{Y}} = -\frac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{X} - F'(\mathcal{S}).$$

Multiplying $\mathcal{A}$ on both sides and noting that $\mathcal{A}d_{\mathcal{X}} = 0$, we have

$$\mathcal{A}F''(\mathcal{S})\mathcal{A}^T d_{\mathcal{Y}} = \frac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{A}\mathcal{X} + \mathcal{A}F'(\mathcal{S}),$$

or

$$\mathcal{A}F''(\mathcal{S})\mathcal{A}^T d_{\mathcal{Y}} = \frac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{B} + \mathcal{A}F'(\mathcal{S}),$$

which is a least-squares problem where $\mathcal{A}$ is scaled to $\mathcal{A}(F''(\mathcal{S}))^{1/2}$.

Therefore, the search direction $d_{\mathcal{X}}, d_{\mathcal{Y}}, d_{\mathcal{S}}$ determined by dual scaling can be computed by solving the following system of linear equations:

$$\begin{aligned} \mathcal{A}F''(\mathcal{S})\mathcal{A}^T d_{\mathcal{Y}} &= \tfrac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{B} + \mathcal{A}F'(\mathcal{S}), \\ d_{\mathcal{X}} &= F''(\mathcal{S})\mathcal{A}^T d_{\mathcal{Y}} - \tfrac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{X} - F'(\mathcal{S}), \\ d_{\mathcal{S}} &= -\mathcal{A}^T d_{\mathcal{Y}}. \end{aligned} \tag{4.13}$$

4.2. Primal scaling. The theoretical potential reduction algorithm using primal-scaling can be described as follows.

```
Algorithm PDP.
{$\gamma$ and $\Delta$ are fixed constants such that $\gamma \geq 1$, $0 < \Delta < 1$, and $\frac{\gamma(\gamma(1-\Delta)-\Delta)}{1+\gamma} > \frac{\Delta^2}{2(1-\Delta)^2}$}.
Step_1 Compute the search direction $(d_{\mathcal{X}}, d_{\mathcal{Y}}, d_{\mathcal{S}})$ using (4.6) and (4.8).
Step_2 Compute $\lambda = \sqrt{d_{\mathcal{X}}^T F''(\mathcal{X}) d_{\mathcal{X}}}$.
    If $\lambda > \Delta$ then
        $\mathcal{X}^+ = \mathcal{X} + \frac{1}{1+\lambda} d_{\mathcal{X}}$,   (primal step-size $\alpha = \frac{1}{1+\lambda}$)
        $\mathcal{S}^+ = \mathcal{S}$.                    (dual step-size   $\beta = 0$)
    else
        $\mathcal{X}^+ = \mathcal{X}$,                    (primal step-size $\alpha = 0$)
        $\mathcal{S}^+ = \mathcal{S} + \frac{\langle \mathcal{S},\mathcal{X} \rangle}{\rho} d_{\mathcal{S}}$,   (dual step-size   $\beta = \frac{\langle \mathcal{S},\mathcal{X} \rangle}{\rho}$)
    endif
```

According to Nesterov and Nemirovskii [22], we have the following theorem.

Theorem 4.2. *Starting from any strictly feasible primal solution $\mathcal{X}^0$ and strictly dual feasible solution $(\mathcal{Y}^0; \mathcal{S}^0)$, an ϵ-optimal solution to problem (2.2) can be obtained by repeated application of Algorithm PDP for at most $O(\gamma\sqrt{2m}\log(\langle \mathcal{X}^0, \mathcal{S}^0\rangle/\epsilon) + \phi_{2m}(\mathcal{X}^0, \mathcal{S}^0) - 2m\log m)$ iterations.* □

As in the dual-scaling case, we can also simplify the system of linear equations defined by (4.6) and (4.8) as follows.

Consider the primal form. Using $d_{\mathcal{S}} = -\mathcal{A}^T d_{\mathcal{Y}}$, we have

$$-\mathcal{A}^T d_{\mathcal{Y}} + F''(\mathcal{X}) d_{\mathcal{X}} = -\frac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{S} - F'(\mathcal{X})$$

or

$$\begin{aligned} d_{\mathcal{X}} - (F''(\mathcal{X}))^{-1}\mathcal{A}^T d_{\mathcal{Y}} &= -\frac{\rho}{\mathcal{X}^T\mathcal{S}}(F''(\mathcal{X}))^{-1}\mathcal{S} - (F''(\mathcal{X}))^{-1}F'(\mathcal{X}) \\ &= -\frac{\rho}{\mathcal{X}^T\mathcal{S}}(F''(\mathcal{X}))^{-1}\mathcal{S} + \mathcal{X}. \end{aligned}$$

Here we have used relation (3.3) implying

$$(F''(\mathcal{X}))^{-1}F'(\mathcal{X}) = -\mathcal{X}.$$

Also note that there is a close form for $(F''(\mathcal{X}))^{-1}$ given by (3.2). Multiplying $\mathcal{A}$ on both sides and noting $\mathcal{A}d_{\mathcal{X}} = 0$, we have

$$\mathcal{A}(F''(\mathcal{X}))^{-1}\mathcal{A}^T d_{\mathcal{Y}} = \frac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{A}(F''(\mathcal{X}))^{-1}\mathcal{S} - \mathcal{A}\mathcal{X},$$

or

$$\mathcal{A}(F''(\mathcal{X}))^{-1}\mathcal{A}^T d_{\mathcal{Y}} = \frac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{A}(F''(\mathcal{X}))^{-1}\mathcal{S} - \mathcal{B},$$

which again is a least-squares problem where $\mathcal{A}$ is scaled to $\mathcal{A}(F''(\mathcal{X}))^{-1/2}$.

Therefore, the search direction $d_{\mathcal{X}}, d_{\mathcal{Y}}, d_{\mathcal{S}}$ determined by primal scaling can be computed by solving the following system of linear equations:

$$\begin{aligned} \mathcal{A}(F''(\mathcal{X}))^{-1}\mathcal{A}^T d_{\mathcal{Y}} &= \tfrac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{A}(F''(\mathcal{X}))^{-1}\mathcal{S} - \mathcal{B}, \\ d_{\mathcal{X}} &= (F''(\mathcal{X}))^{-1}\mathcal{A}^T d_{\mathcal{Y}} - \tfrac{\rho}{\mathcal{X}^T\mathcal{S}}(F''(\mathcal{X}))^{-1}\mathcal{S} + \mathcal{X}, \\ d_{\mathcal{S}} &= -\mathcal{A}^T d_{\mathcal{Y}}. \end{aligned} \tag{4.14}$$

4.3. Joint scaling. The theoretical potential-reduction algorithm using primal–dual joint scaling generates the search direction from

$$d_{\mathcal{S}} + F''(\mathcal{Z})d_{\mathcal{X}} = -\frac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{S} - F'(\mathcal{X}),$$

where $\mathcal{Z}$ is chosen to satisfy

$$\mathcal{S} = F''(\mathcal{Z})\mathcal{X}.$$

According to Nesterov and Todd [23], there is a unique $\mathcal{Z} := ((\kappa_1; z_1); \dots; (\kappa_m; z_m))$ such that

$$(t_i; s_i) = f''(\kappa_i; z_i)(\tau_i; x_i), \quad i = 1, \dots, m.$$

In fact, for any $(\tau; x) \in \text{int}K$ and $(t; s) \in \text{int}K$ we have a unique $(\kappa; z) \in \text{int}K$ with

$$(t; s) = f''(\kappa; z)(\tau; x),$$

where

$$\kappa = \zeta\tau + \eta t \quad \text{and} \quad z = \zeta x - \eta s,$$

where

$$\zeta = \frac{1}{\sqrt{\delta(\tau; x)\delta(t; s) + \tau t + x^T s}} \quad \text{and} \quad \eta = \zeta\frac{\delta(\tau; x)}{\delta(t; s)}.$$

One can verify that

$$\delta^2(\kappa; z) = \frac{2\delta(\tau; x)}{\delta(t; s)},$$

so that from (3.1) and (3.2)

$$f''(\kappa; z) = \frac{\delta(t;s)}{\delta(\tau;x)} \begin{pmatrix} -1 & 0 \\ 0 & I_d \end{pmatrix} + \frac{\delta^2(t;s)}{\delta^2(\tau;x)} \begin{pmatrix} \kappa^2 & -\kappa z^T \\ -\kappa z & zz^T \end{pmatrix},$$

and

$$(f''(\kappa; z))^{-1} = \frac{\delta(\tau;x)}{\delta(t;s)} \begin{pmatrix} -1 & 0 \\ 0 & I_d \end{pmatrix} + \begin{pmatrix} \kappa^2 & \kappa z^T \\ \kappa z & zz^T \end{pmatrix}.$$

The joint-scaling algorithm can be described as follows.

Algorithm PDJ.

```
Step_1 Compute the scaling point $\mathcal{Z} := ((\kappa_1; z_1); (\kappa_2; z_2); \ldots; (\kappa_m; z_m))$ from
            $\kappa = \zeta\tau + \eta t$   and   $z_i = \zeta_i x_i - \eta_i s_i, \quad i = 1, 2, \ldots, m$
       where
            $\zeta_i = \frac{1}{\sqrt{\delta(\tau_i;x_i)\delta(t_i;s_i)+\tau_i t_i + x_i^T s_i}}$   and   $\eta_i = \zeta_i \frac{\delta_i(\tau_i;x_i)}{\delta(t_i;s_i)}, \quad i = 1, 2, \ldots, m.$
Step_2 Compute the search direction $(d_{\mathcal{X}}, d_{\mathcal{Y}}, d_{\mathcal{S}})$ using (4.6), (4.9), and (4.10).
Step_3 Let $\sigma(\mathcal{Z})$ be the largest primal feasible step-size form $\mathcal{X}$ along direction $\mathcal{Z}$.
       Let $\sigma(d_{\mathcal{X}})$ be the largest primal feasible step-size form $\mathcal{X}$ along direction $d_{\mathcal{X}}$.
       Let $\sigma(d_{\mathcal{S}})$ be the largest dual feasible step-size form $\mathcal{S}$ along direction $d_{\mathcal{S}}$.
       Choose the joint step-size $\bar{\alpha}$ by
            $\bar{\alpha} = \min\{\frac{1}{\sigma(\mathcal{Z})^2+\sigma(d_{\mathcal{X}})}, \quad \frac{1}{\sigma(\mathcal{Z})^2+\sigma(d_{\mathcal{S}})}\}.$
Step_4 Update the approximate solution by
            $\mathcal{X}^+ = \mathcal{X} + \bar{\alpha} d_{\mathcal{X}}, \quad \mathcal{S}^+ = \mathcal{S} + \bar{\alpha} d_{\mathcal{S}}, \quad \mathcal{Y}^+ = \mathcal{Y} + \bar{\alpha} d_{\mathcal{Y}}.$
```

According to Nesterov and Todd [23], we have the following theorem.

Theorem 4.3. *Starting from any strictly feasible primal solution $\mathcal{X}^0$ and strictly dual feasible solution $(\mathcal{Y}^0; \mathcal{S}^0)$, an ϵ-optimal solution to problem (2.2) can be obtained by repeated application of Algorithm PDJ for at most $O(\gamma\sqrt{2m}\log(\langle \mathcal{X}^0, \mathcal{S}^0 \rangle/\epsilon) + \phi_{2m}(\mathcal{X}^0, \mathcal{S}^0) - 2m\log m)$ iterations.* □

As in the cases of dual scaling and primal scaling, we can simplify the system of linear equations defined by (4.6) and (4.9) as follows.

Using $d_{\mathcal{S}} = -\mathcal{A}^T d_{\mathcal{Y}}$, we have

$$-\mathcal{A}^T d_{\mathcal{Y}} + F''(\mathcal{Z}) d_{\mathcal{X}} = -\frac{\rho}{\mathcal{X}^T \mathcal{S}} \mathcal{S} - F'(\mathcal{X})$$

or

$$\begin{aligned} d_{\mathcal{X}} - (F''(\mathcal{Z}))^{-1} \mathcal{A}^T d_{\mathcal{Y}} &= -\frac{\rho}{\mathcal{X}^T \mathcal{S}} (F''(\mathcal{Z}))^{-1} \mathcal{S} - (F''(\mathcal{Z}))^{-1} F'(\mathcal{X}) \\ &= -\frac{\rho}{\mathcal{X}^T \mathcal{S}} \mathcal{X} - F'(\mathcal{S}). \end{aligned}$$

Here we have used relations

$$(F''(\mathcal{Z}))^{-1} \mathcal{S} = \mathcal{X}$$

and

$$(F''(\mathcal{Z}))^{-1} F'(\mathcal{X}) = F'(\mathcal{S}).$$

Multiplying $\mathcal{A}$ on both sides and noting $\mathcal{A} d_{\mathcal{X}} = 0$ and $\mathcal{A}\mathcal{X} = \mathcal{B}$, we have

$$\mathcal{A}(F''(\mathcal{Z}))^{-1} \mathcal{A}^T d_{\mathcal{Y}} = \frac{\rho}{\mathcal{X}^T \mathcal{S}} \mathcal{B} + \mathcal{A} F'(\mathcal{S}),$$

which again is a least-squares problem where $\mathcal{A}$ is scaled to $\mathcal{A}(F''(\mathcal{Z}))^{-1/2}$.

Therefore, the search direction $d_{\mathcal{X}}, d_{\mathcal{Y}}, d_{\mathcal{S}}$ determined by joint scaling can be computed by solving the following system of linear equations:

$$\begin{aligned} \mathcal{A}(F''(\mathcal{Z}))^{-1}\mathcal{A}^T d_{\mathcal{Y}} &= \tfrac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{B} + \mathcal{A}F'(\mathcal{S}), \\ d_{\mathcal{X}} &= (F''(\mathcal{Z}))^{-1}\mathcal{A}^T d_{\mathcal{Y}} - \tfrac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{X} - F'(\mathcal{S}), \\ d_{\mathcal{S}} &= -\mathcal{A}^T d_{\mathcal{Y}}. \end{aligned} \tag{4.15}$$

5. Complexity and implementation. As we have seen, the number of iterations required (as stated in Theorems 4.1–4.3) to compute an ϵ-optimal solution to problem (2.2) depends on the initial point $(\mathcal{X}^0, \mathcal{S}^0, \mathcal{Y}^0)$. In this section, we discuss initial point selection and other computational issues for solving problem (2.2) using the algorithms presented in section 3.

5.1. Initial point. The algorithms discussed in the previous section all require a pair of strictly primal–dual interior feasible solutions. In the following, we give one such pair.

Let

$$\bar{c} = \max_{1 \le i \le m} \|c_i\|,$$

and

$$u^0 = 0, \quad s_i^0 = c_i, \quad t_i^0 = \sqrt{\|c_i\|^2 + m\bar{c}^2}, \quad i = 1, 2, \ldots, m,$$

and

$$\tau_i^0 = 1, \quad x_i^0 = 0, \quad i = 1, 2, \ldots, m.$$

Then, one can verify that $\mathcal{X}$ is an interior feasible solution to (P) and $\mathcal{S}$ and $\mathcal{Y}$ form an interior feasible solution to (D). One can also verify that

$$\langle \mathcal{X}^0, \mathcal{S}^0 \rangle = (\mathcal{X}^0)^T \mathcal{S}^0 = \sum_{i=1}^m t_i^0 \tau_i^0 = \sum_{i=1}^m \sqrt{\|c_i\|^2 + m\bar{c}^2} \le \bar{c}m\sqrt{1+m}$$

and the initial value

$$\begin{aligned} \phi_{2m}(\mathcal{X}^0, \mathcal{S}^0) - 2m\log m &= 2m\log(\langle \mathcal{X}^0, \mathcal{S}^0 \rangle) + F(\mathcal{X}^0) + F(\mathcal{S}^0) - 2m\log m \\ &= 2m\log(\langle \mathcal{X}^0, \mathcal{S}^0 \rangle) + F(\mathcal{S}^0) - 2m\log m \\ &= 2m\log(\langle \mathcal{X}^0, \mathcal{S}^0 \rangle) - m\log(m\bar{c}^2) - 2m\log m \\ &\le 2m\log(m\sqrt{1+m}\bar{c}) - m\log(m\bar{c}^2) - 2m\log m \\ &= m\log(1+m) - m\log m \\ &= m\log(1+1/m) \\ &\le 1. \end{aligned}$$

With this initial point, we have the following corollary.

COROLLARY 5.1. *Let the initial feasible primal solution $\mathcal{X}^0$ and dual feasible solution $(\mathcal{Y}^0; \mathcal{S}^0)$ be given as above. Then, an ϵ-optimal solution to problem* (2.2) *can* *be obtained by the potential reduction algorithms in at most* $O(\gamma\sqrt{m}(\log(\bar{c}/\epsilon)+\log m))$ *iterations, where*

$$\bar{c} = \max_{1\le i\le m} \|c_i\|. \qquad \square$$

Note that if positive semidefinite program (3.6) is solved, the iteration complexity bound will be $O(\gamma\sqrt{md}(\log(\bar{c}/\epsilon)+\log md))$, which is $\sqrt{d}$ higher than the bound given by the above corollary.

5.2. Search direction. At each step of the potential-reduction algorithm, we need to compute the search direction $d_{\mathcal{X}}$, $d_{\mathcal{S}}$, and $d_{\mathcal{Y}}$ by solving a system of linear equations. In what follows, we will show that this can be further simplified, taking advantage of the special structure of the problem.

Consider the search direction defined by dual scaling (4.7). For $i = 1,\ldots,m$, it can be decomposed as

$$\begin{pmatrix} d_{\tau_i} \\ d_{x_i} \end{pmatrix} + \left(\frac{2}{\delta^2(t_i;s_i)}\begin{pmatrix} -1 & 0 \\ 0 & I_d \end{pmatrix} + \frac{4}{\delta^4(t_i;s_i)}\begin{pmatrix} (t_i)^2 & -t_i(s_i)^T \\ -t_i s_i & s_i(s_i)^T \end{pmatrix}\right)\begin{pmatrix} d_{t_i} \\ d_{s_i} \end{pmatrix}$$

$$= -\frac{\rho}{\mathcal{X}^T\mathcal{S}}\begin{pmatrix} \tau_i \\ x_i \end{pmatrix} - \frac{2}{\delta^2(t_i;s_i)}\begin{pmatrix} -t_i \\ s_i \end{pmatrix}. \tag{5.1}$$

Note that $s_i = c_i - A_i^T u$, $d_{s_i} = -A_i^T d_u$, $\tau_i = 1$, and $d_{\tau_i} = 0$ for $i = 1,\ldots,m$. The system can be written as

$$\begin{aligned}\left(-\frac{2}{\delta^2(t_i;s_i)} + \frac{4(t_i)^2}{\delta^4(t_i;s_i)}\right)d_{t_i} + \frac{4t_i}{\delta^4(t_i;s_i)}(s_i)^T A_i^T d_u &= -\frac{\rho}{\mathcal{X}^T\mathcal{S}} + \frac{2}{\delta^2(t_i;s_i)}t_i,\\ d_{x_i} - \frac{2}{\delta^2(t_i;s_i)}A_i^T d_u + \frac{4}{\delta^4(t_i;s_i)}(-t_i d_{t_i} s_i - s_i(s_i)^T A_i^T d_u) &= -\frac{\rho}{\mathcal{X}^T\mathcal{S}}x_i - \frac{2}{\delta^2(t_i;s_i)}s_i.\end{aligned}$$

From the first equation we have

$$d_{t_i} = \frac{\delta^2(t_i;s_i)t_i - \frac{\rho\delta^4(t_i;s_i)}{2\mathcal{X}^T\mathcal{S}} - 2t_i(s_i)^T A_i^T d_u}{2(t_i)^2 - \delta^2(t_i;s_i)}.$$

Substituting this relation into the second equation, we have

$$d_{x_i} + \frac{2}{\delta^2(t_i;s_i)}\left(\frac{2}{2(t_i)^2 - \delta^2(t_i;s_i)}s_i(s_i)^T - I_d\right)A_i^T d_u$$

$$= -\frac{\rho}{\mathcal{X}^T\mathcal{S}}x_i + \left(\frac{2(1 - \frac{\rho}{\mathcal{X}^T\mathcal{S}}t_i)}{2(t_i)^2 - \delta^2(t_i;s_i)}\right)s_i.$$

Moreover, since

$$\sum_{i=1}^m A_i x_i = 0, \qquad \sum_{i=1}^m A_i d_{x_i} = 0,$$

we have

$$\left(\sum_{i=1}^{m} \frac{2}{\delta^2(t_i;s_i)} \left(\frac{2}{2(t_i)^2-\delta^2(t_i;s_i)} A_i s_i (s_i)^T A_i^T - A_i A_i^T\right)\right) d_u$$
$$= \sum_{i=1}^{m} \left(\frac{2(1-\frac{\rho}{\mathcal{X}^T\mathcal{S}} t_i)}{2(t_i)^2-\delta^2(t_i;s_i)}\right) A_i s_i. \tag{5.2}$$

Note that the system for computing d_u may not have full rank. If that is the case, any feasible solution is acceptable.

It requires $O(mn^2 d)$ operations to set up the system (5.2) for computing d_u. Solving the system requires $O(n^3)$ operations. Once d_u is computed, $O(mnd)$ operations are required to compute d_x and d_s. Therefore, the number of arithmetic operations in each iteration is bounded by $O(n^3 + mn^2 d)$. The following theorem follows from Corollary 5.1 and the above analysis.

Theorem 5.2. *Let the initial feasible primal solution $\mathcal{X}^0$ and dual feasible solution $(\mathcal{Y}^0;\mathcal{S}^0)$ be given as above. Then, an ϵ-optimal solution to problem* (2.1) *can be obtained by the potential reduction algorithms in at most $O(\gamma\sqrt{m}(\log(\bar{c}/\epsilon)+\log m))$ iterations, where*

$$\bar{c} = \max_{1\le i\le m} \|c_i\|,$$

and each iteration requires $O(n^3 + mn^2 d)$ arithmetic operations. □

Note that if γ is chosen as a constant and the problem is normalized such that $\bar{c} = 1$, i.e., all of c_i is within the unit ball in R^d, then the iteration bound is $O(\sqrt{m}(\log(1/\epsilon)+\log m))$. We will further discuss this issue in following applications.

6. Applications. In this section, we will apply the algorithms presented in the previous sections to solve the ESFL problem, the EMFL problem, and the SMT problem under a given topology. We will also take advantage of the special structures of these special problems and obtain improved computational complexity results wherever possible.

6.1. The ESFL problem. Let $a_1, a_2, \ldots, a_M$ be M points in R^d, the d-dimensional Euclidean space. Let $w_1, w_2, \ldots, w_M$ be M positive weights. Find a point $x \in R^d$ that will minimize

$$f(x) = \sum_{i=1}^{M} w_i ||x - a_i||. \tag{6.1}$$

This is called the ESFL problem.

In the ESFL problem, $a_1, a_2, \ldots, a_M$ represent the respective locations of M clients in a given region and x represents the location of a prospective service center. $w_1, w_2, \ldots, w_M$ represent the respective amount of service requests of the clients to the service center. The ESFL problem is concerned with finding the location for the service center to minimize the sum of weighted Euclidean distances from the service center to each of the clients. For more information on this problem, see [17, 19].

The ESFL problem can be easily transformed into a special case of problem (2.1) where $m = M$, $n = d$ and $c_i = w_i a_i$, $A_i^T = w_i I_d$, $i = 1, 2, \ldots, M$. It follows from Theorem 5.1 that Theorem 6.1 holds.

Theorem 6.1. *An ϵ-optimal solution to the ESFL problem* (6.1) *can be computed using any of our potential reduction algorithms in $O(\sqrt{M}(\log(\bar{c}/\epsilon)+\log M))$ iterations* *where* $\bar{c} = \max_{1\le i\le m} \|w_i a_i\|$, *and each iteration requires* $O(d^3 + d^2M)$ *arithmetic operations.* □

6.2. The EMFL problem. Let $a_1, a_2, \ldots, a_M$ be M points in R^d, the d-dimensional Euclidean space. Let w_{ji}, $j = 1, 2, \ldots, N$, $i = 1, 2, \ldots, M$, and $v_{jk}, 1 \le j < k \le N$ be given nonnegative numbers. Find a point $x = (x_1; x_2; \ldots; x_N) \in R^{dN}$ that will minimize

$$f(x) = \sum_{j=1}^{N}\sum_{i=1}^{M} w_{ji}||x_j - a_i|| + \sum_{1\le j<k\le N} v_{jk}||x_j - x_k||. \tag{6.2}$$

This is the so-called EMFL problem. For ease of notation, we assume that $v_{jj} = 0$ for $j = 1, 2, \ldots, N$ and that $v_{jk} = v_{kj}$ for $1 \le k < j \le N$.

In the EMFL problem, $a_1, a_2, \ldots, a_M$ represent the locations of M existing facilities; $x_1, x_2, \ldots, x_N$ represent the locations of N new facilities; the objective function $f(x)$ is the sum of weighted Euclidean distances from each new facility to each existing facility and those between each pair of new facilities; and our goal is to find optimal locations for the new facilities, i.e., to minimize $f(x)$.

In problem (6.2), some of the weights w_{ji} and v_{jk} may be zero. Let m be the number of nonzero weights in (6.2). Then the EMFL problem (6.2) is the minimization of m Euclidean norms. Without loss of generality, we assume that for each $j \in \{1, 2, \ldots, N\}$ there exists a nonzero w_{ji} for some $i \in \{1, 2, \ldots, M\}$ or a nonzero v_{jk} for some $k \in \{1, 2, \ldots, N\}$.

To transform the EMFL problem (6.2) into an instance of problem (2.1), we simply do the following. Let $u = (x_1; x_2; \ldots; x_N)$. It is clear that $u \in R^n$ where $n = dN$. For each nonzero w_{ji}, there is a corresponding term of Euclidean norm $||c(w_{ji}) - A(w_{ji})^T u||$ where $c(w_{ji}) = w_{ji}a_i$, and $A(w_{ji})^T$ is a row of N blocks of d-by-d matrices whose jth block is $w_{ji}I_d$ and whose other blocks are all zero. For each nonzero v_{jk}, there is a corresponding term of Euclidean norm $||c(v_{jk}) - A(v_{jk})^T u||$ where $c(v_{jk}) = 0$, and $A(v_{jk})^T$ is a row of N blocks of d-by-d matrices whose jth and kth blocks are $-v_{jk}I_d$ and $v_{jk}I_d$, respectively, and whose other blocks are all zero.

Now it is clear that we have transformed the EMFL problem (6.2) into an instance of (2.1) where $n = dN$, and m is the number of nonzero weights w_{ji} and v_{jk}. Note that the system (5.2) can be set up with $O(md^2)$ operations. Therefore, it follows from Theorem 5.1 that we have the following theorem.

THEOREM 6.2. *An ϵ-optimal solution to the EMFL problem* (6.2) *can be computed using any of our algorithms in* $O(\sqrt{MN}(\log(\bar{c}/\epsilon) + \log(MN)))$ *iterations where* $\bar{c} = \max_{1\le j\le n\ \ 1\le i\le m} \|w_{ji}a_i\|$, *and each iteration requires* $O(d^3N^3 + MNd^2)$ *arithmetic operations.* □

6.3. The SMT problem. The *Euclidean SMT problem* is given by a set of points $P = \{p_1, p_2, \ldots, p_N\}$ in the Euclidean plane and asks for the shortest planar straight-line graph spanning P. The solution takes the form of a tree, called the *SMT*, that includes all the given points, called *regular points*, along with some extra vertices, called *Steiner points*. It is known that there are at most $N - 2$ Steiner points and the degree of each Steiner point is at most 3. See [12, 20] for details.

DEFINITION 6.3 (see [12, 14, 15]). *A* full Steiner topology *of point set P is a tree graph whose vertex set contains P and $N - 2$ Steiner points and that the degree of each vertex in P is exactly* 1 *and that the degree of each Steiner vertex is exactly* 3.

Computing an SMT for a given set of N points in the Euclidean plane is NP-hard. However, the problem of computing the shortest network under a given full Steiner topology can be solved efficiently. Recently, there have been increased interests in this latter problem, and several algorithms have been proposed [14, 15, 28]. We will formulate this problem as a special case of problem (2.1).

Let $m = 2N - 3$, $d = 2$, and $n = 2N - 4$. Let $u \in R^{2N-4}$ represent the locations of the $N - 2$ Steiner points. Without loss of generality, we may order the edges in the given full Steiner topology in such a way that each of the first N edges connects a regular point to a Steiner point. For $i = 1, 2, \ldots, N$, c_i is p_{i_1} where i_1 is the index of the regular point on the ith edge; $A_i^T \in R^{2\times n}$ is a row of $N - 2$ 2-by-2 block matrices where only the i_2th block is I_2 and the rest are all zero, where i_2 is the index of the Steiner point on the ith edge. For $i = N + 1, N + 2, \ldots, m$, $c_i = 0$ and $A_i^T \in R^{2\times n}$ is a row of $N - 2$ 2-by-2 block matrices where the i_1st block is $-I_2$, the i_2nd block is I_2, and the rest of the blocks are all zero, where i_1 and i_2 are the indices of the two Steiner points on the ith edge. It is clear that we have transformed the problem of computing a shortest network under a full Steiner topology into an instance of (2.1), where $d = 2$, $n = 2N - 4$, and $m = 2N - 3$. Therefore, it can be solved efficiently using our interior-point algorithm.

Note that we can move the point set P so that its gravitational center is the origin. Therefore, the Euclidean norms of the regular points are bounded by the largest pairwise distance among the points in P which corresponds to the constant $\bar{c}$ in previous theorems. Furthermore, we will show in the following that the search direction can be computed in $O(N)$ arithmetic operations using a technique known as *Gaussian elimination on leaves of a tree* [28].

Since $A_i^T \in R^{2\times N}$ contains at most two nonzero 2-by-2 blocks, the system (5.2) can be set up in $O(N)$ operations. The left-hand-side matrix of (5.2) (call it H) consists of $(N - 2)$-by-$(N - 2)$ blocks of 2-by-2 matrices. The (i, j) block of H is nonzero only if there is an edge in the topology which connects the ith and the jth Steiner points. Now consider the tree spanning the $N - 2$ Steiner points. We may *delete* a leaf vertex ea in the tree as follows: let eb be the (unique) vertex in the tree that is connected to ea by an edge in the tree. We *delete* the vertex ea and the edge $(ea,\ eb)$ from the tree by choosing $H(2 * ea - 1,\ 2 * ea - 1)$ as the pivot element and eliminate the entries $H(2 * ea - 0,\ 2 * ea - 1)$, $H(2 * eb - 1,\ 2 * ea - 1)$, and $H(2 * eb - 0,\ 2 * ea - 1)$. Then use $H(2 * ea - 0,\ 2 * ea - 0)$ as the pivot element and eliminate the entries $H(2 * eb - 1,\ 2 * ea - 0)$ and $H(2 * eb - 0,\ 2 * ea - 0)$. All of this can be done in $O(1)$ operations and will not make a zero block nonzero. In other words, *deleting* a leaf vertex in the tree requires $O(1)$ operations. Therefore, Gaussian elimination on leaves of a tree requires $O(N)$ operations. In the reverse order, back substitution can be done in $O(N)$ operations, too. Therefore, we have Theorem 6.4.

THEOREM 6.4. *An ϵ-optimal solution to the shortest network under a given full Steiner topology of N regular points in the Euclidean plane can be computed using our potential-reduction algorithms in $O(\sqrt{N}(\log(\bar{c}/\epsilon) + \log N))$ iterations where $\bar{c}$ is the largest pairwise distance among the regular points and each iteration requires $O(N)$ arithmetic operations. Therefore, the computation of an ϵ-optimal solution requires $O(N\sqrt{N}(\log(\bar{c}/\epsilon) + \log N))$ arithmetic operations.* □

The problem of computing the shortest network under a full Steiner topology was first studied by Hwang [14], Hwang and Weng [15], and Smith [28]. Hwang [14] presented a linear time exact algorithm that can output the shortest network under a given full Steiner topology if there exists a nondegenerate SMT corresponding to that given topology and quits otherwise. Hwang and Weng [15] presented an $O(N^2)$ time graphical algorithm that can output the shortest network under a given full Steiner topology if the shortest network under the given topology is a tree with maximum vertex degree 3 and quits otherwise. Our algorithm can always output an ϵ-optimal network under the given topology in $O(N\sqrt{N}(\log(\bar{c}/\epsilon)+\log N))$ operations where $\bar{c}$ is the largest pairwise distance among the given points. This resolves an open problem of [15].

7. Computational examples. We have implemented all three versions of our algorithm using MATLAB. From our *preliminary* implementation, it seems that the one using dual scaling is numerically simpler and stabler. Therefore, we implemented that algorithm for computing the shortest network under a tree topology in Fortran 77, using Gaussian elimination on leaves of the topology tree. In the following, we present some preliminary computational results on the shortest network problem. Extensive computational study of the algorithms will be given in a separate paper.

TABLE 7.1
The coordinates of the 10 *regular points in example* 1.

index	x-coordinate	y-coordinate	index	x-coordinate	y-coordinate
9	2.30946900	9.20821100	14	7.59815200	0.61583600
10	0.57736700	6.48093800	15	8.56812900	3.07917900
11	0.80831400	3.51906200	16	4.75750600	3.75366600
12	1.68591200	1.23167200	17	3.92609700	7.00879800
13	4.11085500	0.82111400	18	7.43649000	7.68328400

The program was run on a Silicon Graphics Indy workstation. In our implementation, we used $\gamma = 2m$ to take *long steps* instead of using the conservative theoretical parameter $\gamma = 1$. Also, we used 0.9 times the largest feasible step-size as the actual step-size rather than using the theoretical step-size or a line search. For our implementation, we index the Steiner points first, followed by the regular points.

TABLE 7.2
The tree topology for example 1.

edge-index	ea-index	eb-index	edge-index	ea-index	eb-index
1	9	7	10	18	8
2	10	1	11	5	6
3	11	2	12	6	4
4	12	3	13	4	3
5	13	4	14	3	2
6	14	5	15	2	1
7	15	5	16	1	7
8	16	6	17	7	8
9	17	8			

Our first example contains 10 regular points. The coordinates of the 10 regular points are given in Table 7.1. The tree topology is given in Table 7.2 where for each edge, indices of its two vertices are shown next to the index of the edge. This topology is the best topology obtained by a branch-and-bound algorithm. Therefore, the shortest network under this topology is actually the SMT for the given 10 regular points.

Our algorithm solves this problem to 10^{-8} in 0.045 seconds and a total of 23 iterations. Table 7.3 shows the computer output of this test run. The second column in Table 7.3 shows the cost of the current network (i.e., the sum of Euclidean norms in the current network). The third column shows the duality gap, which is an upper bound of the error in the cost of the current network to the cost of the optimal

Table 7.3
Output of our algorithm for example 1.

iteration	network-cost	duality-gap	pstep-max	dstep-max
1	67.4046273974	755.3696677104	28.8384401017	0.1356841643
2	55.4697474888	195.1824411479	3.4806842442	0.1032272123
3	27.4932167097	101.3496514586	0.4832738411	0.1638326757
4	27.0322340903	26.0099590874	0.1113384478	0.1119122550
5	26.1012759902	9.1871566968	0.0403664605	0.1191177601
6	25.6601657571	2.5422959471	0.0099090385	0.1017762723
7	25.4826595690	0.8430133790	0.0022051467	0.0996860187
8	25.3997342761	0.3156519930	0.0006569311	0.1251074829
9	25.3713549379	0.1277519346	0.0001667930	0.1478311776
10	25.3613423325	0.0715146951	0.0002153768	0.3040757280
11	25.3575447731	0.0263880927	0.0000626213	0.3321649420
12	25.3565967482	0.0128961955	0.0000577043	0.5297753443
13	25.3562709801	0.0021763160	0.0000098621	0.1682928011
14	25.3561300062	0.0004901604	0.0000020309	0.1142469220
15	25.3560841523	0.0001476805	0.0000006305	0.1145905399
16	25.3560721582	0.0000397667	0.0000001962	0.1007746545
17	25.3560692545	0.0000107249	0.0000000618	0.0897091127
18	25.3560681805	0.0000026435	0.0000000156	0.0732031761
19	25.3560678856	0.0000008525	0.0000000057	0.0818257275
20	25.3560678157	0.0000002438	0.0000000015	0.0789620241
21	25.3560677874	0.0000000757	0.0000000005	0.0834184991
22	25.3560677824	0.0000000206	0.0000000001	0.0763003478
23	25.3560677802	0.0000000065	0.0000000000	0.0825900991

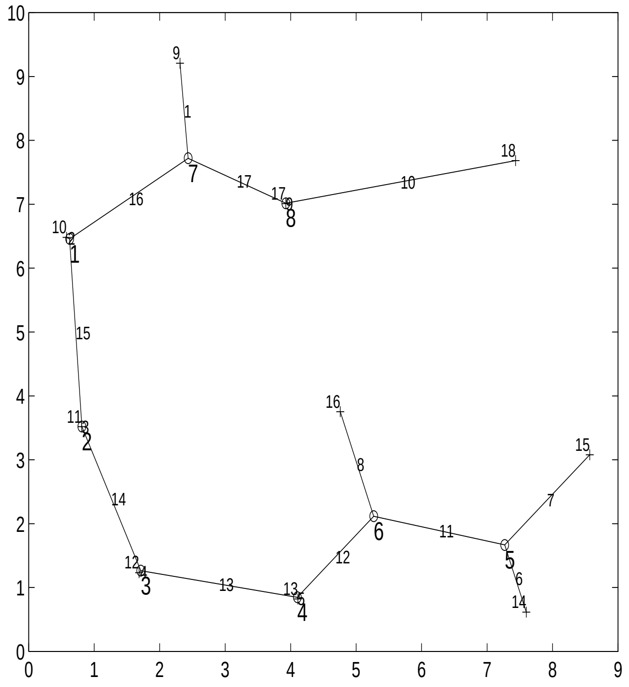


Fig. 7.1. *The shortest network for* 10 *regular points in example* 1.

TABLE 7.4
The topology and the coordinates of the four regular points in example 2.

point-index	x-coord	y-coord	point-index	x-coord	y-coord
3	-100.0	1.0	5	-100.0	-1.0
4	100.0	1.0	6	100.0	1.0

edge-index	ea-index	eb-index	edge-index	ea-index	eb-index
1	3	1	4	6	2
2	4	1	5	1	2
3	5	2			

TABLE 7.5
Output of our algorithm for example 2.

iteration	network-cost	duality-gap	pstep-max	dstep-max
1	40.1995024845	120.9404740550	0.8445579680	0.0492658697
2	41.3468150068	13.7530974274	0.0724695587	0.0339618552
3	40.2654764899	3.0068405143	0.0185627882	0.0299007875
4	40.2539043360	0.4948832387	0.0027833196	0.0183486445
5	40.2010107181	0.1731075550	0.0015791427	0.0255792520
6	40.2020816006	0.0325421036	0.0002280125	0.0173067376
7	40.2001533446	0.0115207903	0.0001157000	0.0248950063
8	40.1997465956	0.0022693804	0.0000146742	0.0171813134
9	40.1995061111	0.0008119015	0.0000084984	0.0253539092
10	40.1995148034	0.0001551869	0.0000010559	0.0173411905
11	40.1995055233	0.0000546467	0.0000005664	0.0248868315
12	40.1995036402	0.0000107280	0.0000000713	0.0171141485
13	40.1995024994	0.0000038458	0.0000000411	0.0253093321
14	40.1995025427	0.0000007393	0.0000000049	0.0173829301
15	40.1995024992	0.0000002599	0.0000000027	0.0249213172
16	40.1995024900	0.0000000508	0.0000000003	0.0170733567
17	40.1995024846	0.0000000183	0.0000000002	0.0252981953
18	40.1995024848	0.0000000035	0.0000000000	0.0174074280

(shortest) network. The last two columns show the largest primal and dual feasible step-sizes.

The final solution is shown in Figure 7.1, where regular points are labeled by "+" and Steiner points are labeled by "o." We can see from Figure 7.1 that the shortest network is degenerate [15] where five edges (each connecting a regular point to a Steiner point) shrink. This problem can be solved using the graphical method of [15] but is very difficult for algorithms like HAP [9]. For comparison, we have used HAP to solve the same problem by setting $\epsilon = 10^{-8}$ and using the locations of Steiner points generated by one step of our algorithm as the starting point for HAP. Because the problem is degenerate, HAP ran poorly compared with our algorithm. To get a solution as good as the one obtained using 10 iterations of our algorithm, HAP used 39.512 seconds and 248500 iterations. No matter how long we let it run, HAP failed to find a solution whose cost function is better than 25.3561402805 which can be obtained by our algorithm in 14 iterations.

Our second example has four regular points. The purpose of this example is to show that our algorithm can compute the shortest network under a tree topology where two Steiner points coincide. The algorithm in [15] will quit on this problem before it finds the shortest network. The coordinates of the four regular points and the tree topology in this example are given in Table 7.4. This topology is not the best topology. Therefore, the shortest network under this topology is not the SMT for the given four regular points.

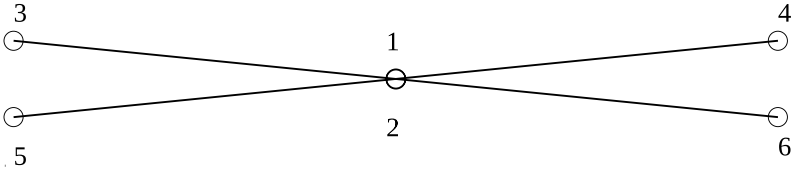


FIG. 7.2. *The shortest network for four regular points in example* 2.

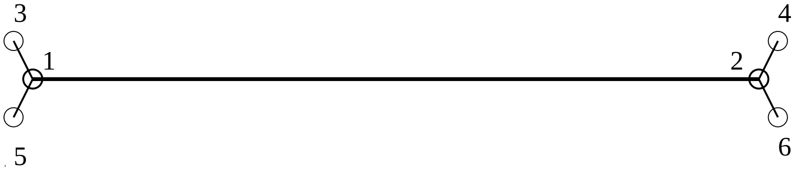


FIG. 7.3. *The SMT for four regular points in example* 2.

Our algorithm solves the second problem to 10^{-8} in 0.022 seconds and a total of 18 iterations. Table 7.5 shows the computer output of this test run. The shortest network under this topology has a cost of 40.1995 and is illustrated in Figure 7.2.

Figure 7.3 shows the SMT, which is the shortest network under a different topology. The corresponding cost is 23.4641. We would like to point out that the algorithms of [14] and [15] can both find the shortest network under this topology.

8. Conclusions. In this paper, we have transformed the problem of minimizing a sum of Euclidean norms into a standard convex programming problem in its dual conic form where the cone and its associated barrier are self-scaled [23]. We then presented an efficient primal–dual potential reduction algorithm for solving this problem. In applications, we have shown that computing an ϵ-optimal solution of the shortest network under a tree topology interconnecting N regular points requires only $O(N\sqrt{N}(\log(\bar{c}/\epsilon)+\log N))$ arithmetic operations, where $\bar{c}$ is the largest pairwise distance among the given point set.

When applied to compute the shortest network under a tree topology interconnecting N regular points, our algorithm does not suffer from degeneracies and it compares favorably with the $O(N^2)$ algorithm of [15] in both theoretical complexity and ease of implementation. When applied to EMFL problems, our algorithm compares favorably with the algorithm of [32] because our algorithm has a better complexity result and stores the locations of the new facilities in the dual variable u while the latter does not provide such information directly. Our implementation is only preliminary. Computational issues of our algorithm are under investigation and will be reported in another paper.

REFERENCES

[1] K. D. ANDERSEN, *An efficient Newton barrier method for minimizing a sum of Euclidean norms*, SIAM J. Optim., 6 (1996), pp. 74–95.

[2] K. D. ANDERSEN AND E. CHRISTIANSEN, *A Symmetric Primal–Dual Newton Method for Minimizing a Sum of Norms*, manuscript, Odense University, Denmark, 1995.

[3] P. H. CALAMAI AND A. R. CONN, *A second-order method for solving the continuous multifacility location problem*, in Numerical Analysis: Proceedings of the Ninth Biennial Conference, G. A. Watson, ed., Dundee, Scotland, Lecture Notes in Mathematics 912, Springer-Verlag, Berlin, 1982, pp. 1–25.

[4] P. H. Calamai and A. R. Conn, *A projected Newton method for l_p norm location problems*, Math. Programming, 38 (1987), pp. 75–109.

[5] R. Chandrasekaran and A. Tamir, *Open questions concerning Weiszfeld's algorithm for the Fermat–Weber location problem*, Math. Programming, 44 (1989), pp. 293–295.

[6] R. Chandrasekaran and A. Tamir, *Algebraic optimization: The Fermat–Weber location problem*, Math. Programming, 46 (1990), pp. 219–224.

[7] A. R. Conn and M. L. Overton, *A Primal–Dual Interior Point Method for Minimizing a Sum of Euclidean Distances*, 1995, manuscript.

[8] D. den Hertog, *Interior Point Approach to Linear, Quadratic and Convex Programming*, Kluwer Academic Publishers, Norwell, MA, 1994.

[9] J. W. Eyster, J. A. White, and W. W. Wierwille, *On solving multifacility location problems using a hyperboloid approximation procedure*, AIIE Transactions, 5 (1973), pp. 1–6.

[10] R. L. Francis and A. V. Cabot, *Properties of a multifacility location problem involving Euclidean distances*, Naval Res. Logist., 19 (1972), pp. 335–353.

[11] R. L. Francis, Leon F. McGinnis, Jr., and John A. White, *Facility Layout and Location: An Analytical Approach*, Prentice–Hall, Englewood Cliffs, NJ, 1991.

[12] E. N. Gilbert and H. O. Pollak, *Steiner minimal trees*, SIAM J. Appl. Math., 16 (1968), pp. 1–29.

[13] O. Güler, *Barrier Functions in Interior Point Methods*, manuscript, 1994.

[14] F. K. Hwang, *A linear time algorithm for full Steiner trees*, Oper. Res. Lett., 4 (1986), pp. 235–237.

[15] F. K. Hwang and J. F. Weng, *The shortest network under a given topology*, J. Algorithms, 13 (1992), pp. 468–488.

[16] M. Kojima, Tokyo Institute of Technology, Japan, 1995, private communication.

[17] H. W. Kuhn, *On a pair of dual nonlinear programs*, in Nonlinear Programming, J. Abadie, ed., North–Holland, Amsterdam, 1967, pp. 39–54.

[18] H. W. Kuhn, *A note on Fermat's problem*, Math. Programming, 4 (1973), pp. 98–107.

[19] R. F. Love, J. G. Morris, and G. O. Wesolowsky, *Facilities Location: Models & Methods*, North–Holland, Amsterdam, 1988.

[20] Z. A. Melzak, *On the problem of Steiner*, Canad. Math. Bull., 16 (1961), pp. 143–148.

[21] W. Miehle, *Link length minimization in networks*, Oper. Res., 6 (1958), pp. 232–243.

[22] Yu. E. Nesterov and A. Nemirovskii, *Interior Polynomial Algorithms in Convex Programming*, SIAM, Philadelphia, 1994.

[23] Yu. E. Nesterov and M. J. Todd, *Self-Scaled Barriers and Interior-Point Methods for Convex Programming*, manuscript.

[24] L. M. Ostresh, *The multifacility location problem: Applications and descent theorems*, J. Regional Science, 17 (1977), pp. 409–419.

[25] M. L. Overton, *A quadratically convergent method for minimizing a sum of Euclidean norms*, Math. Programming, 27 (1983), pp. 34–63.

[26] J. B. Rosen and G. L. Xue, *On the convergence of Miehle's algorithm for the Euclidean multifacility location problem*, Oper. Res., 40 (1992), pp. 188–191.

[27] J. B. Rosen and G. L. Xue, *On the convergence of a hyperboloid approximation procedure for solving the perturbed Euclidean multifacility location problem*, Oper. Res., 41 (1993), pp. 1164–1171.

[28] W. D. Smith, *How to find Steiner minimal trees in Euclidean d-space*, Algorithmica, 7 (1992), pp. 137–177.

[29] C. Y. Wang et al., *On the convergence and rate of convergence of an iterative algorithm for the plant location problem*, Qufu Shiyun Xuebao, 2 (1975), pp. 14–25 (in Chinese).

[30] E. Weiszfeld, *Sur le point par lequel le somme des distances de n points donnes est minimum*, Tôhoku Math. J., 43 (1937), pp. 355–386.

[31] G. L. Xue, *Algorithms for Computing Extreme Points of Convex Hulls and the Euclidean Facilities Location Problem*, Ph.D. thesis, Computer Science Department, University of Minnesota, Minneapolis, MN, 1991.

[32] G. L. Xue, J. B. Rosen, and P. M. Pardalos, *A polynomial time dual algorithm for the Euclidean multifacility location problem*, in Proceedings of Second Conference on Integer Programming and Combinatorial Optimization, Pittsburgh, PA, 1992, pp. 227–236.

[33] Y. Ye, *Interior-point algorithms for quadratic programming*, in Recent Developments in Mathematical Programming, S. Kumar, ed., Gordon and Breach, New York, 1991, pp. 237–262.

SIAM J. OPTIM.
Vol. 7, No. 4, pp. 1037–1053, November 1997

A DYNAMIC ADAPTIVE RELAXATION SCHEME APPLIED TO THE EUCLIDEAN STEINER MINIMAL TREE PROBLEM*

FRANÇOIS CHAPEAU-BLONDEAU[†], FABRICE JANEZ[‡], AND JEAN-LOUIS FERRIER[‡]

Abstract. The Steiner problem is an NP-hard optimization problem which consists of finding the minimal-length tree connecting a set of N points in the Euclidean plane. Exact methods of resolution currently available are exponential in N, making exact minimal trees accessible for only small size problems (up to $N \approx 100$). An acceptable suboptimal solution is provided by the minimum spanning tree (MST) which has been shown computable in an $O(N \log N)$ step. We propose here an $O(N)$ process that is able to relax a given initial Steiner tree into a local minimum of its length. This process, based on a physical analogy, simulates the dynamics of a fluid film which relaxes under surface tension forces and stabilizes in an equilibrium configuration minimizing its total length, through purely local interactions. To improve the solution to the Steiner problem, this $O(N)$ relaxation scheme is applied to reduce the length of the MST. This results in a heuristic of a very low $O(N \log N)$ complexity for the Steiner problem, whose performance is shown to compare quite favorably with that of the best available heuristics. Large problem sizes up to $N = 10000$ were successfully tackled. A characterization of the asymptotic behavior of the solution of the Steiner problem shows a stabilization to a nonvanishing positive value of the average length reduction achieved over the MST and predicts an average length for the minimal Steiner tree of about 3% below $0.65N^{1/2}$ for large N.

Key words. Steiner problem, minimal tree, minimum spanning tree, optimization, relaxation

AMS subject classifications. 90C35, 05C35, 49-04

PII. S1052623494275069

1. Introduction.

The Steiner problem is an optimization problem which consists of finding the shortest possible tree connecting a given finite set of N points in the Euclidean plane [1], [2]. A concrete embodiment of this problem is to devise the shortest road network connecting a given set of cities. For this reason, we shall call here cities the points that have to be connected in the Steiner problem. The expression of the solution requires one, in general, to introduce additional points, the Steiner points. The solution of the problem is the minimal Steiner tree, and it is given as a set of linear edges connecting the cities through the medium of the Steiner points. Although very simple to state, the Steiner problem has been proven NP-hard when defined on the usual continuous Euclidean metric. It becomes NP-complete if the Euclidean metric is discretized. The Euclidean Steiner problem is thus at least as difficult as any NP-complete problem [3]. Available algorithms yielding the exact minimal Steiner tree are exponential in N and are now limited to problem sizes of about $N = 100$ cities [4]. In order to tackle larger size problems, heuristic algorithms, leading only to suboptimal Steiner trees, have been developed for the Steiner problem. An acceptable suboptimal solution is provided by the MST of the set of cities, which can be computed in an $O(N \log N)$ procedure [41], [31]. The MST also serves as a basis for many heuristics that implement further improvements upon it [26].

In another area of optimization, new algorithms have recently appeared that

* Received by the editors October 3, 1994; accepted for publication (in revised form) September 9, 1996.

http://www.siam.org/journals/siopt/7-4/27506.html

[†] Faculté des Sciences, Université d'Angers, 2 boulevard Lavoisier, 49000 Angers, France (chapeau@ univ-angers.fr).

[‡] Laboratoire d'Ingénierie des Systèmes Automatisés, Université d'Angers, 62 avenue Notre Dame du Lac, 49000 Angers, France (ferrier@istia.univ-angers.fr).

mimic the evolution of physical systems in order to solve optimization problems. Examples are offered by simulated annealing [5], Brownian motion or diffusion [6], neural network models [7], [8], elastic net methods [9], and genetic algorithms [10]. These new approaches have been applied mainly to combinatorial optimization problems, the prototype being the traveling salesman problem [11]. The Steiner problem is not, strictly, a combinatorial optimization problem because the Steiner points that need be introduced for its resolution have positions which, a priori, can vary continuously in the Euclidean plane. This mixed character of the Steiner problem, which exhibits both combinatorial and continuous optimization aspects, adds a special difficulty to its treatment. From an applied standpoint, many practical applications are faced with the Steiner minimal tree problem, as for instance the definition of communication networks or the wiring of electric devices; these can benefit greatly from an efficient resolution of the Steiner problem. In addition, minimal trees can serve as tools for the quantitative characterization of complex sets, branching architectures, or fractally growing structures [12]–[15]. They can also play a role in the representation and processing of data for pattern recognition tasks [16], [17].

In this paper, we propose an $O(N)$ relaxation scheme, inspired from the evolution of a physical system, which is able to relax a given initial Steiner tree into a local minimum of its length. The approach consists of the simulation, in an adapted way, of the dynamics of a fluid film (a soap film) which relaxes under forces due to surface tension, to a configuration that minimizes its total length. When associated with an explicit procedure to construct an initial Steiner tree, the relaxation scheme offers a complete heuristic for the Steiner problem. The relaxation scheme is applied here to an initial tree derived from the MST. The performance of the resulting heuristic is then analyzed and compared for the resolution of Steiner problems with up to $N = 10000$ cities.

2. The Steiner problem. For the Steiner problem as stated in section 1, the lengths are evaluated by means of the usual Euclidean distance. We shall use here the term node to indifferently designate a city or a Steiner point as defined in section 1. We define a Steiner tree as a network of linear edges, which forms a connected graph without a cycle, and connects the given set of nodes. The solution of the Steiner problem is given by the Steiner tree of minimal length or minimal Steiner tree. General properties can be established for the minimal Steiner tree of an N-city set in the Euclidean plane [1]:

(a) Any angle between two edges has to be at least 120°; consequently every node is connected to the minimal tree by at most three edges.

(b) A Steiner point is connected to the minimal tree by exactly three edges, which together form three 120° angles.

(c) The number of Steiner points is at most $N - 2$.

Exact algorithms have been proposed that determine the minimal Steiner tree for a set of N cities [18]–[25] and [4]. See [26], [27], and [4] for recent surveys. All these exact algorithms have exponential complexity in N, making them usable only for small size problems. To date, an upper limit is set in [4] where problems of size up to $N = 100$ are exactly solved.

For larger size problems, heuristics have been proposed [28]–[40] that yield only suboptimal Steiner trees with lengths slightly larger than that of the minimal Steiner tree. See also [26] and [27] for a recent survey.

To evaluate the quality of the solution tree produced by a given algorithm it is useful to compare its length with the length of the MST of the corresponding set of cities. The MST of a set of N cities is the shortest possible tree formed by connecting the cities with $N-1$ linear edges with no addition of Steiner points. An algorithm is available (see [41] and [31]), which relies on the Delaunay triangulation and the Voronoi diagram of the N-city set, to yield its MST in an $O(N \log N)$ procedure. The length reduction R achieved by a given suboptimal Steiner tree over the MST is defined as the ratio

$$R = \frac{\text{length of MST} - \text{length of suboptimal Steiner tree}}{\text{length of MST}} .$$

Different upper bounds have been conjectured and tested for the length reduction R [1], [42]. Recently, a general proof has been given [43] that no tree can be found that achieves a length reduction R larger than $1-\sqrt{3}/2$ (roughly 13.398%). However, for actual Steiner problems that were exactly solved, the length reductions obtained were significantly smaller than this theoretical upper bound. In the exact resolution of [22], the maximum reduction reported is 7.55% for an $N = 5$ city problem, and it drops to 5.77% for an $N = 15$ city problem; the average reduction is 3.08% for $N = 5$ and 3.24% for $N = 15$. In view of these results, it seems that for large values of N the average length reduction R of the exact minimal Steiner tree cannot be expected to be larger than about 3.5%. We shall show in the following that the relaxation scheme we propose, when applied to the MST, achieves length reductions that come close to this value.

3. Description of the relaxation scheme.

3.1. Physical analogy. The relaxation scheme we propose is based on a physical analogy, which is also presented in [44], and which refers to the following phenomenon. A fluid film with high surface tension (typically a soap film) is hooked between pins (the cities of a Steiner problem) and its width is kept constant. Forces due to surface tension are unit forces exerted along the film. Under these forces the film relaxes to an equilibrium configuration that minimizes the potential energy associated with surface tension (gravity is neglected). In the presence of a constant width for the film, this minimum of energy corresponds to a minimum of the length of the film between the pins.

3.2. Initialization. For application to the Steiner problem, the relaxation scheme we propose has to be provided with an initial Steiner tree that will be relaxed into a local minimum of its length. A Steiner tree, in general, incorporates the set of N cities connected through a certain number of Steiner points. The relaxation scheme operates on a special class of initial Steiner trees that conform with a general property of the minimal Steiner tree. In this condition, the final Steiner tree obtained after relaxation of such an initial Steiner tree will generally provide a good approximate solution to the Steiner problem. This special class is defined by the property that each Steiner point in an initial Steiner tree is endowed with exactly three incoming edges connecting it to other nodes of the initial Steiner tree and possibly to itself in some situations.

When provided with such an initial Steiner tree, the relaxation scheme then consists of the iterative implementation of two basic processes: an evolution process and an interaction process.

3.3. Evolution process. Each Steiner point S in the Steiner tree is allowed to move under the resultant $\overrightarrow{F}$ (as defined in Fig. 3) of the three surface tension forces exerted by the three edges incoming on S. The displacement of S is proportional to $\overrightarrow{F}$, with a proportionality coefficient λ. To improve the stabilization in a suboptimal Steiner tree when $\overrightarrow{F}$ decreases while the algorithm converges, the parameter λ is gradually reduced to zero with iterations. This prevents the oscillation of a Steiner point S about its equilibrium position in the event that an edge of S with one of its three neighbors has a length approaching zero. The resultant force $\overrightarrow{F}$ on S represents the opposite of the gradient (relative to the coordinates of S) of the sum of the lengths between S and its three neighbors and consequently the opposite of the gradient relative to S of the total length of the tree. The evolution process can thus be viewed as a gradient descent displacing the Steiner point S in the direction yielding, locally, the maximum length reduction to the tree. This gradient descend operates with a fixed topology for the connections between the nodes of the tree. In general, it would terminate early in a poor local minimum of the length of the tree since the topology of connections is not optimized. We shall now introduce the interaction process, which aims at reorganizing the topology of connections to give access to trees with small total length.

3.4. Interaction process. This process consists of the possibility of a reorganization of the connections between two neighboring Steiner points. The interaction process is illustrated in Fig. 1 and takes place as follows. Let us consider a Steiner point S approaching, in the evolution process, another Steiner point S' to which it is connected. Before interaction each one of these two Steiner points possesses three connections, among which is the connection SS' which will remain untouched in the interaction process. The triplet of connections for S are with the set of nodes $\{A_1, A_2, S'\}$ and for S' with the nodes $\{A_3, A_4, S\}$. If, in its approach, S comes within a distance of T from S', then an interaction will be allowed. In the interaction process, S considers the eventuality of exchanging one of its neighbors $\{A_1, A_2\}$ for one of the neighbors $\{A_3, A_4\}$ of S'. For S, to decide this exchange three possible triplets of connections are examined, the current one $\{A_1S, A_2S, S'S\}$ and two potential ones, $\{A_3S, A_2S, S'S\}$ and $\{A_1S, A_4S, S'S\}$. The configuration $\{A_3S, A_4S, S'S\}$ is not interesting since it represents a simple permutation of the situations of S and S' in the tree with no change to its topology of connections. For each of the three possible triplets of connections for S, the resultant force on S is computed (as defined in the evolution process of section 3.3). These three forces are compared based on their magnitudes, and the triplet of connections that produced the maximum resultant force, be it the current configuration, is retained for S. The resulting, complementary, change of node in the exchange is applied to S'. This completes the interaction process.

For the relaxation of a fluid film under surface tension forces, a minimal energy at equilibrium is equivalent to a minimal total length. In such a situation where length is energy, the interaction distance T can be interpreted as a physical temperature for the Steiner tree. One can consider that the Steiner points, around their actual positions in the tree, experience a permanent random thermal motion of magnitude T. Interaction then takes place when the two clouds of diameter T associated with two Steiner points collide. The temperature T of the Steiner tree is gradually decreased to zero during operation in order to gradually reduce the possibility of interaction and to freeze the tree in a minimum of energy.

Within the physical analogy of the relaxation of a fluid film, both the evolution and interaction processes seek to imitate different aspects of the deformation which minimizes the potential energy or total length of the film. The evolution process alone

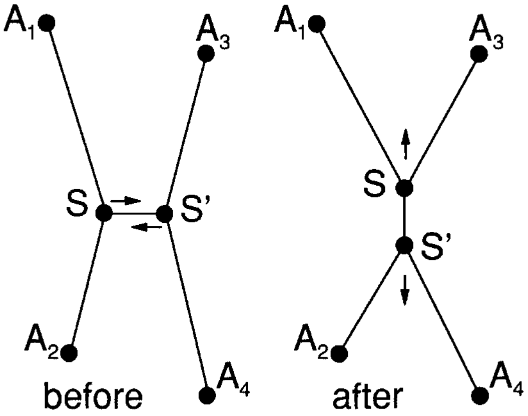


FIG. 1. *Interaction process: reorganization of the two triplets of connections of two neighboring Steiner points S and S' coming within a distance of T and showing the topology before and after interaction.*

displaces the Steiner points along the resultant force, which represents a displacement of the Steiner points in the direction of the maximum length reduction, in the presence of a fixed topology of connections. The interaction process changes the topology of connections to induce locally the maximum resultant force, which represents a change of topology in the direction of the maximum length reduction.

It can be noted that our algorithm bears some similarity with cellular automata [45]. A Steiner point can be considered as an automaton whose state is made up with both the position and the 3-connectivity of the Steiner point. These automata are organized in a network, and they change their state through local interactions with neighbors in the net. The usefulness lies in the collective behavior of the system, which leads, through "microscopic" interactions, to a "macroscopic" configuration realizing a global performance or condition.

4. Application of the relaxation scheme to the MST. When the relaxation scheme is complemented by an explicit procedure to construct an initial Steiner tree, the resulting algorithm offers a complete heuristic for the Steiner problem. We show in the following that a heuristic leading to good suboptimal trees can be obtained when the relaxation scheme is applied to an initial Steiner tree derived from the MST as we now explain.

4.1. An initial Steiner tree derived from the MST. In the MST of the set of N cities, Steiner points are added in order to transform it into the initial Steiner tree that will undergo the relaxation. Figure 2 illustrates how this creation of the Steiner points is performed. In the MST, every city is considered once and processed as follows. For a city with only one incoming edge, no Steiner point is created. For a city with two incoming edges (thus with two neighboring nodes), one Steiner point is created and connected to the city (Fig. 2a). The two neighboring nodes of this city are disconnected from the city and reconnected to the newly created Steiner point. The original city ends up connected to the same pair of neighboring nodes but through the medium of a Steiner point receiving a total of three edges. In a similar way, for a city connected to n neighboring nodes, $n-1$ Steiner points are created. The connections are redistributed between these $1+n+(n-1)$ nodes in order for the original city to end up connected to the same n original nodes but through the medium of the $n-1$ Steiner points, each of them receiving a total of three edges. For illustration, this redistribution of the connections is depicted in Fig. 2a for $n=2$, in Fig. 2b for $n=3$, and in Fig. 2c for $n=4$. A simple geometric argument shows that n cannot be larger than 6, and in practice cities with $n=5$ or 6 incoming edges are very rare in the MST. In practice, the newly created Steiner points are not stacked on top of one another at the location of the original city, but they are distributed around the original city, a very small distance (in comparison with the length scale set by λ) apart from one another and from the city, much like the way they appear in Fig. 2, this in order to avoid a temporary singularity of the type $\overrightarrow{0}/0$ when computing the resultant force on them for the first time. When every city of the initial MST has been processed once as explained, the MST has become the initial Steiner tree, which serves as the starting tree for the relaxation scheme. For an N-city problem, this initialization process creates a total of $N-2$ Steiner points.

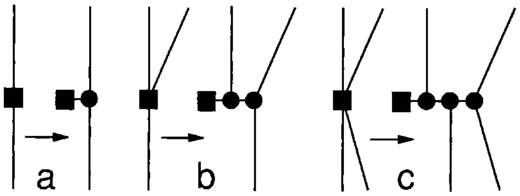


FIG. 2. *Initialization process which transforms the MST into an initial Steiner tree: creation of the Steiner points (solid circles) for a city (solid square) of the MST, with* $n=2$ *in* (a)*,* $n=3$ *in* (b)*, and* $n=4$ *in* (c)*, incoming edges.*

4.2. Operation of the complete heuristic formed by the relaxation of the MST. The relaxation scheme applied to the initial Steiner tree leads to the algorithm described in Fig. 3.

We want to show that the algorithm of Fig. 3, when applied to the initial Steiner tree derived from the MST, provides a good solution tree to the Steiner problem. In the N-city Steiner problems that are considered, the cartesian coordinates of the cities in the Euclidean plane are randomly drawn, with uniform probability, in the unit square $[0,1]\times[0,1]$. In an N-city problem, a natural unit of length is provided by $\sigma_N = N^{-1/2}$. Such a σ_N gives an image of the average separation between a city and its nearest neighboring city in the unit square for an N-city problem. The definition of σ_N allows one to express the parameters λ and T as used in Fig. 3 with numerical values (in units of σ_N) that keep the same meaning whatever the size N of the problem. An initial value for λ that we found satisfactory and that we retained for operation of the algorithm is $0.02\sigma_N$. A larger initial value for λ could make the Steiner tree relax more rapidly to equilibrium, but at the same time useful interactions between Steiner points coming close enough could be missed, leading to an equilibrium Steiner tree of lower quality (of greater length). The initial value for T was selected as $0.15\sigma_N$. Larger values would tend to disorganize the Steiner tree too much, while lower values do not allow enough useful interactions between Steiner points (see Table 1 and its explanation given below).

The schedule that has been used to decrease parameters T and λ is a simple one, consisting of a succession of plateaus of descending values. T is first reduced to zero, at constant λ, in five steps of the same magnitude (equal to one-fifth of the initial value of T): the first step is taken at iteration $k=100$, the four last steps at iterations

```
Set $\sigma_N = N^{-1/2}$ as the unit of length.
Initialization : construct an initial Steiner tree.
Initialize $T$ and $\lambda$ to small fractions of $\sigma_N$.
Iteration step $k = 1$
Repeat
  For each Steiner point $S$ of the Steiner tree
    Evolution :
      $S$ is connected to 3 neighboring nodes $A$, $B$ and $C$
      Compute the resultant force on $S$ as $\overrightarrow{F} = \frac{\overrightarrow{SA}}{\|\overrightarrow{SA}\|} + \frac{\overrightarrow{SB}}{\|\overrightarrow{SB}\|} + \frac{\overrightarrow{SC}}{\|\overrightarrow{SC}\|}$
      Displace $S$ by $\overrightarrow{OS} \longleftarrow \overrightarrow{OS} + \lambda\overrightarrow{F}$
    If among the 3 neighbors of $S$ is a Steiner point distant of less than $T$
      Interaction :
        Call $S'$ the Steiner point neighbor of $S$ such that $\|\overrightarrow{SS'}\| < T$
        ($S'$ can be $A$, $B$ or $C$. If there are more than one
        possible $S'$ then pick one of them at random).
        Exchange neighbors between $S$ and $S'$ as explained in the text
        and in Fig. 1.
    EndIf
  EndFor
  Decrease $T$ and $\lambda$ according to a predefined schedule.
  $k \longleftarrow k + 1$
Until a criterion for convergence is satisfied.
```

FIG. 3. *Complete algorithm for the N-city Steiner problem, which results from the application of the relaxation scheme to an initial Steiner tree.*

$k = 120$, 140, 160, and 180, respectively. Then, λ is allowed to decay. Starting at iteration $k = 200$, the value of λ is divided by two each time 20 new iterations have been performed. Such a process is applied until iteration $k = 400$ is reached. At this point λ has been reduced below $10^{-5}\sigma_N$. This sets the criterion of convergence, marking the end of the algorithm. The overall convergence for an N-city problem can thus be obtained after an absolute number of iterations of 400, whatever the size N of the city set. We did not address the question of optimizing the schedule for decreasing T and λ. The value of 400 iterations for convergence can probably be reduced without degrading the quality of the solution tree. What we aimed at with the presented schedule was to have a simple procedure leading to good equilibrium Steiner trees while preserving a complexity of $O(N)$ for the relaxation scheme when performed until convergence.

The importance of allowing, by means of a nonzero temperature T, interactions between Steiner points is demonstrated in Table 1. We show in Table 1, for problems of various sizes N, the length reduction R (in percents) achieved by the solution tree obtained with different initial values for the temperature T. For each condition, the value of R given in Table 1 has been averaged over 100 different problems of size N. With a zero initial value for T, no interaction is allowed and the length reduction R remains small; as already mentioned R passes through a maximum for an initial T around $0.15\sigma_N$. The role of the interaction process can be interpreted as the ability to select, among the various Steiner tree topologies that are accessible in the vicinity of the initial tree, topologies that can induce local length reductions to the tree.

The relaxation scheme is devised to produce local length reductions to the initial tree through displacements of Steiner points (evolution process) and changes in the topology of connections (interaction process). It can thus be reasonably expected

Table 1

Influence of the initial temperature T (leftmost column) for various problem sizes N, and showing the average length reduction R in percents. The initial value of $0.15\sigma_N$ is the one we retained for T in the application of the relaxation scheme to the MST.

	$N = 50$	$N = 100$	$N = 500$
$T = 0$	1.762	1.665	1.663
$T = 0.01\sigma_N$	2.530	2.598	2.621
$T = 0.15\sigma_N$	2.754	2.824	2.812
$T = 0.25\sigma_N$	1.894	2.003	1.913

(prior to the experimental verification that will follow) that the scheme will converge to a good solution tree with reduced length relative to the initial tree. We emphasize that an important property, which justifies that a fixed number of iterations is appropriate for good convergence, is that the parameters T and λ, which control the local transformations of the tree, scale as $O(N^{-1/2})$. With this property, what our algorithm basically does is to apply, to the MST in which neighboring nodes are separated by distances of $O(N^{-1/2})$, a fixed number of local length reductions at the $O(N^{-1/2})$ scale. This appears to be a reasonable strategy to converge to a solution tree with reduced length relative to the initial MST, without the need to resort to a number of local length reductions that would scale with N instead of being constant. We shall see that this reasonable a priori expectation concerning the convergence is totally confirmed by the experimental evaluation of the algorithm that will follow.

4.3. Algorithmic complexity. The relaxation scheme, formed by the evolution and interaction processes described in section 3, involves only local calculations in the tree at the level of each Steiner point and its three neighbors. For the relaxation of the MST, the initial Steiner tree that is constructed in section 4.1 incorporates a number of Steiner points that is no larger than N. It follows then that the relaxation scheme alone, when performed until the convergence obtained after a fixed number of iterations, has a complexity of $O(N)$.

The transformation of the MST into the initial Steiner tree as described in section 4.1 is also $O(N)$.

Now if we turn to the complete heuristic for the Steiner problem that results from the $O(N)$ relaxation of the MST, the overall complexity obviously will depend on the complexity of the determination of the MST.

An algorithm exists (see [41], [31]) that uses the Delaunay triangulation and the Voronoi diagram of the N-city set to construct its MST in an $O(N \log N)$ procedure.

In this work, to test the quality of the solution trees provided by our $O(N)$ relaxation of the MST, we constructed the MST with the very simple algorithm consisting of growing the MST by incorporating to it at each step the unconnected point that has the shortest distance with the points already connected in the MST. This method is $O(N^3)$; at the same time it is straightforward to implement, and it allowed us to concentrate our effort on the relaxation scheme which forms the original contribution of this work. However, the MST determined for every N-city set is the same, whether computed with the straightforward $O(N^3)$ method we use or the more elaborate $O(N \log N)$ procedure. In the following, we evaluate and compare the quality of the suboptimal trees resulting from the $O(N)$ relaxation of the MST. The association of our $O(N)$ relaxation scheme to the $O(N \log N)$ determination which exists for the MST offers an $O(N \log N)$ heuristic for the Steiner problem that shares the

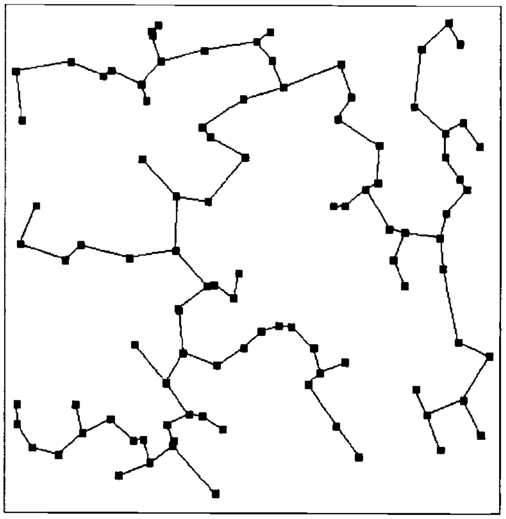


FIG. 4. *The MST, with length* 6.485*, for a typical* $N = 100$ *city problem. The small black squares represent the* 100 *cities randomly distributed in the unit square* $[0, 1] \times [0, 1]$.

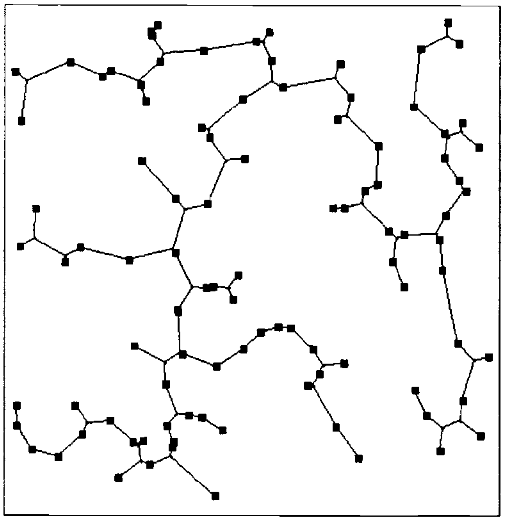


FIG. 5. *The suboptimal Steiner tree, with length* 6.294*, obtained after application of the relaxation scheme to the problem of Fig.* 4 *and achieving a length reduction of* $R = 2.945\%$ *over the MST. A Steiner point is located in every place where three edges meet at* 120°.

performance we report in the following.

We experimentally verified that the relaxation procedure we propose, by itself, requires a computer time which is, as expected, linear in N. When run on an Intel 486 processor with 33 MHz clock, typical computer times for the complete relaxation procedure alone up to convergence (not including the initialization process that computes the MST) are 4 seconds for $N = 100$, 20 seconds for $N = 500$, 40 seconds for $N = 1000$, 200 seconds for $N = 5000$, 400 seconds for $N = 10000$, and with a dispersion among different runs being less than 2%.

4.4. Experimental conditions. For illustration of the method, Fig. 4 shows, for a typical Steiner problem with $N = 100$ cities, the MST (of length 6.485) which serves both as a starting point to construct the initial Steiner tree and as a reference to evaluate the reduction in length reached by the suboptimal Steiner tree.

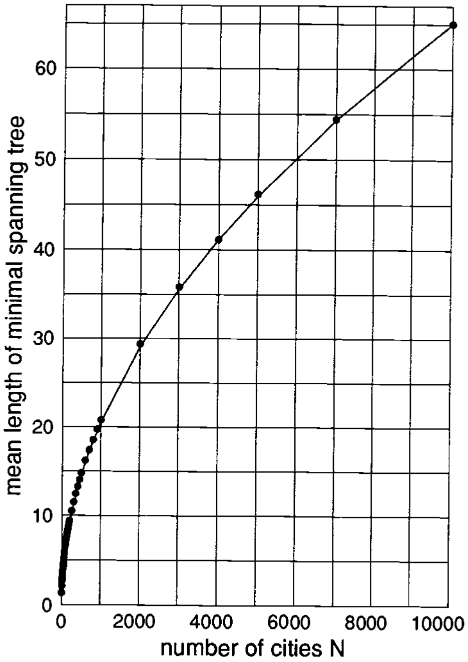


FIG. 6. *Mean length of the MST as a function of the size* N *of the city set, and fitted to a law of the form* $0.65N^{1/2}$ *(solid line) with a correlation coefficient better than* 0.99.

We then show, in Fig. 5, the suboptimal Steiner tree (of length 6.294) obtained after convergence of the relaxation scheme and achieving here a length reduction of $R = 2.945\%$.

With this method that relaxes the MST, we have performed resolution of Steiner problems with sizes up to $N = 10000$ cities. For each tested size N, many different problems were generated by random selection of the N cities as explained in section 4.2, and in order to form a statistical ensemble Ω_N of problems with a given size of N cities. Statistics were then performed over Ω_N which yielded the following quantities:

i) for the MSTs constructed over the N-city problems of Ω_N: the mean and standard deviation for their length distribution;

ii) for the suboptimal Steiner trees obtained after application of the relaxation scheme for the N-city problems of Ω_N: the mean, standard deviation, and minimum and maximum values of the length reduction R.

The evolution of these quantities was then studied as a function of the number of cities N in the Steiner problems. For the statistics, card(Ω_N), the cardinality of Ω_N (the number of problems in Ω_N), was chosen as card$(\Omega_N) = 10^4 N^{-1/2}$ for $N \leq 300$, card$(\Omega_N) = 10^3 N^{-1/2}$ for $300 < N \leq 1000$, and card$(\Omega_N) = 5$ for $N > 1000$.

In these conditions for performing the statistics, Fig. 6 shows the mean of the length of the MST as a function of N. Table 2 displays typical values evaluated for the mean and standard deviation of this length. The data of Fig. 6 and Table 2 give

Table 2

Mean and standard deviation for the length of the MST for various problem sizes N.

N	10	50	100	500	1000	5000	10000
mean	2.092	4.825	6.736	14.826	20.776	46.174	65.028
st. dev.	0.281	0.225	0.217	0.210	0.199	0.145	0.138

an image of the (low) dispersion of the results in the averaging procedure over the statistical ensembles Ω_N. We were able to fit the variation of the mean length of the MST to a law of the form $0.65N^{1/2}$ with a correlation coefficient better than 0.99.

The quality of the solution trees we obtained for the Steiner problems of the statistical ensembles Ω_N is illustrated by the data in the last row of Tables 3 and 4 and in Fig. 7.

5. Evaluation and comparison. In Table 3, the quality of the suboptimal Steiner trees resulting from the relaxation of the MST is compared with that of the solution trees yielded by other resolution methods for the Steiner problem.

As a basis for comparison, we selected

– the exact method of [22], which offers results up to $N = 15$, knowing that for the exact resolutions extended up to $N = 100$ in [4] quantitative data that would fit into our comparison were not available;

– the $O(N \log N)$ heuristic of [31], which represents, among the efficient heuristics, the one with the smallest algorithmic complexity;

– two heuristics of [36] and [37], which represent, among the efficient heuristics, the ones that generally yield the shortest suboptimal trees. No exact algorithmic complexities are derived in [36] or [37] for these two methods, but estimations are proposed, $O(N^{1.317})$ and $O(N^{2.19})$, that result from the average computation time on a Cray X-MP/28.

Table 3 gives, for all these different methods, the maximum value, the mean and the standard deviation of the length reduction R, and the number of problems of size N that were considered for the statistics.

For the maximum length reduction R in Table 3, it can be noticed that, in every condition, the best maximum was always found by our relaxation of the MST. This is certainly because we explored much larger populations of problems. Over the more than 20000 problem instances that we solved in this study, the maximum R that we report come close (11.909% for $N = 5$) but always conform with the theoretical upper bound of 13.398% established in [43].

When compared with the $O(N \log N)$ heuristic of [31], our approach leads in general to solution trees of better quality. This applies except for the mean R in the case $N = 30$ and in the limit case $N = 10$. However, in this last condition the mean length reduction of [31] is found larger than that of the exact method of [22], and it is also the case with the heuristics of [36] and [37]. As there is no possibility that a heuristic yields better results than an exact method, we suggest that the mean value of R for $N = 10$ in [31], as well as in [36] and [37], obtained by averaging over a small population of problems and associated with a relatively high standard deviation, is marked with statistical fluctuations. The heuristic proposed in [31] is tested therein up to $N = 50$. For increasing N approaching $N = 50$, this heuristic of [31] seems to entail a steady decay for the mean R, while our resolution (and that of [36] and [37]) maintain a mean R constantly above 2.710%. This saturation to a constant value as N increases, rather than a steady decay of the mean R, is a trend that will appear

Table 3

Maximum value, mean and standard deviation of the length reduction R (in percents) achieved over the MST, and the number of problems tested with different algorithms for the resolution of the N-city Steiner problem.

	N	5	10	15	20	30	40	50	60
	max. R	7.55	5.89	5.77					
Exponential	mean R	3.08	3.00	3.24					
exact method of [22]	st. dev.								
	nb. pb.	25	25	25					
	max. R		6.847		4.227	4.554	4.014	3.443	
$O(N \log N)$	mean R		3.173		2.333	2.769	2.663	2.568	
heuristic of [31]	st. dev.		2.09		0.70	0.89	0.64	0.57	
	nb. pb.		15		15	15	15	15	
	max. R		6.168		4.737	4.752	4.174	3.620	3.576
$O(N^{1.317})$	mean R		3.138		3.015	2.868	3.024	2.841	2.946
heuristic of [36]	st. dev.		1.863		1.008	0.721	0.631	0.400	0.404
	nb. pb.		15		15	15	15	15	15
	max. R		6.168		4.758	4.838	4.127	3.703	3.666
$O(N^{2.19})$	mean R		3.223		3.123	2.948	2.972	2.921	3.178
heuristic of [37]	st. dev.		1.875		0.972	0.754	0.633	0.423	0.371
	nb. pb.		15		15	15	15	15	15
	max. R	11.909	9.082	8.026	6.088	5.694	5.497	5.531	5.207
Our relaxation scheme	mean R	2.727	2.711	2.744	2.732	2.715	2.712	2.729	2.723
applied to the MST:	st. dev.	2.211	1.515	1.190	1.013	0.827	0.738	0.629	0.600
$O(N \log N)$	nb. pb.	4472	3162	2581	2236	1825	1581	1414	1290

Table 4

Minimum, maximum, mean and standard deviation of the length reduction R (in percents) achieved over the MST, and number of problems tested with two different approaches for the resolution of the N-city Steiner problem.

	N	100	300	500	700	1000	3000	5000	7000	10000
	min. R	2.286		2.668		2.807				
$O(N^{1.317})$	max. R	3.467		3.316		3.283				
heuristic	mean R	2.952		3.052		3.017				3.000
of [36]	st. dev.	0.370		0.169		0.128				
	nb. pb.	15		15		15				1
	min. R	1.162	2.025	2.348	2.459	2.504	2.799	2.772	2.717	2.738
Our relaxation	max. R	4.604	3.622	3.368	3.242	3.052	2.885	2.822	2.787	2.832
scheme applied	mean R	2.755	2.757	2.815	2.803	2.779	2.842	2.791	2.762	2.786
to the MST:	st. dev.	0.468	0.266	0.178	0.180	0.135	0.036	0.022	0.024	0.031
$O(N \log N)$	nb. pb.	1000	577	44	37	31	5	5	5	5

largely confirmed in the following when much larger N's are considered.

When compared with the $O(N^{1.317})$ and $O(N^{2.19})$ heuristics of [36] and [37], it appears that our relaxation of the MST yields slightly longer suboptimal trees. Nevertheless, since our relaxation of the MST represents an $O(N \log N)$ heuristic for the Steiner problem, our approach can still trade off favorably.

The heuristic of [36] offers results that allow us to carry on the comparison above $N = 50$, up to $N = 10000$, as reported in Table 4. In addition, Fig. 7 represents the evolution of the maximum and mean length reduction, that we obtained with our relaxation of the MST, as a function of N.

The data of Table 4 and Fig. 7 show that our relaxation of the MST leads to suboptimal solution trees that keep good positive length reduction over the whole range tested, up to $N = 10000$ cities. For large N, our results confirm over many

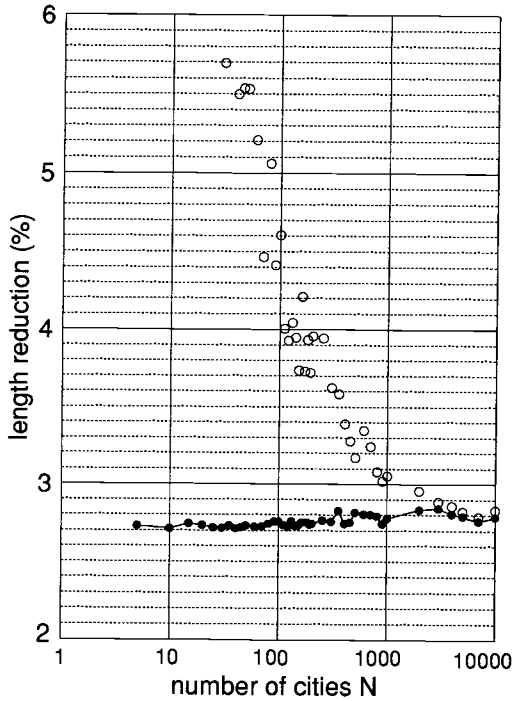


FIG. 7. *Maximum (open circles) and mean (solid circles) of the length reduction R (in percents) achieved by the suboptimal Steiner tree over the MST as a function of the size N of the city set.*

examples, after one first incursion up to $N = 10000$ by [36], that trees can be found that achieve, on average, a nonvanishing length reduction over the MST. These results, as well as those of [36], further indicate that, for large N, the exact minimal trees of Steiner problems also achieve, on average, a nonvanishing length reduction over the MST.

With increasing N, our mean length reduction R in Table 4 and Fig. 7 seems to stabilize to a constant value around 2.8%. This is confirmed both by a maximum value of R which tends to the mean of R and by a standard deviation for R which goes to zero with increasing N. In the same conditions, the heuristic of [36], although much less data are available for it, seems to display the same type of saturation for the mean R but to a higher value around 3.0%. This confirms the fact already observed in Table 3 that the heuristic of [36] yields shorter suboptimal trees on average but still with an algorithmic complexity higher than $O(N \log N)$.

Furthermore, the data of Table 3 reveal that the exact minimal trees (when accessible) and the suboptimal trees found by good heuristics exhibit close values for the mean length reduction. In view of this proximity of behavior, the results of Table 4, although characterizing properties of the *suboptimal trees*, can serve as a basis to conjecture properties of the *minimal trees* of Steiner problems with large N. If we use our extended results of Table 4 to support the possibility of saturation of the mean length reduction R for large N, together with the less numerous results of [36] for a better estimation of the value of this saturation, we can thus propose that the exact minimal trees for Steiner problems with large N will display an average length reduction in the vicinity of 3%. After the analysis of the mean length of the MST, as performed in Fig. 6, we can conjecture that, for large N, the average length of the minimal Steiner tree will be approximately 3% below $0.65N^{1/2}$.

6. Discussion. The heuristic of [36] essentially considers all connected subgraphs of the MST which contain four cities, determines the minimal Steiner tree for each subgraph, which is then incorporated, through some of its Steiner points specially selected, onto the solution tree under construction. This heuristic thus performs a systematic local search for every four-city subgraph of the MST in order to discover the length reduction that is locally optimal (maximal). This heuristic is more of a classic style of combinatorics in graphs. In contrast, our relaxation algorithm closely adheres to a physical analogy that we prove fruitful. It performs length reductions in a uniform and fast way, under the sole control of surface tension forces, in a purely local manner at the level of each Steiner point and its three neighbors. This produces deformations to the tree that are fast with no systematic search of local optimality but with a global convergence to good solution trees as expected from the analogy under test. This simple and uniform procedure results in a heuristic with a low and provable complexity of $O(N \log N)$. The heuristic of [36], relatively more complicated with its systematic local search, shows a higher complexity that is only empirically estimated and, at the same time, slightly shorter solution trees.

Another interesting heuristic has been proposed for the Euclidean Steiner problem [32], [33] which is based on a simulated annealing approach [5]. Works in [35], [46], and [37] also rely, in part, on simulated annealing techniques for the resolution of various versions of the Steiner problem. The heuristic of [32] starts with a random Steiner tree. Tree transformations are implemented which consist in snipping off a randomly selected branch and, after patching the broken branch, attach this to another randomly selected branch. This results in the possibility of constructing any given tree from any other, while permanently preserving the full connectivity of the tree. These tree transformations are then accepted or rejected, depending on the change of length they entail, within the usual probabilistic scheme under the control of a temperature parameter which is gradually reduced [5]. The heuristic is tested in [32] up to $N = 70$ cities. The results in [32] are presented in a way that does not allow them to fit into the comparison of Tables 3 and 4. The quality of the solution trees in [32] is not evaluated against the MST. The scaling of the method with size N of the problem is not addressed in a way that makes possible the precise determination of its algorithmic complexity. When run on an IBM 3081 computer, the best computing times reported are 17 seconds for $N = 20$ and 160 seconds for $N = 50$. With cities chosen uniformly at random in the square $[-10, 10] \times [-10, 10]$, reference [32, page 196] reports for $N = 50$ a typical solution tree of length 1808.54. When rescaled to the unit square $[0, 1] \times [0, 1]$ this gives a length of 90.43, which appears well above the 4.825 mean length of the MST for $N = 50$ as estimated in Table 2 and Fig. 6. Compared with our relaxation scheme that implements only local transformations to the tree at a length scale $O(N^{-1/2})$, the heuristic of [32] realizes random transformations at a length scale $O(1)$ that are unable to keep the complexity below or at $O(N \log N)$ while obtaining performances comparable with ours. This is because a performant solution tree requires length adjustments at the scale $O(N^{-1/2})$, and a total of at least $O(N^{3/2})$ transformations involving $O(1)$-length changes are required for this goal.

Recently, another interesting approach, based on a neural network algorithm, has been proposed for the Steiner problem [38]. This neural method can be described as using a piecewise-linear curve which self-organizes to find a suboptimal tree. In the report of [38] the method, tested up to $N = 100$, never performs better than Beasley's

[36], and its solution trees have lengths that remain, on average, 1.56% above those of [36]. The complexity of the algorithm is not given explicitly in [38], and it is at least $O(N^2)$ since each iteration involves the calculation of a matrix of distances relative to $O(N)$ points.

Another recent heuristic for the Steiner problem is described in [39], then refined and experimentally evaluated in [40]. An interesting characteristic of this heuristic that is shared by very few of the algorithms evoked here is that it can solve Steiner problems in a space of arbitrary dimension, while our approach in its present form, as well as those, for instance, in [22, 4, 31, 36, 37], is limited to the plane. The complexity of this heuristic is not explicitly established, but it is certainly above $O(N \log N)$. For Steiner problems in the plane, the evaluation in [40] is limited to $N = 25$, and the best performance leads to solution trees whose average length is 2.342% below the length of the MST. With our solution trees, in the same conditions, the mean length reduction is always found above 2.710%.

For the Steiner problem in the Euclidean plane, a performance guarantee is proved in [47] which states the existence of a polynomial-time heuristic that will display a performance ratio (the minimum ratio of lengths between the minimal Steiner tree and the approximation solution for the same set of cities) strictly larger than the Steiner ratio $\sqrt{3}/2$. Reference [47], relying on the recent work of [48], also suggests a polynomial-time greedy algorithm that does not use the MST and that has the performance guarantee mentioned above. Although polynomial, the complexity of this heuristic in the plane is not given explicitly in [47], and it may be large and is certainly larger than quadratic. Also, the property that is proved in [47] does not exclude the possibility of obtaining a suboptimal tree longer than the MST for given problems. A performance guarantee with our $O(N \log N)$ heuristic relaxing the MST is the obtainment of a solution which can, at least, be made as good as this tree. Furthermore, the experimental results of Table 4 show that our algorithm was always found to converge to a solution tree strictly shorter than the initial MST.

7. Conclusion.

We have presented a heuristic for the Steiner problem which is based on a physical analogy with the relaxation of a fluid film under surface tension forces. A uniform and purely local evolution scheme results for the Steiner tree, which translates into a low and provable complexity of $O(N \log N)$ for the heuristic, and allows us to tackle very large problems. The performance of this heuristic was compared with that of the best available heuristics with low complexity. Compared with [31], which represents the heuristic with the smallest complexity, our method generally leads, with the same low complexity of $O(N \log N)$, to shorter solution trees. In turn, the heuristics of [36] and [37] lead in general to solution trees slightly shorter than ours but with complexities higher than our $O(N \log N)$.

Beyond these quantitative performances we want to emphasize a specific character of our method, which is to put the Steiner problem in the more novel framework of analog relaxation of a physical type, establishing a connection with energy minimization in physical systems that revealed a fruitful analogy in other areas of optimization. In contrast, the other known algorithms with comparable performance are more of a classic style of combinatorics in graphs.

In particular, our relaxation scheme can be applied to any initial Steiner tree instead of that derived from the MST. With an initial Steiner tree randomly constructed in an $O(N)$ step, we were able to obtain solution trees achieving a positive length reduction R for problem sizes up to $N \approx 100$. This type of approach can lead, for the Steiner problem, to low-complexity heuristics that do not use the MST. More elaborate schedules for the evolution of T, inspired by thermodynamic analogies and incorporating slow cooling and possibly heating phases, may also bring improvement to the performance of the fluid-film relaxation heuristic as introduced here.

REFERENCES

[1] E. N. Gilbert and H. O. Pollak, *Steiner minimal trees*, SIAM J. Appl. Math., 16 (1968), pp. 1–29.

[2] M. Berne and R. Graham, *The shortest-network problem*, Sci. Am., 260 (1989), pp. 84–89.

[3] M. R. Garey, R. L. Graham, and D. S. Johnson, *The complexity of computing Steiner minimal trees*, SIAM J. Appl. Math., 32 (1977), pp. 835–859.

[4] E. J. Cockayne and D. E. Hewgill, *Improved computation of plane Steiner minimal trees*, Algorithmica, 7 (1992), pp. 219–229.

[5] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, *Optimization by simulated annealing*, Science, 220 (1983), pp. 671–680.

[6] B. Gidas, *The Langevin equation as a global minimization algorithm*, in Disordered Systems and Biological Organization, NATO ASI Series F, Vol. 20, E. Bienenstock, F. Fogelman-Soulié, and G. Weisbuch, eds., Springer-Verlag, Berlin, 1986, pp. 321–326.

[7] J. Hopfield and D. W. Tank, *Neural computation of decisions in optimization problems*, Biological Cybernetics, 52 (1985), pp. 141–152.

[8] F. Favata and R. Walker, *A study of the application of Kohonen-type neural networks to the travelling salesman problem*, Biological Cybernetics, 64 (1991), pp. 463–468.

[9] R. Durbin and D. Willshaw, *An analogue approach to the travelling salesman problem using an elastic net method*, Nature, 326 (1987), pp. 689–691.

[10] H. Muhlenbein, M. Georges–Schleuter, and O. Kramer, *Evolution algorithms in combinatorial optimization*, Parallel Comput., 7 (1988), pp. 65–75.

[11] C. Peterson, *Parallel distributed approaches to combinatorial optimization: Benchmark studies on the traveling salesman problem*, Neural Computation, 2 (1990), pp. 261–269.

[12] C. Dussert, G. Rasigni, and A. Llebaria, *Quantization of directional properties in biological structures using the minimal spanning tree*, J. Theoretical Biology, 135 (1988), pp. 295–302.

[13] A. Dress and A. Von Haeseler, eds., *Trees and Hierarchical Structures*, Springer-Verlag, Berlin, 1990.

[14] C. Cherniak, *Local optimization of neuron arbors*, Biological Cybernetics, 66 (1992), pp. 503–510.

[15] R. Van De Weygaert, B. J. T. Jones, and V. J. Martinez, *The minimal spanning tree as an estimator for generalized dimensions*, Phys. Lett. A, 169 (1992), pp. 145–150.

[16] E. Bienenstock and R. Doursat, *Elastic matching and pattern recognition in neural networks*, in Neural Networks from Models to Applications, L. Personnaz and G. Dreyfus, eds., Institut pour le Développement de la Science, L'Education et la Technologie, Paris, 1989, pp. 472–482.

[17] A. L. Yuille, *Generalized deformable models, statistical physics, and matching problems*, Neural Computation, 2 (1990), pp. 1–24.

[18] Z. A. Melzak, *On the problem of Steiner*, Canad. Math. Bull., 4 (1961), pp. 143–148.

[19] E. J. Cockayne, *On the Steiner problem*, Canad. J. Math., 10 (1967), pp. 431–450.

[20] E. J. Cockayne, *On the efficiency of the algorithm for Steiner minimal trees*, SIAM J. Appl. Math., 18 (1970), pp. 150–159.

[21] W. M. Boyce, *An improved program for the full Steiner tree problem*, ACM Trans. Math. Software, 3 (1977), pp. 359–385.

[22] P. Winter, *An algorithm for the Steiner problem in the Euclidean plane*, Networks, 15 (1985), pp. 323–345.

[23] E. J. Cockayne and D. E. Hewgill, *Exact computation of Steiner minimal trees in the plane*, Inform. Process. Lett., 22 (1986), pp. 151–156.

[24] D. Trietsch and F. K. Hwang, *An improved algorithm for Steiner trees*, SIAM J. Appl. Math., 50 (1990), pp. 244–263.

[25] W. D. Smith, *How to find Steiner minimal trees in Euclidean d-Space*, Algorithmica, 7 (1992), pp. 137–177.

[26] F. K. Hwang and D. S. Richards, *Steiner tree problems*, Networks, 22 (1992), pp. 55–89.

[27] F. K. Hwang, D. S. Richards, and P. Winter, *The Steiner Tree Problem*, Annals of Discrete Mathematics 53, North–Holland, Amsterdam, 1992.

[28] S. K. Chang, *The generation of minimal trees with a Steiner topology*, J. ACM, 19 (1972), pp. 699–711.

[29] J. Soukup, *Minimum Steiner trees, roots of a polynomial and other magic*, ACM/SIGMAP Newsletter, 22 (1977), pp. 37–51.

[30] P. Korhonen, *An algorithm for transforming a spanning tree into a Steiner tree*, in Proc. 9th Int. Math. Prog. Symp., Vol. 2, North–Holland, Amsterdam, 1979, pp. 349–357.

[31] J. M. Smith, D. T. Lee, and J. S. Liebman, *An $O(n \log n)$ heuristic for Steiner minimal tree problems on the Euclidean metric*, Networks, 11 (1981), pp. 23–39.

[32] M. Lundy, *Application of the annealing algorithm to combinatorial problems in statistics*, Biometrika, 72 (1985), pp. 191–198.

[33] M. Lundy and A. Mees, *Convergence of an annealing algorithm*, Math. Programming, 34 (1986), pp. 111–124.

[34] W. D. Smith, *Studies in Computational Geometry Motivated by Mesh Generation*, Ph.D. dissertation, Princeton University, Princeton, NJ, 1988.

[35] J. Hesser, R. Manner, and O. Stucky, *Optimization of Steiner tree using genetic algorithms*, in Proc. 3rd Int. Conf. Genetic Algorithms, 1989, pp. 231–236.

[36] J. E. Beasley, *A heuristic for Euclidean and rectilinear Steiner problems*, European J. Operational Research, 58 (1992), pp. 284–292.

[37] J. E. Beasley and F. Goffinet, *A Delaunay triangulation based heuristic for the Euclidean Steiner problem*, Networks, 24 (1994), pp. 215–224.

[38] Jayadeva and B. Bhaumik, *A neural network for the Steiner minimal tree problem*, Biological Cybernetics, 70 (1994), pp. 485–494.

[39] K. Kalpakis and A. T. Sherman, *Probabilistic analysis of an enhanced partitioning algorithm for the Steiner tree problem in R^d*, Networks, 24 (1994), pp. 147–159.

[40] S. Ravada and A. T. Sherman, *Experimental evaluation of a partitioning algorithm for the Steiner tree problem in R^2 and R^3*, Networks, 24 (1994), pp. 409–415.

[41] M. I. Shamos and D. Hoey, *Closest-point problems*, in Proc. 16th Annual Symp. on Foundations of Computer Science, 1975, pp. 151–162.

[42] F. R. K. Chung and F. K. Hwang, *A lower bound for the Steiner tree problem*, SIAM J. Appl. Math., 34 (1978), pp. 27–36.

[43] D. Z. Du and F. K. Hwang, *The Steiner Ratio Conjecture of Gilbert and Pollak Is True*, Proc. Nat. Acad. Sci. U.S.A., 87 (1990), pp. 9464–9466.

[44] R. Courant and H. Robbins, *What Is Mathematics?*, Oxford University Press, New York, 1941.

[45] S. Wolfram, *Theory and Applications of Cellular Automata*, World Scientific, Singapore, 1986.

[46] K. A. Dowsland, *Hill-climbing, simulated annealing and the Steiner problem in graphs*, Engineering Optimization, 17 (1991), pp. 91–107.

[47] D. Z. Du, *On better heuristics for Steiner minimum trees*, Math. Programming, 57 (1992), pp. 193–202.

[48] A. Z. Zelikovsky, *The 11/6-Approximation Algorithm for the Steiner Problem on Networks*, 1992, manuscript.

SIAM J. Optim.
Vol. 7, No. 4, pp. 1054–1068, November 1997

SHORTEST NETWORKS FOR SMOOTH CURVES*

J. F. WENG†

Abstract. In this paper we set up a new model of shortest networks that interconnects a set of smooth curves and avoids a set of smoothly bounded obstacles. Using the hexagonal coordinate system we show how the problem of determining a full Steiner tree with a given topology in such a network can be converted to a problem of solving a set of simultaneous equations. Moreover, the number of equations is linearly dependent on the number of curves and obstacles if all curves and all boundaries of obstacles are convex. Hence, any existing numerical methods and computer programs for solving equations can be used to solve this shortest network problem.

Key words. shortest networks, smoothly bounded obstacles, coordinate system

AMS subject classifications. 05C05,49K99

PII. S1052623494271667

1. Introduction.

Given a set A of points $a_1, a_2, ...$ (referred to as *terminals*) in the Euclidean plane, the shortest network interconnecting A is called the *Steiner minimal tree* on A. The vertices of the tree that are not in the given set are called *Steiner points*. It is well known that Steiner minimal trees satisfy an *angle condition*: all angles at the vertices are not less than 120°. A tree satisfying this angle condition is called a *Steiner tree*. A Steiner tree is called *full* if every terminal is of degree 1. The topology (i.e., the graph structure) of a (full) Steiner tree is called a (*full*) *Steiner topology*. It has been proved [8] that any Steiner topology is a degeneracy of a full Steiner topology caused by the collapse of Steiner points into terminals.

The problem of constructing Steiner minimal trees is usually called the *Steiner problem* [5], [6]. This problem has some generalizations. Instead of points, Cockayne and Melzak [3] considered the network connecting compact sets. Chen [1] determined the network connecting a straight line and two points on the same side. Trietsch [13], [14] studied the more general case: to determine the Steiner network interconnecting given points and existing networks. A number of people [12], [17], [11] have studied shortest networks with polygonal obstacles. In this paper we study a new generalization of the Steiner problem that is defined as follows.

Given: Collection $\mathfrak{C}$ of *objects* $C_1, C_2, ..., C_k$ and collection $\mathfrak{M}$ of *obstacles* $M_1, M_2, ..., M_l$ placed on the plane. Objects and obstacles are assumed to be pairwise disjoint, and the boundaries of objects and obstacles are assumed to be smooth closed curves. As degenerate cases some objects can be *single points*. Denote by $\boldsymbol{C}$ and $\boldsymbol{M}$ the union of the regions in $\mathfrak{C}$ and $\mathfrak{M}$, respectively.

Find: The minimum length network N of curves such that

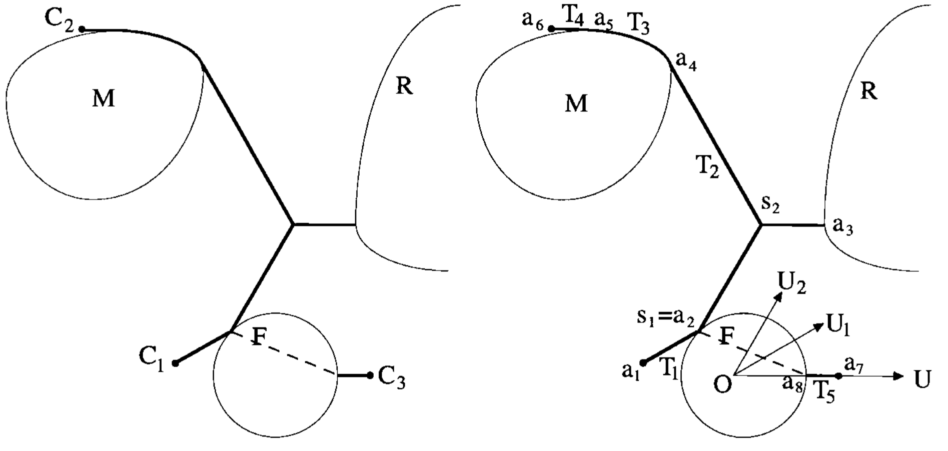
(1) N is disjoint from the interior of $\boldsymbol{M}$, and

(2) $N \cup \boldsymbol{C}$ is a connected set.

We refer to this problem as the shortest networks for smooth curves (SNSC) problem. Comparing the SNSC problem with all other generalizations mentioned above, there are two significant differences:

[*] Received by the editors July 22, 1994; accepted for publication (in revised form) August 12, 1996. This research was supported by a grant from Australian Research Council.
http://www.siam.org/journals/siopt/7-4/27166.html

[†] Department of Mathematics, University of Melbourne, Victoria 3052, Australia (weng@mundoe.maths.mu.oz.au).



(1) (2)

Fig. 1.

(1) Instead of points, the network is required to connect a set of smooth closed curves, the boundaries of objects.

(2) Instead of polygonal lines, all obstacles have smooth boundaries.

Below is an instance of the SNSC problem (Fig. 1(1)): construct a shortest highway network N connecting three cities C_1, C_2, C_3, a river R, and a farm F where the network cannot penetrate a mountain M. Thus, M is an obstacle, C_1, C_2, C_3, F, and R are objects. Since the area of C_1, C_2, C_3 is very small compared with F and M, these cities can be regarded as three single points.

Remark 1. Surely, R cannot be a closed curve. However, since the river is very long, it can be treated as a part of a closed curve.

Remark 2. In the network design problem there are two ways to treat obstacles. Either the edges of the desired network are allowed to penetrate the obstacles with some penalties, or no penetration is allowed at all. In practice, which approach is suitable depends on the conditions of a real problem. In this paper we study the second approach.

By minimality, N must be a forest. More exactly, N can be decomposed into subtrees $T_1, T_2, ..., T_r$ so that each T_i is either

(1) a nontrivial path along the boundary of an obstacle or

(2) a tree made up of straight lines whose only intersection with $\boldsymbol{C}$ and $\boldsymbol{M}$ is at its degree 1 vertices.

The paths of type (1) will be called *joints*, and the trees of type (2) are in fact ordinary *full Steiner trees* defined in the beginning of this section. Hence, all of the degree 1 vertices of a full Steiner tree will be points of $\boldsymbol{C}$ or $\boldsymbol{M}$; these points will be called *terminal (vertices)* of the Steiner tree.

In the example shown in Figure 1(2), N is decomposed into five trees $T_i (1 \leq i \leq 5)$ and has eight terminal vertices $a_i (1 \leq i \leq 8)$. T_3 is a joint while all other trees are full Steiner trees. Note that the road inside F joining two trees of N, marked by a dashed line in Figure 1(2), is a local road and hence is ignored.

Clearly, the endpoints of a joint in N must be terminals of two full Steiner trees. In other words, a joint is fully determined by the full trees that are connected by the joint. Hence, we can temporarily put aside all joints in the process of construction of N. Consequently, if we know how to construct a full Steiner subtree for a given topology, then by exhausting all possible decompositions of the topology of N we can find all feasible solutions and then select the shortest one from them as the required network.

Since the original Steiner problem has been proved to be NP-hard [4], as a generalization, the SNSC problem is also NP-hard. The crucial point in the Steiner problem is that there exists an exponential number of possible full Steiner topologies [6]. However, for a given full Steiner topology on a given set of points, the Steiner tree is easy to construct. Melzak [10] first proposed a method for constructing a full Steiner tree with a given topology. The running time of the improved version of Melzak's method is linear [7]. Later an algebraic method [15], [9], called the *hexagonal coordinate method*, was developed which is equivalent to Melzak's geometric method but more natural and efficient. Along this line the SNSC problem without obstacles was partly studied in [15]. In this paper, by the same approach, we show, given a full Steiner topology, how the problem of constructing a full Steiner subtree in an SNSC problem can be converted to a problem of solving a set of simultaneous equations. Moreover, the number of equations is linearly dependent on the number of curves and obstacles if all curves and all boundaries of obstacles are convex. Hence, any existing numerical methods and computer programs solving equations can be used to solve this shortest network problem.

The paper is organized as follows. Section 2 is a brief review of the hexagonal coordinate method. Sections 3 and 4 give full details of the solution to the SNSC problem for objects and obstacles with convex boundaries. A numerical example is also given. In the last section we discuss how the conditions of smoothness, convexity, and closure can be removed or weakened in this method.

2. The hexagonal coordinate method.

In this section we give a brief review of the hexagonal coordinate method to solve the original Steiner problem [15], [9]. Suppose T is a full Steiner tree (not necessarily minimal). Then its edges have only three directions. This property leads us to consider a coordinate system with three axes such that they are 120° apart. Let O be the origin, axis OU be the 0° line, axis OV be the 120° line, and axis OW be the 240° line. There are three possible definitions of the coordinates of a point p, and we use the following one. Suppose q_u, q_v, q_w are the feet of the lines through p and perpendicular to OU, OV, OW, respectively; then the distances from O to q_u, q_v, q_w are defined to be the coordinates of p and denoted by $u(p), v(p), w(p)$ (or just u, v, w), respectively (Fig. 2).

This coordinate system is called a *hexagonal coordinate system*. Clearly, u, v, w satisfy the following equation which is called the *closure of coordinates*:

$$u + v + w = 0. \tag{1}$$

If the edges of T are not just parallel to the axes, then we can rotate the axes of the *base coordinate system* defined above at a certain angle α so that the new axes OU', OV', OW' are parallel to the edges of T. Define

$$l = \cos\alpha, \quad k = \frac{\sin\alpha}{\sqrt{3}}$$

to be the *rotation coefficients*; then they satisfy the Pythagorean theorem

$$l^2 + 3k^2 = 1. \tag{2}$$

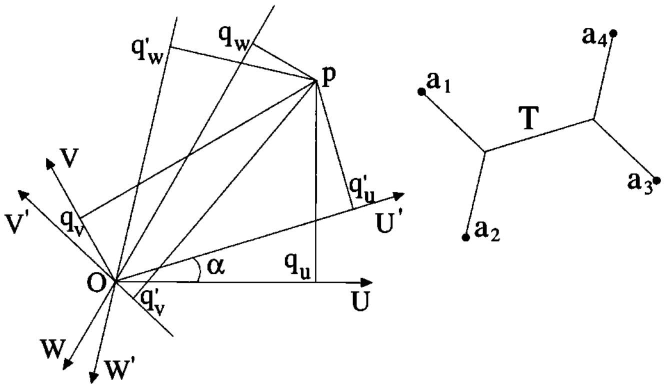


FIG. 2.

It has been proved [15] that the transformations between the old and new coordinates are

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} l & k & -k \\ -k & l & k \\ k & -k & l \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \tag{3}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} l & -k & k \\ k & l & -k \\ -k & k & l \end{bmatrix} \begin{bmatrix} u' \\ v' \\ w' \end{bmatrix}, \tag{4}$$

where u', v', w' are new coordinates of p. Note that by (1) we have

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = 0,$$

and hence

$$\begin{bmatrix} l & k & -k \\ -k & l & k \\ k & -k & l \end{bmatrix} \begin{bmatrix} l & -k & k \\ k & l & -k \\ -k & k & l \end{bmatrix} = \begin{bmatrix} l & -k & k \\ k & l & -k \\ -k & k & l \end{bmatrix} \begin{bmatrix} l & k & -k \\ -k & l & k \\ k & -k & l \end{bmatrix} = I.$$

The advantage of the hexagonal coordinate system is that the coordinates of a Steiner point s are linearly dependent on the coordinates of any two adjacent points in the new system. Suppose s with coordinates u'_s, v'_s, w'_s is adjacent to p_i with coordinates u'_i, v'_i, w'_i $(i = 1, 2, 3)$. If $sp_1 \| OU$, $sp_2 \| OV$, $sp_3 \| OW$, then

$$v'_1 - v'_s = w'_1 - w'_s,\; w'_2 - w'_s = u'_2 - u'_s,\; u'_3 - u'_s = v'_3 - v'_s, \tag{5}$$

and the lengths of these edges are

$$L(sp_1) = |u'_s - u'_1|\,,\; L(sp_2) = |v'_s - v'_2|\,,\; L(sp_3) = |w'_s - w'_3|\,. \tag{6}$$

After changing the subscripts of a_i, we may assume that the anticlockwise circumferential order of the terminals of T is $a_1a_2...$ [8]. If the edge incident with a_i $(i = 1, 2, ...)$ is parallel to $OU'/OV'/OW'$, then its coordinate $u'_i/v'_i/w'_i$ is defined to be the *(first) characteristic coordinate* and referred to as x'_i. Then, $v'_i/w'_i/u'_i$ is the *second characteristic coordinate* and referred to as y'_i, and, finally, $w'_i/u'_i/v'_i$ is the *third characteristic coordinate* and referred to as z'_i. Since T is a full tree, the vertices of T can be partitioned into two subsets so that one subset is labeled as positive, another as negative, and each edge joins a positive vertex with a negative vertex. Define $\epsilon_i = 1$ if a_i is a positive vertex, otherwise $\epsilon_i = -1$. Then, by induction, we can prove [9] that the length of T is

$$L = \sum_i \epsilon_i x'_i, \tag{7}$$

and the following *characteristic equation* holds

$$\sum_i \epsilon_i (y'_i - z'_i) = 0. \tag{8}$$

Note that this equation completely describes the topology of T. Go back to the old (base) hexagonal coordinate system and let x_i, y_i, z_i be the corresponding characteristic coordinates of a_i in the old coordinate system. (For example, if $x'_i = u'_i$, then $x_i = u_i$ and so on.) Then from (1), (3), and (8) we obtain the following *equation of rotation*:

$$l \cdot F_l - 3k \cdot F_k = 0, \tag{9}$$

where $F_l = \sum_i \epsilon_i (y_i - z_i)$, $F_k = \sum_i \epsilon_i x_i$. Now the hexagonal coordinate method can be stated as follows. First, by the given full topology work out its characteristic equation and the expression of the length of T. Next, solve the equation of rotation (9) with the Pythagorean theorem (2) to obtain k, l. Then, calculate the coordinates of all terminals in the new coordinate system by transformation (3) and the coordinates of all Steiner points by (5). Finally, transform these coordinates back into the old coordinate system by (4). As to the length of the tree, we can obtain it directly by (7). A numerical illustration of this method has been given in [9].

3. Determining terminal vertices. In the SNSC problem, in addition to the positions of Steiner points, we need to determine the positions of terminal vertices that are *constrained* on the given closed curves, i.e., the boundaries of objects and obstacles. For simplicity, in this and the next section we assume all curves are convex. Since the curves are smooth, we have the following theorem that is the basis of our method.

THEOREM 3.1. *Suppose p is a terminal vertex of a shortest network N for smooth curves.*

1. *If p lies on the boundary of an object, then p is either of degree 1 or 2. If p is of degree 1, then its edge is the normal line of the curve at p. If p is of degree 2, then the two edges, belonging to two full subtrees, form the same angle with the normal line of the curve at p.*

2. *If p lies on the boundary of an obstacle, then the edge at p is the tangent of the curve.*

Proof. Since any angle in a Steiner tree is not less than 120°, and all edges must lie outside the objects, the degree of p is not more than two when p lies on the boundary of an object. When p is of degree 2, by the minimality of N the two edges at p should satisfy the reflection law of the light in physics. The rest of the theorem is trivial. $\square$

The three kinds of terminals classified in Theorem 3.1 are referred to as *extreme terminals, reflection terminals*, and *tangent terminals*, respectively.

(1) Tangent terminals. Suppose an obstacle has a smooth boundary $f(u,v,w) = 0$. Let $f_x = \frac{\partial f}{\partial x}$. To differentiate the above equation we have

$$df = f_u du + f_v dv + f_w dw = 0.$$

On the other hand, from $u+v+w=0$ we have

$$du + dv + dw = 0.$$

Eliminating dw we obtain $f_u du + f_v dv - f_w du - f_w dv = 0$, i.e.,

$$\frac{dv}{du} = \frac{f_u - f_w}{f_w - f_v}. \tag{10}$$

Similarly we have

$$\frac{dw}{du} = \frac{f_v - f_u}{f_w - f_v}. \tag{11}$$

If the tangent line of $f = 0$ at a point a on its boundary meets the axis OU at angle α, then

$$\tan\alpha = \frac{dy}{dx} = \frac{d\left((v-w)/\sqrt{3}\right)}{du} = \frac{1}{\sqrt{3}}\left(\frac{dv}{du} - \frac{dw}{du}\right)$$

$$= \frac{1}{\sqrt{3}}\left(\frac{f_u - f_w}{f_w - f_v} - \frac{f_v - f_u}{f_w - f_v}\right) = \frac{1}{\sqrt{3}}\left(\frac{2f_u - f_v - f_w}{f_w - f_v}\right), \tag{12}$$

where x, y are the usual Cartesian coordinates. Especially if the line is parallel to OU, then $\tan\alpha = 0$. It follows that

$$2f_u - f_v - f_w = 0.$$

Now suppose the edge at a in N is parallel to the new axis OU'; then the coordinates of a should satisfy $2f_{u'} - f_{v'} - f_{w'} = 0$. To get the expression in the old coordinates, differentiate

$$f(u,v,w) = f(u(u',v',w'), v(u',v',w'), w(u',v',w')) = 0$$

with respect to u',

$$\begin{aligned} f_{u'} = f_u(&u_{u'}u'_{u'} + u_{v'}v'_{u'} + u_{w'}w'_{u'}) \\ +f_v(&v_{u'}u'_{u'} + v_{v'}v'_{u'} + v_{w'}w'_{u'}) \\ +f_w(&w_{u'}u'_{u'} + w_{v'}v'_{u'} + w_{w'}w'_{u'}). \end{aligned}$$

From equation (4), we have $u_{u'} = l,\ u_{v'} = -k$, and so on; from $u' + v' + w' = 0$, we have $u'_{u'} = 1,\ v'_{u'} = -1$, and so on. Substituting them into the expression of $f_{u'}$ we get

$$f_{u'} = f_u l + f_v(2k - l) + f_w(-2k - l).$$

Similarly,

$$\begin{aligned} f_{v'} &= f_u(-2k - l) + f_v l + f_w(2k - l),\\ f_{w'} &= f_u(2k - l) + f_v(-2k - l) + f_w l. \end{aligned}$$

Hence, the condition of a tangent terminal whose incident edge is parallel to OU' is

$$2f_{u'} - f_{v'} - f_{w'} = f_u\,(2l) + f_v\,(3k - l) + f_w\,(-3k - l) = 0,$$

i.e.,

$$l \cdot (2f_u - f_v - f_w) - 3k \cdot (f_w - f_v) = 0.$$

In the same way, we can get the condition of a tangent terminal whose incident edge is parallel to OV' or OW'. This proves the following theorem.

THEOREM 3.2. *If the incident edge of a tangent terminal is parallel to OU' or OV' or OW', respectively, then its coordinates satisfy the optimum condition*

$$l \cdot (2f_u - f_v - f_w) - 3k \cdot (f_w - f_v) = 0 \tag{13}$$

or

$$l \cdot (-f_u + 2f_v - f_w) - 3k \cdot (f_u - f_w) = 0 \tag{14}$$

or

$$l \cdot (-f_u - f_v + 2f_w) - 3k \cdot (f_v - f_u) = 0, \tag{15}$$

respectively.

(2) Extreme terminals. Suppose $f = 0$ is the boundary of an object and a is a point on it. Assume the tangent line at a meets OU at angle α and the normal line meets OU at angle β; then $\tan\alpha \tan\beta = -1$. By equation (12)

$$\tan\beta = -\sqrt{3}\left(\frac{f_w - f_v}{2f_u - f_v - f_w}\right). \tag{16}$$

Especially if the normal line is parallel to OU, then $\tan\beta = 0$, i.e.,

$$f_w - f_v = 0.$$

Now suppose a is an extreme terminal and its incident edge is parallel to OU'; then the condition becomes $f_{w'} - f_{v'} = 0$. Using the same technique as stated above, this condition can be represented in the old coordinates as follows:

$$f_{w'} - f_{v'} = f_u(2k) + f_v(-k - l) + f_w(-k + l) = 0,$$

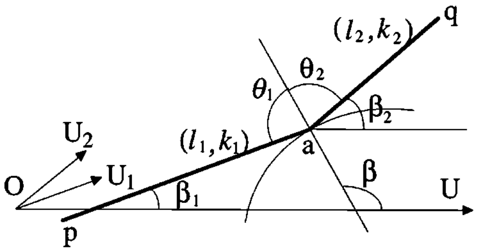


FIG. 3.

i.e.,

$$l \cdot (f_w - f_v) + k \cdot (2f_u - f_v - f_w) = 0.$$

Similarly, we can derive the condition of an extreme terminal whose incident edge is parallel to OV' or OW'. This proves the following theorem.

THEOREM 3.3. *If the incident edge of an extreme terminal is parallel to OU' or OV' or OW', respectively, then its coordinates satisfy the optimum condition*

$$l \cdot (f_w - f_v) + k \cdot (2f_u - f_v - f_w) = 0 \tag{17}$$

or

$$l \cdot (f_u - f_w) + k \cdot (-f_u + 2f_v - f_w) = 0 \tag{18}$$

or

$$l \cdot (f_v - f_u) + k \cdot (-f_u - f_v + 2f_w) = 0, \tag{19}$$

respectively.

(3) Reflection terminals. Suppose two subtrees T_1 and T_2 join at a reflection terminal a which lies on $f = 0$, the boundary of an object. Suppose pa is the edge belonging to subtree T_1 with rotation coefficients l_1, k_1, and aq is another edge belonging to subtree T_2 with rotation coefficients l_2, k_2. Suppose pa or its extension meets OU at angle β_1, aq or its extension meets OU at angle β_2, and the normal line at a meets OU at angle β (Fig. 3).

Let pa, aq meet the normal line at angles θ_1, θ_2, respectively. Then by Theorem 3.1, we have $\theta_1 = \theta_2$ and

$$\tan\theta_1 = \tan(\beta_1 - \beta) = \left(\frac{\tan\beta_1 - \tan\beta}{1 + \tan\beta_1 \tan\beta}\right)$$

$$= \tan\theta_2 = \tan(\beta - \beta_2) = \left(\frac{\tan\beta - \tan\beta_2}{1 + \tan\beta \tan\beta_2}\right).$$

However,

$$\tan\beta_1 = \frac{\sin\beta_1}{\cos\beta_1} = \frac{\sqrt{3}k_1}{l_1}, \ \tan\beta_2 = \frac{\sin\beta_2}{\cos\beta_2} = \frac{\sqrt{3}k_2}{l_2}$$

by the definition of rotation coefficients. Hence, substituting $\tan\beta_1, \tan\beta_2$ with these expressions and substituting $\tan\beta$ with equation (16), after simplification we obtain the following theorem from $\tan\theta_1 = \tan\theta_2$.

THEOREM 3.4. *The optimum condition of reflection terminals is*

$$\begin{aligned}(l_1k_2 + k_1l_2)\left(-2f_u^2 + f_v^2 + f_w^2 + 2f_uf_v + 2f_uf_w - 4f_vf_w\right)\\ + (l_1l_2 - 3k_1k_2)\left(f_w^2 - 2f_uf_w + 2f_uf_v - f_v^2\right) = 0.\end{aligned} \tag{20}$$

4. Constructing SNSC.

As we pointed out in section 1, N is a union of joints and full Steiner trees, and for each possible decomposition we consider only the full Steiner subtrees. Let T be a full subtree whose topology is known. There are two cases.

Case 1. T does not contain reflection terminals. Such a subtree is called an *isolated* tree and can be determined alone. Suppose T spans n terminals $a_i, i = 1, 2, ..., n$. Write down

- the equation of rotation $l \cdot F_l - 3k \cdot F_k = 0$,
- the Pythagorean theorem $l^2 + 3k^2 = 1$,

and for each terminal a_i that is not a single point, its three *associate equations*:

- the closure of its coordinates $u_i + v_i + w_i = 0$,
- the constraint equation $f_i = 0$, and
- the optimum condition by Theorem 3.2 or 3.3.

Solving the $3n+2$ equations simultaneously, we can get l, k, and $3n$ coordinates of a_i.

Case 2. If T is not isolated, then it shares some reflection terminals with other full Steiner trees. In that case, these subtrees have to be solved simultaneously. For example, suppose $T = T_1 \cup T_2$ so that T_1 and T_2 share a reflection terminal a_r. Then write down the equations for the terminals of T_1 and T_2, respectively, as stated above. The two sets of equations, having different rotation coefficients l_1, k_1 and l_2, k_2, are connected by the associate equations of $a_r : u_r + v_r + w_r = 0,\ f_r = 0,$ and the optimum condition by Theorem 3.4. Solving these equations simultaneously, we get l_1, k_1, l_2, k_2, and the coordinates of all constrained terminals including a_r.

Note that the equations stated above, called the *determinative equations* (with respect to the given full topology), are only necessary conditions for the existence of full subtrees. If the determinative equations have no solution, then no subtrees exist for the given topology. However, once a solution exists, then we can determine all Steiner points after the coordinates of terminals are obtained and compute the length of the tree as stated in section 2. Note also that the number of equations is linearly dependent on the number of involved objects and obstacles no matter if T is isolated or not.

On the other hand, since the determinative equations are nonlinear, there are possibly multiple solutions. In fact, for a convex smooth closed curve, there are two tangents through a point outside the curve (Fig. 4(1)). Similar cases also exist for extreme terminals and reflection terminals (Figs. 4(2) and 4(3)).

If in a solution an edge incident to an extreme terminal or a reflection terminal intersects given curves (Figs. 4(2) and 4(3)), then the solution is not a real solution and should be excluded. As to a tangent terminal, it involves two full trees, say T_1 and T_2, that are connected by a joint J lying on the boundary of an obstacle M. Let e_1 and e_2 be the edges in T_1 and T_2 meeting J, respectively. There are three possibilities (Fig. 5). T_1 and T_2 are a feasible pair of subtrees in N only if the extensions of e_1 and e_2 intersect and M lies in the angle formed by the extensions as shown in Fig. 5(3).

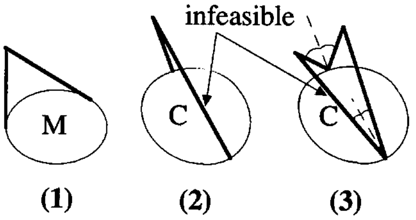


FIG. 4.

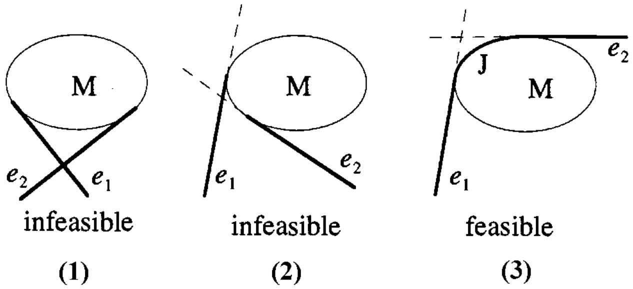


FIG. 5.

Thus, we have converted the problem of determining full subtrees to a problem of solving a set of simultaneous equations. By exhausting all possible decompositions of N, we can find all feasible solutions and select the shortest one as the desired network.

In real problems, preliminary geometric considerations can eliminate many impossible decompositions of N. Below, as an illustration, we use this *hexagonal coordinate method* to solve the example given in the first section. The data are conveniently designed so that the solution can be checked by hand.

Example 1. Suppose the coordinates of C_1, C_2, C_3 and the boundaries of F, R, M are as follows:

C_1 :	$a_1 = (-5, 4, 1)$,
C_2 :	$a_6 = (-11, 22 + \frac{3\sqrt{6}}{2}, -11 - \frac{3\sqrt{6}}{2})$,
C_3 :	$a_7 = (9, -4.5, -4.5)$,
F :	$f = 3(u + 2 - 2\sqrt{3})^2 + (v - w)^2 - 72 = 0$,
R :	$f = (v - w - 18)^2 - 6(u - 6) = 0$,
M :	$f = 9(u + 9)^2 + 5(v - w - 33)^2 - 270 = 0$.

Determine the shortest network N.

Solution. As three single points, C_1, C_2, C_3 are denoted by a_1, a_6, a_7, respectively. First, since a_7 is very closed to F, N should be decomposed into two parts. The part connecting a_7 and F consists of only one edge a_7a_8, where a_8 is an extreme terminal on F. This part is denoted by T_5. Similarly, since a_6 is very closed to M, there should be a tangent point a_5 on M such that a_4a_5 is a tree in N. Let T_4 be this tree and T_3 be the joint meeting T_4 at a_5. The other endpoint of T_3 is denoted by a_4. Let T be the tree connecting a_1, F, R, and a_4. Since F and R are objects, there should be an extreme terminal on each of them in T, say a_2 and a_3, respectively. Thus, T will connect 4 points $a_i, 1 \leq i \leq 4$, and we have a decomposition $N = T \cup T_3 \cup T_4 \cup T_5$. Using the hexagonal coordinate method, it is not hard to obtain the solutions for T_4

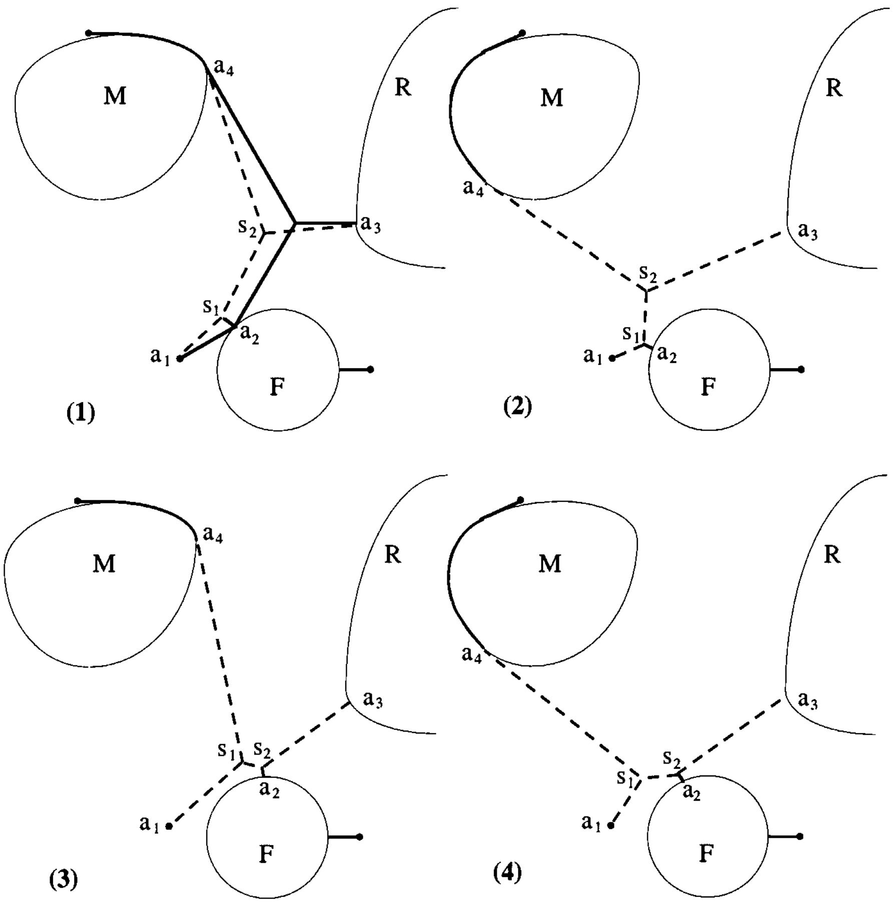


Fig. 6.

and T_5:

$$a_5 = \left(-9, \left(21 + \frac{3\sqrt{6}}{2}\right), \left(-12 - \frac{3\sqrt{6}}{2}\right)\right), \ |T_4| = 2,$$

$$a_8 = \left(2(\sqrt{6} + \sqrt{3} - 1), (1 - \sqrt{6} - \sqrt{3}), (1 - \sqrt{6} - \sqrt{3})\right), |T_5| = 2.6369.$$

Since T joins four points, there are two full topologies: in one topology a_1, a_2 join a Steiner point s_1, and a_3, a_4 join another Steiner point s_2, while in another topology a_1, a_4 join a Steiner point s_1, and a_2, a_3 join another Steiner point s_2. First we compute the tree T with the first topology using the hexagonal coordinate method. Since a_4, as a tangent terminal, may lie on the right side or the left side of M, we obtain 2 solutions as shown in Figs. 6(1) and 6(2).

However, we find that s_1 lies inside F in the solution of Fig. 6(1). Therefore, the solution is infeasible. It means the Steiner point s_1 should collapse into a_2, and T

should be decomposed into two trees: one tree T_1 joins a_1 and a_2, and another tree T_2 connects a_2, a_3, and a_4. So, we recompute T taking a_2 as a reflection terminal. Below are the details.

Suppose the new coordinate system for T_1 is $OU_1V_1W_1$ with rotation coefficients l_1, k_1, and a_1a_2 is parallel to OU_1. Suppose the new coordinate system for T_2 is $OU_2V_2W_2$ with rotation coefficients l_2, k_2, and a_2s_2 is parallel to OU_2.

The equations for T_1 include the following:

- equation of rotation

$$\begin{aligned}&(-v_1 + w_1 + v_2 - w_2) \cdot l_1 - 3\,(-u_1 + u_2) \cdot k_1 \\ &= (-3 + v_2 - w_2) \cdot l_1 - 3\,(5 + u_2) \cdot k_1 = 0,\end{aligned}$$

- Pythagorean theorem, $l_1^2 + 3k_1^2 = 1$.

The equations for T_2 include the following:

- equation of rotation

$$(-v_2 + w_2 - w_3 + u_3 - u_4 + v_4) \cdot l_2 - 3\,(-u_2 - v_3 - w_4) \cdot k_2 = 0,$$

- Pythagorean theorem, $l_2^2 + 3k_2^2 = 1$.

Associate equations of the extreme terminal a_3 are as follows:

$$u_3 + v_3 + w_3 = 0,$$

$$(v_3 - w_3 - 18)^2 - 6(u_3 - 6) = 0,$$

$$\begin{aligned}&l_2 \cdot (f_{u_3} - f_{w_3}) + k_2 \cdot (-f_{u_3} + 2f_{v_3} - f_{w_3}) = \\ &= l_2 \cdot (2v_3 - 2w_3 - 42) + k_2 \cdot (6v_3 - 6w_3 - 102) = 0.\end{aligned}$$

Associate equations of the tangent terminal a_4 are as follows:

$$u_4 + v_4 + w_4 = 0,$$

$$9(u_4 + 9)^2 + 5(v_4 - w_4 - 33)^2 - 270 = 0,$$

$$\begin{aligned}&l_2 \cdot (-f_{u_4} - f_{v_4} + 2f_{w_4}) - 3k_2 \cdot (f_{v_4} - f_{u_4}) \\ &= l_2 \cdot (-18u_4 - 30v_4 + 30w_4 + 828) - 3k_2 \cdot (-18u_4 + 10v_4 - 10w_4 - 492) = 0.\end{aligned}$$

Associate equations of the reflection terminal a_2 are as follows:

$$u_2 + v_2 + w_2 = 0,$$

$$f = 3(u_2 + 2 - 2\sqrt{3})^2 + (v_2 - w_2)^2 - 72 = 0,$$

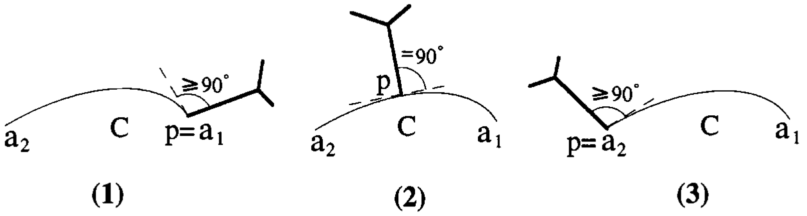


Fig. 7.

$$\begin{aligned}&(l_1k_2+k_1l_2)\left(-2f_{u_2}^2+f_{v_2}^2+f_{w_2}^2+2f_{u_2}f_{v_2}+2f_{u_2}f_{w_2}-4f_{v_2}f_{w_2}\right)\\&\quad+(l_1l_2-3k_1k_2)\left(f_{w_2}^2-2f_{u_2}f_{w_2}+2f_{u_2}f_{v_2}-f_{v_2}^2\right)\\&=(l_1k_2+k_1l_2)\left(-72(u_2+2-2\sqrt{3})^2+24(v_2-w_2)^2\right)\\&\quad(l_1l_2-3k_1k_2)\left(48(u_2+2-2\sqrt{3})(v_2-w_2)\right)=0.\end{aligned}$$

Solving the above set of equations we obtain the solution

$$l_1=\frac{\sqrt{3}}{2},\ \ k_1=\frac{\sqrt{3}}{6},\ \ l_2=\frac{1}{2},\ \ k_2=\frac{1}{2},$$

$$a_2=(-2,4,-2),\ a_{3=}(6,6,-12),\ a_4=(-4,20,-16),\ s_2=(2,8,-10),$$

$$|T_1|=2\sqrt{3},\ |T_2|=24.$$

As to the length of T_3, it equals 5.8473 by the standard line integral.

Similarly, we can compute T with the topology and the specific path round M shown in Figs. 6(2)–6(4). Comparing all four solutions, we find Fig. 6(1) gives the minimal length of T. Thus, we conclude that the network in Fig. 1(2) is the required shortest network.

5. Discussions. In the development of the hexagonal coordinate method, we have set some restrictions: the boundaries of objects and obstacles are smooth, convex closed curves. Now we discuss these conditions and show how to remove or weaken them.

(1) Nonclosed curves. For example, as we said in Remark 1, although the river R in the example is a nonclosed curve, it can be regarded as a part of a closed curve. Such a treatment is especially appropriate if R is very long. However, if R is not very long, then the constrained point a_3 may not be an interior point of R but one of its endpoints, the source or the mouth. Similar cases exist for extreme terminals and reflection terminals on objects. Let us have a close look at the case for extreme terminals. If an object C is a convex nonclosed curve a_1a_2 and if the network N is assumed to be on the convex side of C, then the constrained point on C, say p, has three possible positions as shown in Fig. 7.

If p is assumed to be an interior point, then p should be treated as an extreme terminal on C. Its incident edge in the solution should be perpendicular to C (Fig. 7(2)). If p is assumed to coincide with a_1 or a_2, then p should be treated as a single point located at a_1 or a_2. In the solution, the angle between its incident edge and C should be not less than 90° (Fig. 7(1) and 7(3)). In a word, we need first to compute

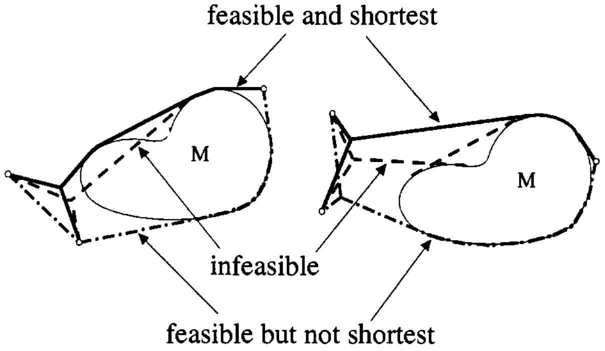


FIG. 8.

the network with the three possible assumptions separately, and then check which assumption is right.

(2) Nonconvex curves. In principle, the hexagonal coordinate method can also be applied to nonconvex curves. For example, if the boundary of an obstacle M is nonconvex, then more than two tangent lines can be drawn from a point outside M. It causes nothing but the number of solutions of the determinative equations to increase, and we need to spend more time to check which one is feasible and is the shortest. Figure 8 shows two of many possibly encountered cases. If an object is not convex, then there may exist many possible positions for extreme points or reflection points. Similarly, we need to check and compare all solutions of the determinative equations that result from the nonconvexity.

(3) Piecewise differentiable curves. If an object is not smooth but piecewise differentiable, then each piece can be treated as stated in (1). For example, suppose an object C is a convex nonclosed curve consisting of two differentiable pieces a_1a_2 and a_2a_3. If the network N is assumed to be on the convex side of C, then the constrained point p on C has five possible positions: p is either an interior point on a_1a_2 or on a_2a_3, or p coincides with a_1, a_2, or a_3. Similarly, if the boundary of an obstacle M is convex, consisting of some differentiable arcs, then the constrained point p on M may be an interior point of an arc or the meeting point of two arcs. We can compute each case with different assumptions of the position of p, and, again, check and compare all solutions of the determinative equations afterwards.

Summing up, based on calculus, the hexagonal coordinate method gives a general approach to the Steiner problem for curves with obstacles. That is, it may apply to the Steiner problem for nonclosed, nonconvex and piecewise differentiable curves that are either objects or boundaries of obstacles, though it causes the computation to increase significantly.

Acknowledgment. The author wishes to thank a referee for a great number of helpful comments and suggestions. Especially, the referee pointed out that the method described in this paper can also be applied to piecewise differentiable curves.

REFERENCES

[1] G. X. CHEN, *The shortest path between two points with a (linear) constraint*, Knowledge and Appl. of Math., 4 (1980), pp. 1–8 (in Chinese).

[2] E. J. Cockayne, *On the efficiency of the algorithm for Steiner minimal trees*, SIAM J. Appl. Math., 18 (1970), pp. 150–159.

[3] E. J. Cockayne and Z. A. Melzak, *Steiner problem for set-terminals*, Quart. Appl. Math., 26 (1967), pp. 213–218.

[4] M. R. Garey, L. R. Graham, and D. S. Johnson, *The complexity of computing Steiner minimal trees*, SIAM J. Appl. Math., 32 (1977), pp. 835–859.

[5] E. N. Gilbert, *Minimum cost communication networks*, Bell System Tech. J., 46 (1967), pp. 2209–2227.

[6] E. N. Gilbert and H. O. Pollak, *Steiner minimal trees*, SIAM J. Appl. Math., 16 (1968), pp. 1–19.

[7] F. K. Hwang, *A linear time algorithm for full Steiner trees*, Oper. Res. Lett., 4 (1986), pp. 235–237.

[8] F. K. Hwang, D. S. Richard, and P. Winter, *The Steiner tree problem*, North–Holland, Amsterdam, The Netherlands, 1992.

[9] F. K. Hwang and J. F. Weng, *Hexagonal coordinate systems and Steiner minimal trees*, Discrete Math., 62 (1986), pp. 49–57.

[10] Z. A. Melzak, *On the problem of Steiner*, Canad. Math. Bull., 4 (1961), pp. 143–148.

[11] J. S. Proven, *An approximation scheme for finding Steiner trees with obstacles*, SIAM J. Comput., 17 (1988), pp. 920–934.

[12] J. M. Smith, *Steiner Minimal Trees with Obstacles*, Tech. report, Dept. of Ind. Eng. and Oper. Res., Univ. of Massachusetts, 1982.

[13] D. Trietsch, *Augmenting Euclidean networks — The Steiner case*, SIAM J. Appl. Math., 45 (1985), pp. 855–860.

[14] D. Trietsch, *Interconnecting networks in the plane: The Steiner case*, Networks, 20 (1990), pp. 93–108.

[15] J. F. Weng, *Generalized Steiner problem and hexagonal coordinate*, Acta. Math. Appl. Sinica, 8 (1985), pp. 383–397 (in Chinese).

[16] J. F. Weng, *Steiner polygons in the Steiner problem*, Geom. Dedicata, 52 (1994), pp. 119–127.

[17] P. Winter and J. M. Smith, *Steiner minimal trees for three points with one convex polygonal obstacle*, Ann. Oper. Res., 33 (1991), pp. 577–599.

SIAM J. Optim.
Vol. 7, No. 4, pp. 1069–1083, November 1997

CONVERGENCE OF PROXIMAL-LIKE ALGORITHMS*

MARC TEBOULLE[†]

Abstract. We analyze proximal methods based on entropy-like distances for the minimization of convex functions subject to nonnegativity constraints. We prove global convergence results for the methods with approximate minimization steps and an ergodic convergence result for the case of finding a zero of a maximal monotone operator. We also consider linearly constrained convex problems and establish a quadratic convergence rate result for linear programs. Our analysis allows us to simplify and extend the available convergence results for these methods.

Key words. convex optimization, proximal methods, maximal monotone operator

AMS subject classifications. 90C25, 90C30

PII. S1052623495292130

1. Introduction.

Consider the convex minimization problem

$$(P) \quad f_* = \inf\{f(x) : \; x \in \mathbb{R}^p_+\}, \tag{1.1}$$

where $f : \; \mathbb{R}^p \mapsto (-\infty, +\infty]$ is a closed proper convex function and $\mathbb{R}^p_+ := \{x \in \mathbb{R}^p \; x_j \geq 0, \, j = 1, \dots, p\}$. Recently [9] we proposed to solve (P) via the iterative scheme: start with $x_0 \in \mathbb{R}^p_{++} := \{x \in \mathbb{R}^p : \; x_j > 0, \, j = 1, \dots, p\}$ and generate the sequence $\{x_k\}$ by

$$x^k = \arg\min_{x \in \mathbb{R}^p} \{f(x) + \lambda_k^{-1} d_\varphi(x, x^{k-1})\}, \tag{1.2}$$

where λ_k is a sequence of positive numbers and $d_\varphi(x, y) := \sum_{j=1}^p y_j \varphi(y_j^{-1} x_j)$ is a distance-like function based on a strictly convex function φ (see section 2 for a precise definition and properties). Algorithm (1.2) is in fact a proximal-type algorithm (see, e.g., Martinet [18], Rockafellar [25]), where here $d_\varphi(\cdot, \cdot)$ replaces the usual quadratic term $1/2||x - x^k||^2$. However, the fundamental difference here is that the term d_φ is used to force the iterates $\{x^k\}$ to stay in the interior of the nonnegative orthant $\mathbb{R}^p_{++}$, namely algorithm (1.2) will automatically generate a *positive* sequence $\{x^k\}$.

The motivation for studying algorithms of the form (1.2) can be found in several recent studies. In [7], the method (1.2) with the particular choice $\varphi(t) = -\log t + t - 1$ was studied and convergence was proved when f is a convex differentiable function with compact level sets and locally Lipschitz-continuous gradients. Convergence results for a more general class of functions φ were recently derived in [9] under weaker assumptions than [7]. The extension of method (1.2) to the more general linearly constrained convex problems and to variational inequalities on polyhedra were recently analyzed in [1]. The application of method (1.2) to the dual functional of a convex program gives rise to several interesting nonquadratic augmented Lagrangian methods [9], [26]. These include, for example, algorithms given in [2], [21], and [27]. These methods have an important practical advantage over the classical augmented

* Received by the editors September 20, 1995; accepted for publication (in revised form) June 19, 1996. This author was partially supported by National Science Foundation grant DMS-9401871 and by Israeli Ministry of Science grant 9636-1-96.

http://www.siam.org/journals/siopt/7-4/29213.html

[†] School of Mathematical Sciences, Tel-Aviv University, Ramat-Aviv 69978, Israel (teboulle@math.tau.ac.il).

Lagrangian, which is derived from the quadratic proximal method, since they preserve the second-order differentiability (if the objectives and constraints are given C^2), and, therefore, Newton-type algorithms can be applied. For further details on the derivation of nonquadratic augmented Lagrangian methods and their properties, we refer the reader to [9], [26], and to the more recent work of [22], which also demonstrates how distances of the type d_φ naturally emerge in the context of constrained optimization.

Convergence analysis of methods like (1.2) has proven to be rather involved and surprisingly difficult; see, e.g., [1], [9], [10]. The main purpose of this work is to present a simplified approach to the convergence analysis of methods based on (1.2) and to prove new convergence results. Building on the works of Güler [8] and Lemaire [15], developed for the classical quadratic proximal methods, we extend their analysis for proximal methods based on (1.2). Starting with two simple and general inequalities for the proximal-like methods (1.2), we develop an elegant analysis which allows us to substantially strengthen and extend available convergence results for these methods. In particular, we establish global convergence results for an inexact proximal-like algorithm based on (1.2), and an ergodic-type convergence result for maximal monotone operators. Similar extensions and convergence results for proximal-like methods based on Bregman functions have been given by Kabbadj [12] and more recently in strengthened form by Kiwiel [11]. However, it should be noted that the analysis of proximal-like methods based on Bregman distances does not carry over to method (1.2), (except for the case $\varphi(t) = t\log t - t + 1$, for which the two distances coincide; see, e.g., [26]). This is due mainly to the fact that the nice "Pythagoras-type" property noticed in [6, Lemma 3.1], which holds for Bregman distances, does not hold in general for the distances d_φ.

In the next section, we give the definition of d_φ, collect some of its properties, and give some examples. In section 3, we state our algorithm and the basic assumptions. In section 4, we present two fundamental estimates and prove global convergence of the methods allowing inexact minimization in (1.2). The convergence of an algorithm based on (1.2) for finding a zero of a maximal monotone operator is analyzed in section 5. The last section considers applications to linearly constrained convex problems and linear programs and extends recent results derived in [1]. In an appendix, we state two results on convergence of nonnegative real sequences. Notation used in this paper and not explicitly defined can be found in Rockafellar's book [24].

2. Distance-like functions: φ-divergences. We start by recalling the definition of φ-divergences and some of their basic properties as used in the context of optimization; see, e.g., [26] and references therein for further details.

Let $\varphi : \mathbb{R} \to (-\infty, +\infty]$ be a closed proper convex function. We denote its domain by $\operatorname{dom}\varphi := \{t : \varphi(t) < +\infty\} \neq \emptyset$ with $\operatorname{dom}\varphi \subseteq [0, +\infty)$. We assume that φ satisfies the following:

(i) φ is twice continuously differentiable on $\operatorname{int}(\operatorname{dom}\varphi) = (0, +\infty)$.

(ii) φ is strictly convex on its domain.

(iii) $\lim_{t\to 0^+} \varphi'(t) = -\infty$.

(iv) $\varphi(1) = \varphi'(1) = 0$ and $\varphi''(1) > 0$.[1]

We denote by Φ the class of functions satisfying (i)–(iv). Given $\varphi \in \Phi$, the φ-

[1] Note that the class of functions satisfying (i)–(iii) (except for the second-order differentiability) are the so-called class of Legendre functions; see [24, Section 26] .

divergence d_φ is defined for $x, y \in \mathbb{R}^p_{++}$ by

$$d_\varphi(x,y) = \sum_{j=1}^{p} y_j \varphi(x_j/y_j). \tag{2.1}$$

From the strict convexity of φ and (iv) we immediately obtain

$$\varphi(t) \geq 0 \text{ and } \varphi(t) = 0 \text{ iff } t = 1.$$

Using this fact in (2.1) it can be easily verified that d_φ can be viewed as a (nonsymmetric) distance-like function satisfying

$$d_\varphi(x,y) \geq 0 \text{ and } d_\varphi(x,y) = 0 \text{ iff } x = y \ \ \forall (x,y) \in \mathbb{R}^p_{++} \times \mathbb{R}^p_{++}. \tag{2.2}$$

Given $\varphi \in \Phi$, let $\alpha := \varphi''(1) > 0$, and define the following two subclasses of Φ:

$$\Phi_1 = \{\varphi \in \Phi : \ \varphi'(t) \leq \alpha \log t \ \ \forall t > 0\}, \tag{2.3}$$

$$\Phi_2 = \{\varphi \in \Phi_1 : \ \alpha(1 - 1/t) \leq \varphi'(t) \ \ \forall t > 0\}. \tag{2.4}$$

Example 2.1. It can easily be verified that the first three functions given below are in Φ_2, while the last one is in Φ_1:

$$\varphi_1(t) = t \log t - t + 1, \ \text{dom}\varphi = [0, +\infty); \ \alpha = 1.$$
$$\varphi_2(t) = -\log t + t - 1, \ \text{dom}\varphi = (0, +\infty); \ \alpha = 1.$$
$$\varphi_3(t) = (\sqrt{t} - 1)^2, \ \text{dom}\varphi = [0, +\infty); \ \alpha = 1/2.$$
$$\varphi_4(t) = t + t^{-1} - 2, \ \text{dom}\varphi = (0, +\infty); \ \alpha = 2.$$

The first example φ_1 plays an important role in the convergence analysis of the algorithms based on (1.2). For $\varphi = \varphi_1$ we have

$$d_\varphi(x,y) := H(x,y) = \sum_{j=1}^{p} \ x_j \log \frac{x_j}{y_j} + y_j - x_j, \tag{2.5}$$

which is the so-called Kullback–Leibler relative entropy distance functional [17].

Notice that $H(x,y)$ can be continuously extended to $\mathbb{R}^p_+ \times \mathbb{R}^p_{++}$, adopting the convention that $0 \log 0 = 0$, i.e., H admits points with zero components in its first argument. The next result gives useful properties of H.

LEMMA 2.1.

(i) *The level sets of $H(x,\cdot)$ are bounded for all $x \in \mathbb{R}^p_+$.*

(ii) *If $\{y^k\} \subset \mathbb{R}^p_{++}$ converges to $y \in \mathbb{R}^p_+$, then $\lim_{k\to\infty} H(y, y^k) = 0$.*

(iii) *If $\{z^k\} \subset \mathbb{R}^p_+, \{y^k\} \subset \mathbb{R}^p_{++}$ are such that $\{z^k\}$ is bounded, $\lim_{k\to\infty} y^k = y \in \mathbb{R}^p_+$ and $\lim_{k\to\infty} H(z^k, y^k) = 0$, then $\lim_{k\to\infty} z^k = y$.*

Proof. The proof is elementary using (2.5). □

We will frequently make use of the following useful identity, which is obtained by direct substitution in (2.5):

$$H(c,a) - H(c,b) = \sum_{j=1}^{p} c_j \log b_j/a_j + a_j - b_j \quad \forall a, b \in \mathbb{R}^p_{++}, c \in \mathbb{R}^p_+. \tag{2.6}$$

We conclude this section by giving some important properties of the function φ and the corresponding d_φ, which will be needed in the rest of this paper.

LEMMA 2.2. *Let $\varphi \in \Phi$ and assume that $\varphi \in C^3$ on $\mathbb{R}_{++}$. Then there exists $\eta > 0$ such that*

$$\varphi'(t) \geq \alpha(1 - t^{-1} - \eta(1 - t^{-1})\varphi'(t)) \ \ \forall\, t > 0.$$

Proof. This is just a rewriting of [9, Proposition 2.5], which states that there exists $\eta > 0$ such that

$$\varphi''(1)(t-1) - t\varphi'(t) \leq \varphi''(1)\eta(t-1)\varphi'(t) \quad \forall t > 0.$$

Indeed, with $\alpha = \varphi''(1)$ the above inequality can be rewritten as

$$t\varphi'(t) \geq \alpha(t - 1 - \eta(t-1)\varphi'(t))$$

and which after division by $t > 0$ gives the desired inequality. □

LEMMA 2.3. *Let $z \in \mathbb{R}^p_{++}$ be fixed and $\varphi \in \Phi$. Then, the level sets $L(z, \nu) := \{x \in \mathbb{R}^p_{++} :\ d_\varphi(x, z) \leq \nu\}$ are bounded for all $\nu \geq 0$.*

Proof. It is enough to consider the one-dimensional case, i.e., to show that $h_\zeta(t) := \zeta\varphi(t/\zeta),\ \zeta > 0$ has bounded level sets, which in turn is equivalent to showing that φ has bounded level sets. Since $\{t : \varphi(t) \leq 0\} = \{1\}$ (by strict convexity of φ and (iv)), the conclusion follows from [24, Corollary 8.7.1, p. 70]. □

3. An entropy-like proximal method (EPM).

The Algorithm (1.2) is based on the φ-divergence which, as seen in the previous section, generalized the concept of entropy-like distances. Accordingly, we call the method based on (1.2) an EPM. Problem (P) will be solved by the EPM, allowing approximate computation in the minimization step of (1.2).

We make the following assumptions for problem (P):

(A0) $\inf\{f(x) : x \in \mathbb{R}^p_+\} = f_* > -\infty$.

(A1) $\mathrm{dom} f \cap \mathbb{R}^p_{++} \neq \emptyset$.

The EPM. Given $\varphi \in \Phi,\ x^0 \in \mathbb{R}^p_{++}, \varepsilon_k \geq 0, \lambda_k > 0$, generate the sequence $\{x^k\} \subset \mathbb{R}^p_{++}$ satisfying

$$g^k \in \partial_{\varepsilon_k} f(x^k), \tag{3.1}$$

$$\lambda_k g^k + \Phi'(x^k/x^{k-1}) = 0, \tag{3.2}$$

where

$$\Phi'(b/a) := (\varphi'(b_1/a_1), \ldots, \varphi'(b_p/a_p))^T \ \ \forall a, b \in \mathbb{R}^p_{++} \tag{3.3}$$

and $\partial_\varepsilon f$ denotes the ε-subdifferential of f.

The above algorithm can be considered as an approximate version of the proximal method (1.2) in the following sense. From (3.1), the convexity of φ and the definition of d_φ in (2.1), we obtain, respectively, $\forall u \in \mathbb{R}^p_+$

$$f(u) \geq f(x^k) + \langle u - x^k, g^k \rangle - \varepsilon_k,$$
$$\lambda_k^{-1} d_\varphi(u, x^{k-1}) \geq \lambda_k^{-1} d_\varphi(x^k, x^{k-1}) + \lambda_k^{-1}\langle u - x^k, \Phi'(x^k/x^{k-1})\rangle.$$

Adding the two inequalities and using (3.2) gives

$$f(u) + \lambda_k^{-1} d_\varphi(u, x^{k-1}) \geq f(x^k) + \lambda_k^{-1} d_\varphi(x^k, x^{k-1}) - \varepsilon_k,$$

i.e.,

$$x^k \in \varepsilon_k - \operatorname{argmin}\{f(u) + \lambda_k^{-1} d_\varphi(u, x^{k-1})\}, \tag{3.4}$$

where we use the notation $\varepsilon - \operatorname{argmin} F(x) := \{z : F(z) \leq \inf F + \varepsilon\}$, with F a given function and $\varepsilon \geq 0$.

Lemma 3.1. *For any $y \in \mathbb{R}^p_{++}$ and $\lambda > 0$ we have the following:*

(i) *If (A0) holds, then the function $x \to F(x) := f(x) + \lambda^{-1} d_\varphi(x, y)$ has bounded level sets.*

(ii) *If in addition (A1) holds, then there exists a unique $x(y) \in \mathbb{R}^p_{++}$ such that*

$$x(y) = \operatorname{argmin}_x\{f(x) + \lambda^{-1} d_\varphi(x, y)\}. \tag{3.5}$$

The minimum above is attained at $x(y) > 0$ satisfying

$$-\Phi'\left(\frac{x(y)}{y}\right) \in \lambda \partial f(x(y)), \tag{3.6}$$

where ∂f denotes the subdifferential of f.

Proof. Fix $y, \lambda > 0$. (i) First note that $x \to F(x) := f(x) + \lambda^{-1} d_\varphi(x, y)$ is a closed proper strictly convex function (since $d_\varphi(\cdot, y)$ is strictly convex). Therefore, if the minimum exists it must be unique. To show that $F(x)$ has bounded level sets it suffices to show that for any $\nu \geq f_*$ the level set

$$L(\nu) := \{x : \; F(x) \leq \nu\}$$

is bounded. Let $\nu' := (\nu - f_*)\lambda$ and $L'(\nu') := \{x : \; d_\varphi(x, y) \leq \nu'\}$. Clearly, we have $L(\nu) \subset L'(\nu')$. But from Lemma 2.3, $L'(\nu')$ is bounded and, hence, so is $L(\nu)$.

(ii) By (i), the minimizer $x(y)$ exists and is unique. Moreover, under the additional assumption (A1), writing the optimality conditions for (3.5) and recalling that $\lim_{t \to 0^+} \varphi'(t) = -\infty$ proves that $x(y) > 0$ and that it satisfies (3.6). □

Remark 3.1. Part (ii) of Lemma 3.1 corresponds to the exact version of the EPM, i.e., with $\varepsilon = 0$. The proof of Lemma 3.1 for that version was given in [9] under more stringent assumptions on the problem's data and which appear to be unnecessary.[2]

4. Convergence analysis.

The analysis relies essentially on the following two simple inequalities. Recall in the sequel that the two parameters α and η are fixed and positive (since $\alpha = \varphi''(1) > 0$ and $\eta > 0$ is from Lemma 2.2).

Lemma 4.1. *For any $a, b \in \mathbb{R}^p_{++}$ and $c \in \mathbb{R}^p_+$, we have the following:*

(i) *If $\varphi \in \Phi_1$, and $\varphi \in C^3(0, +\infty)$, then $\langle c - b, \Phi'(b/a)\rangle \leq \alpha[H(c, a) - H(c, b)] + \alpha\eta\langle b - a, \Phi'(b/a)\rangle$.*

(ii) *If $\varphi \in \Phi_2$, then $\langle c - b, \Phi'(b/a)\rangle \leq \alpha[H(c, a) - H(c, b)]$.*

Proof. (i) Since $\varphi \in \Phi_1$, we have $\varphi'(t) \leq \alpha \log t \; \forall t > 0$. Set $t := b_j/a_j$; we then obtain, for each $j = 1, \ldots, p$,

$$c_j \varphi'(b_j/a_j) \leq \alpha c_j \log b_j/a_j. \tag{4.1}$$

From Lemma 2.2 we also obtain

$$-b_j \varphi'(b_j/a_j) \leq \alpha(a_j - b_j + \eta(b_j - a_j)\varphi'(b_j/a_j)). \tag{4.2}$$

[2] We thank K. C. Kiwiel for pointing out this fact.

Adding the two inequalities (4.1) and (4.2), summing over $j = 1, \dots, p$, and using (3.3) we obtain

$$\begin{aligned}\langle c - b, \Phi'(b/a)\rangle &\leq \alpha \left[\sum_{j=1}^{p} c_j \log b_j/a_j + a_j - b_j\right] + \alpha\eta \sum_{j=1}^{p}(b_j - a_j)\varphi'(b_j/a_j)\\ &= \alpha[H(c,a) - H(c,b)] + \alpha\eta\langle b - a, \Phi'(b/a)\rangle,\end{aligned}$$

where the first term in the last equality is from (2.6).

(ii) Since $\varphi \in \Phi_2$, we have $-\varphi'(t) \leq -\alpha(1 - 1/t) \;\; \forall t > 0$, and, hence, for each $j = 1, \dots, p$

$$-b_j\varphi'(b_j/a_j) \leq \alpha(a_j - b_j). \tag{4.3}$$

Proceeding as in the proof of (i), combining (4.1) and (4.3) gives the desired result (ii). □

The following result provides fundamental estimates from which global rate of convergence estimates in terms of function values as well as convergence of the iterates x^k will follow. For simplicity of notation, we will use the following:

$$\delta(x^k, x^{k-1}) := \delta_k = \langle x^k - x^{k-1}, \Phi'(x^k/x^{k-1})\rangle. \tag{4.4}$$

Lemma 4.2. *Let $\{\lambda_k\}$ be an arbitrary sequence of positive numbers and $\sigma_n := \sum_{k=1}^{n} \lambda_k$. Let $\{x^k\}$ be the sequence generated by the EPM given in* (3.1).

(a) *If $\varphi \in \Phi_1$ and $\varphi \in C^3(0, +\infty)$, then $\forall x \in \mathbb{R}^p_+$:*

(i) $\lambda_k(f(x^k) - f(x)) \leq \alpha[H(x, x^{k-1}) - H(x, x^k)] + \alpha\eta\delta_k + \lambda_k\varepsilon_k$.

(ii) $H(x, x^k) \leq H(x, x^{k-1}) + \eta\delta_k + \alpha^{-1}\lambda_k\varepsilon_k \;\; \forall x \in \mathbb{R}^p_+$ *subject to (s.t.)* $f(x) \leq f(x^k)$.

(iii) $\sigma_n(f(x^n) - f(x)) \leq \alpha[H(x, x^0) - H(x, x^n)] + \alpha\eta \sum_{k=1}^{n} \delta_k + \sum_{k=1}^{n} \sigma_k\varepsilon_k$.

(b) *If $\varphi \in \Phi_2$, then $\forall x \in \mathbb{R}^p_+$:*

(i) $\lambda_k(f(x^k) - f(x)) \leq \alpha[H(x, x^{k-1}) - H(x, x^k)] + \lambda_k\varepsilon_k$.

(ii) $H(x, x^k) \leq H(x, x^{k-1}) + \alpha^{-1}\lambda_k\varepsilon_k \;\; \forall x \in \mathbb{R}^p_+$ *s.t.* $f(x) \leq f(x^k)$.

(iii) $\sigma_n(f(x^n) - f(x)) \leq \alpha[H(x, x^0) - H(x, x^n)] + \sum_{k=1}^{n} \sigma_k\varepsilon_k$.

Proof. Using the definition of the ε-subdifferential we have

$$f(x) \geq f(x^k) + \langle g^k, x - x^k\rangle - \varepsilon_k,$$

where $g^k \in \partial_{\varepsilon_k} f(x^k)$. From (3.6), $g^k = -\lambda_k^{-1}\Phi'(x^k/x^{k-1})$. Substituting the latter in the above inequality we then obtain

$$\lambda_k(f(x^k) - f(x)) \leq \langle x - x^k, \Phi'(x^k/x^{k-1})\rangle + \lambda_k\varepsilon_k. \tag{4.5}$$

Case a. Applying Lemma 4.1(i) at the points $c = x, a = x^{k-1}$ and $b = x^k$ and using (4.5) proves (i). The proof of (ii) follows immediately from (i) for any $x \in \mathbb{R}^p_+$ such that $f(x^k) - f(x) \geq 0$. To prove (iii) we first note that by (3.4)

$$x^k \in \varepsilon_k - \operatorname{argmin}\{f(x) + \lambda_k^{-1} d_\varphi(x, x^{k-1})\};$$

i.e., $\forall x \in \mathbb{R}^p_{++}$,

$$f(x) + \lambda_k^{-1} d_\varphi(x, x^{k-1}) \geq f(x^k) + \lambda_k^{-1} d_\varphi(x^k, x^{k-1}) - \varepsilon_k. \tag{4.6}$$

In particular, for $x = x^{k-1}$, recalling that $d_\varphi \geq 0$ and $d_\varphi(x^{k-1}, x^{k-1}) = 0$ we obtain

$$f(x^{k-1}) - f(x^k) \geq \lambda_k^{-1} d_\varphi(x^k, x^{k-1}) - \varepsilon_k \geq -\varepsilon_k. \tag{4.7}$$

Let $\sigma_n = \sum_{k=1}^n \lambda_k$. Using $\sigma_k = \lambda_k + \sigma_{k-1}$ (with $\sigma_0 \equiv 0$), multiplying the above inequality by σ_{k-1}, and summing over $k = 1, \ldots, n$ we obtain

$$\sum_{k=1}^n \sigma_{k-1} f(x^{k-1}) - (\sigma_k - \lambda_k) f(x^k) \geq -\sum_{k=1}^n \sigma_{k-1}\varepsilon_k,$$

which reduces to

$$\sigma_n f(x^n) - \sum_{k=1}^n \lambda_k f(x^k) \leq \sum_{k=1}^n \sigma_{k-1}\varepsilon_k.$$

Now, using Lemma 4.2a(i) and summing over $k = 1, \ldots, n$ we have

$$-\sigma_n f(x) + \sum_{k=1}^n \lambda_k f(x^k) \leq \alpha[H(x, x^0) - H(x, x^n)] + \alpha\eta \sum_{k=1}^n \delta_k + \sum_{k=1}^n \lambda_k \varepsilon_k.$$

Adding the last two inequalities yields

$$\sigma_n(f(x^n) - f(x)) \leq \alpha[H(x, x^0) - H(x, x^n)] + \alpha\eta \sum_{k=1}^n \delta_k + \sum_{k=1}^n (\lambda_k + \sigma_{k-1})\varepsilon_k,$$

which proves (iii), since $\lambda_k + \sigma_{k-1} = \sigma_k$.

Case b: The proof follows the same steps as in Case a, starting now with Lemma 4.1(ii). □

We are now in a position to prove our main convergence result for the case $\varphi \in \Phi_2$. The proof for the case $\varphi \in \Phi_1$ will be similar but requires an additional technical result and will be given in the next theorem. We denote the set of minimizers of f by $X_* := \{x : \ f(x) = \inf_{\mathbb{R}^p_+} f\}$.

THEOREM 4.3. *Let $\varphi \in \Phi_2$ and $\sigma_n = \sum_{k=1}^n \lambda_k$. Then the following hold.*

(i) $f(x^n) - f(x) \leq \alpha\sigma_n^{-1} H(x, x^0) + \sigma_n^{-1} \sum_{k=1}^n \sigma_k \varepsilon_k \ \ \forall x \in \mathbb{R}^p_+$.

(ii) *If $\sigma_n \to \infty$ and $\lambda_k^{-1}\sigma_k\varepsilon_k \to 0$, then $f(x^n) \to f^* = \inf\{f(x) : x \in \mathbb{R}^p_+\}$.*

(iii) *Moreover, if $X_* \neq \emptyset$, $\sigma_n \to \infty$ and $\sum_{k=1}^\infty \lambda_k \varepsilon_k < \infty$, then the sequence $\{x^n\}$ converges to an optimal solution of (P).*

Proof. (i) The proof follows immediately from Lemma 4.2b(iii), since $H(\cdot, \cdot) \geq 0$.

(ii) Passing to the limit in (i), since $\sigma_n \to \infty$ the first term in (i) goes to zero. Invoking Lemma A.1 (see the Appendix) with $a_{nk} := \sigma_n^{-1}\lambda_k$ if $k \leq n, a_{nk} = 0$ otherwise and $u_k := \lambda_k^{-1}\sigma_k\epsilon_k$, we obtain

$$\sum_k a_{nk} u_k = \sigma_n^{-1} \sum_k \sigma_k \varepsilon_k \to 0$$

as $\sigma_n \to \infty$ and $\lambda_k^{-1}\sigma_k\varepsilon_k \to 0$. Therefore, we have $\limsup_{n\to\infty} f(x^n) \leq \inf\{f(x) : x \in \mathbb{R}^p_+\}$, which together with the fact that $f(x^n) \geq \inf\{f(x) : x \in \mathbb{R}^p_+\}$ implies that x^n is a minimizing sequence.

(iii) Let $x^* \in X_*$. Since

$$f(x^k) \geq f(x^*) \quad \forall k, \tag{4.8}$$

we obtain from Lemma 4.2b(ii)

$$H(x^*, x^k) \le H(x^*, x^{k-1}) + \alpha^{-1}\lambda_k \varepsilon_k.$$

Since $\sum_{k=1}^{\infty} \lambda_k \varepsilon_k < \infty$, invoking Lemma A.2 with $v_k := H(x^*, x^k) \ge 0$ and $\beta_k := \alpha^{-1}\lambda_k\varepsilon_k \ge 0$ implies that $\{H(x^*, x^k)\}$ converges, and, hence, by Lemma 2.1(i) that the sequence $\{x^k\}$ is bounded. Let $\{x^{k_j}\}$ be a subsequence converging to $x^\infty \in \mathbb{R}^p_+$. Since $f(x^k) \to f_*$, $f(x^{k_j}) \to f_*$, and, hence, with f being closed we have $f(x^\infty) \le \lim_{k_j\to\infty} f(x^{k_j}) = f_*$ and it follows that $x^\infty \in X^*$. Therefore, $H(x^\infty, x^k)$ converges. Since $\{x^{k_j}\} \in \mathbb{R}^p_{++}$ converges to $x^\infty \in \mathbb{R}^p_+$, then by Lemma 2.1(ii), $H(x^\infty, x^{k_j}) \to 0$, and, hence, $H(x^\infty, x^k) \to 0$. If $y \in \mathbb{R}^p_+$ is another limit point of $\{x^k\}$, then $H(x^\infty, x^{k_j}) \to 0$ as $x^{k_j} \to y$. But by Lemma 2.1(iii) we then have $x^\infty = y$, and, hence, $x^k \to x^\infty \in X_*$. □

Remark 4.1. The above results extend and strengthen the convergence result established in [10] for the exact version of the EPM, i.e., with $\varepsilon_k = 0\ \forall k$. Indeed, to establish convergence, in [10] it was also required that

$$\text{(i)} \lim \lambda_k^{-1} = \infty, \quad \text{(ii)} \sum_{k=1}^{\infty} \lambda_k^{-1} = \infty,$$

while here it is enough to have $\sum \lambda_k \to \infty$ to guarantee global convergence of $\{x^k\}$ (see also section 6). Moreover, a byproduct of our analysis gives for the exact version of the EPM the global rate of convergence estimate

$$f(x^n) - f(x) \le \alpha\sigma_n^{-1} H(x, x^0) \quad \forall x \in \mathbb{R}^p_+. \tag{4.9}$$

Note that this kind of result is much in the spirit of the existing results for the classical quadratic proximal algorithm; see Güler [8] and Lemaire [15]. For proximal-like methods based on Bregman functions, the estimate (4.9) has been derived by Chen and Teboulle [6], and results analogous to Theorem 4.1 have been given by Kiwiel [11].

Remark 4.2. As pointed out by one referee, the condition $\lambda_k^{-1}\sigma_k\varepsilon_k \to 0$ in Theorem 4.3(ii) could be replaced by the simpler condition $\sum \varepsilon_k < +\infty$. First, we note that with the sole condition $\varepsilon_k \to 0$ we have $\liminf_{n\to+\infty} f(x^n) = \inf\{f(x) : x \in \mathbb{R}^p_+\}$. Indeed, from Lemma 4.2b(i) summing over $k = 1, \ldots, n$ we obtain (recalling that $H(\cdot,\cdot) \ge 0$)

$$\sigma_n^{-1} \sum_{k=1}^{n} \lambda_k f(x^k) \le f(x) + \alpha\sigma_n^{-1} H(x, x^0) + \sigma_n^{-1} \sum_{k=1}^{n} \lambda_k \varepsilon_k. \tag{4.10}$$

Passing to the limit as $n \to +\infty$, using Lemma A.1 and [16, Proposition 3.5], it follows from (4.10) that $\liminf_{n\to+\infty} f(x^n) \le \inf\{f(x) : x \in \mathbb{R}^p_{++}\}$, which together with $f(x^n) \ge \inf\{f(x) : x \in \mathbb{R}^p_+\}$ implies that $\liminf_{n\to+\infty} f(x^n) = \inf\{f(x) : x \in \mathbb{R}^p_+\}$. Now, if $\sum \varepsilon_k < +\infty$, then summing the second inequality in (4.7) implies that $\lim_{n\to+\infty} f(x^n)$ exists, and hence we are done.

We now turn to the convergence of the EPM with $\varphi \in \Phi_1$. We will first need the following technical result.

Lemma 4.4. *Let $\varphi \in \Phi$, and let $\{x^k\}$ be generated by the EPM. Assume that $X_* \ne \emptyset$, $\sigma_n \to \infty$ and $\sum \varepsilon_k < \infty$. Then the following hold.*

(i) $\sigma_n^{-1} \sum_{k=1}^{n} \delta_k \to 0$.

(ii) $\sum_{k=1}^{\infty}\delta_k < \infty$ *if* $\lambda_k \in (0,\lambda]$ *for some* $0<\lambda<\infty$*, where* δ_k *is defined in* (4.4).

Proof. First notice that by the gradient inequality for φ with $\varphi(1)=0$, we have $\varphi(t) \le (t-1)\varphi'(t)$. Using the definition of d_φ and (4.4) it follows that

$$0 \le \lambda_k^{-1} d_\varphi(x^k, x^{k-1}) \le \lambda_k^{-1}\langle x^k - x^{k-1}, \Phi'(x^k/x^{k-1})\rangle = \lambda_k^{-1}\delta_k,$$

showing that the sequence $\{\delta_k\}$ is nonnegative.

With $x := x^{k-1}$ in (4.5) and using (4.4) we obtain

$$\lambda_k^{-1}\delta_k \le f(x^{k-1}) - f(x^k) + \varepsilon_k.$$

Summing the above inequality we obtain

$$\begin{aligned}\sum_{k=1}^{n}\lambda_k^{-1}\delta_k &\le f(x^0) - f(x^n) + \sum_{k=1}^{n}\varepsilon_k,\\ &\le f(x^0) - f(x^*) + \sum_{k=1}^{n}\varepsilon_k,\end{aligned}$$

where in the second inequality we used $x^* \in X_*$ with $f(x^n) \ge f(x^*)\ \forall n$. Since we assumed $\sum_{k=1}^{\infty}\varepsilon_k < \infty$, we thus have $\sum_{k=1}^{\infty}\lambda_k^{-1}\delta_k < \infty$, and, hence, $\lambda_k^{-1}\delta_k \to 0$. It is now easy to verify that all the assumptions of Lemma A.1 are satisfied with $a_{nk} := \lambda_k/\sigma_n$ if $k \le n, a_{nk} = 0$ otherwise, and $u_k := \lambda_k^{-1}\delta_k \to 0$, which implies

$$\sum_{k=1}^{n} a_{nk}u_k = \sum_{k=1}^{n}\frac{\lambda_k}{\sigma_n}\frac{\delta_k}{\lambda_k} = \sigma_n^{-1}\sum_{k=1}^{n}\delta_k \to 0 \text{ as } n \to \infty.$$

Finally, to prove (ii) note that since $\lambda_k \in (0,\lambda]$ we have $\lambda^{-1}\sum_{k=1}^{\infty}\delta_k \le \sum_{k=1}^{\infty}\lambda_k^{-1}\delta_k < \infty$. □

THEOREM 4.5. *Let* $\varphi \in \Phi_1$*,* $\varphi \in C^3(0,+\infty)$*, and* $\sigma_n = \sum_{k=1}^{n}\lambda_k$*. Then the following hold.*

(i) $f(x^n) - f(x) \le \alpha\sigma_n^{-1}H(x,x^0) + \alpha\eta\sigma_n^{-1}\sum_{k=1}^{n}\delta_k + \sigma_n^{-1}\sum_{k=1}^{n}\sigma_k\varepsilon_k \ \ \forall x \in \mathbb{R}_+^p$.

(ii) *Let* $X_* \ne \emptyset$*. If* $\sigma_n \to \infty$*, and* $\sum_{k=1}^{\infty}\varepsilon_k < \infty$*, then* $f(x^n) \to f^* = \inf f(x)$.

(iii) *Moreover, under the hypotheses of* (ii)*, if* $\lambda_k \in (0,\lambda]$*, then the sequence* $\{x^n\}$ *converges to an optimal solution of* (P).

Proof. (i) The proof follows immediately from Lemma 4.2a(iii).

(ii) From Lemma 4.2a(i), summing over $k = 1,\dots,n$, we obtain

$$\sigma_n^{-1}\sum_{k=1}^{n}\lambda_k f(x^k) \le f(x) + \alpha\sigma_n^{-1}H(x,x^0) + \alpha\eta\sigma_n^{-1}\sum_{k=1}^{n}\delta_k + \sigma_n^{-1}\sum_{k=1}^{n}\lambda_k\varepsilon_k.$$

Passing to the limit as $n \to +\infty$, noting that by Lemma 4.4(i) the middle term goes to zero; the rest of the proof follows using the same arguments as given in Remark 4.2.

(iii) Let $x^* \in X_*$. Since $f(x^k) \ge f(x^*)\ \ \forall k$, we obtain from Lemma 4.2a(ii)

$$H(x^*, x^k) \le H(x^*, x^{k-1}) + \eta\delta_k + \alpha^{-1}\lambda_k\varepsilon_k.$$

Since λ_k is bounded above, together with $\sum_{k=1}^{\infty}\varepsilon_k < \infty$ we have $\sum_{k=1}^{\infty}\lambda_k\varepsilon_k < \infty$, and by Lemma 4.4(ii) $\sum_{k=1}^{\infty}\delta_k < \infty$. Invoking Lemma A.2 with $v_k := H(x^*, x^k) \ge 0$ and $\beta_k := \alpha\eta\delta_k + \alpha^{-1}\lambda_k\varepsilon_k \ge 0$ implies that $\{H(x^*, x^k)\}$ converges, and, hence, by

Lemma 2.1(i), that the sequence $\{x^k\}$ is bounded. The remainder of the proof is now the same as given in Theorem 4.3(iii). ∎

Remark 4.3. When $\varepsilon_k = 0 \ \forall k$, we recover the convergence result established in [9]. Our proof and analysis is, however, considerably simpler than the one developed in [9] and also provides, as in the case of Φ_2, the global rate of convergence estimate

$$f(x^n) - f(x) \leq \alpha\sigma_n^{-1} H(x, x^0) + \alpha\eta\sigma_n^{-1} \sum_{k=1}^{n} \delta_k.$$

We finally briefly indicate that our analysis could be further simplified by modifying the EPM described in (3.1) and (3.2) in much the same way it is done for the classical quadratic proximal algorithm with approximate minimization steps; see, e.g., [15]. More precisely, one could consider algorithm (3.1)–(3.2) with the additional assumption that the sequence $\{x^k\}$ satisfies

$$F_k(x^k) \leq F_k(x^{k-1}), \tag{4.11}$$

where

$$F_k(x) := f(x) + \lambda_k^{-1} d_\varphi(x, x^{k-1}).$$

Using the definition of F_k, we note that (4.11) implies that

$$f(x^k) + \lambda_k^{-1} d_\varphi(x^k, x^{k-1}) \leq f(x^{k-1}) + \lambda_k^{-1} d_\varphi(x^{k-1}, x^{k-1}) = f(x^{k-1}),$$

namely, since $d_\varphi \geq 0$, that $\{f(x^k)\}$ is nonincreasing. The latter fact allows for deriving our results in an even simpler way. However, it should be noted that to require the nonincreasingness of $\{f(x^k)\}$ may be difficult to realize in practice. We leave to the reader to verify that when $\{x^k\}$ is generated by the EPM satisfying (4.11) one obtains the following modified estimates (compare with Theorems 4.5(i)–4.3(i), respectively):

If $\varphi \in \Phi_1 \cap C^3(0, +\infty)$, then

$$f(x^n) - f(x) \leq \alpha\sigma_n^{-1} H(x, x^0) + \alpha\sigma_n^{-1}\eta \sum_{k=1}^{n} \delta_k + \sigma_n^{-1} \sum_{k=1}^{n} \lambda_k \varepsilon_k \quad \forall x \in \mathbb{R}_+^p.$$

If $\varphi \in \Phi_2$, then

$$f(x^n) - f(x) \leq \alpha\sigma_n^{-1} H(x, x^0) + \sigma_n^{-1} \sum_{k=1}^{n} \lambda_k \varepsilon_k \quad \forall x \in \mathbb{R}_+^p.$$

The convergence results of Theorems 4.3–4.5 then hold for this modified version of the EPM with $\sigma_n \to \infty$, $\varepsilon_k \to 0$ to obtain a minimizing sequence and with $\sum \lambda_k \varepsilon_k < \infty$ to get the global convergence of the sequence $\{x^n\}$ to an optimal solution of (P). A similar convergence result of this type, i.e., assuming that $\{f(x^k)\}$ is nonincreasing, was derived by Kabbadj [12, Theorem 3.6.1] and more recently by Kiwiel [11] for proximal-like methods based on Bregman distances.

5. The EPM for maximal monotone operators.

In this section we extend our analysis to consider the generalization of the EPM to maximal monotone operators. For simplicity of exposition, we will consider the exact version of the algorithm, i.e., $\varepsilon = 0$ and only the case $\varphi \in \Phi_2$.

A set valued map $T:\ \mathbb{R}^p \to \mathbb{R}^p$ is said to be a monotone operator if

$$\langle y'-y, x'-x\rangle \geq 0 \ \ \forall y' \in T(x')\ \forall y \in T(x)$$

$\forall x, x' \in \mathrm{domT} := \{x:\ T(x) \neq \emptyset\}$. A monotone operator is said to be maximal if its graph

$$G(T) = \{(x,y) \in \mathbb{R}^p \times \mathbb{R}^p : y \in T(x)\}$$

is not properly contained in the graph of any other monotone operator.

Let U be a given maximal monotone operator. We want to solve the following problem.

$$\text{Find } \ x^* \in \mathbb{R}^p \ \text{ such that } \ 0 \in T(x^*), \tag{5.1}$$

where

$$T(x) = \begin{cases} U(x) + N_{\mathbb{R}^p_+}(x) & \text{if } x \in \mathbb{R}^p_+, \\ \emptyset & \text{otherwise.} \end{cases}$$

Here $N_{\mathbb{R}^p_+}$ denotes the normal cone of $\mathbb{R}^p_+$ (see [24, p. 215]), which is also maximal monotone with $\mathrm{dom}N_{\mathbb{R}^p_+} = \mathbb{R}^p_+$. Under the assumption $\mathrm{dom}U \cap \mathbb{R}^p_{++} \neq \emptyset$, by [25, Theorem 1], T is also maximal monotone. Note that problem (5.1) is equivalent to the complementary problem associated with the maximal monotone operator U, i.e.,

$$\text{find } \ (x,y) \in \mathbb{R}^p_+ \times \mathbb{R}^p_+ \cap G(U):\ \langle x,y\rangle = 0,$$

while the solution of the convex minimization problem (P) corresponds to the special case $U = \partial f$ under our assumption (A1).

The EPM for solving (5.1) is as follows: given $x^0 > 0$, generate a sequence $\{x^k\}$ satisfying

$$x^k > 0: \ \ 0 \in U(x^k) + \ \lambda_k^{-1}\Phi'(x^k/x^{k-1}). \tag{5.2}$$

We will assume that the sequence $\{x^k\}$ is well defined; i.e., there exists a unique $x^k > 0$ solving (5.2). Some sufficient conditions for the existence of $\{x^k\}$ in the special case $\varphi = \varphi_2$ can be found in the recent work of Auslender and Haddou [1]. More recently, further existence results have also been established in [5], for more general φ, but which also request further assumptions on both the class Φ_2 and also on the operator T (see also Remark 5.1 below).

THEOREM 5.1. *Let* $\{x^k\}$ *be the sequence generated by* (5.2). *Assume that* $T^{-1}(0) \neq \emptyset$, dom $U \cap \mathbb{R}^p_{++} \neq \emptyset$, *and let* $\sigma_n \to \infty$. *Then the following hold.*

(i) $\{x^k\}$ *is bounded.*

(ii) *Every limit point of the averaged sequence* $z^n := \sigma_n^{-1} \sum \lambda_k x^k$ *is a zero of* T.

Proof. Let $(x,y) \in G(T)$. By (5.2) we have $g^k := -\lambda_k^{-1}\Phi'(x^k/x^{k-1}) \in U(x^k)$. Using Lemma 4.1(ii) with $c = x, a = x^{k-1}, b = x^k$ we obtain

$$\lambda_k \langle x - x^k, g_k\rangle \geq \alpha[H(x,x^k) - H(x,x^{k-1})]. \tag{5.3}$$

Since T is monotone and $g^k \in T(x^k) = U(x^k)$ (recall that $x^k > 0$ implies $N_{\mathbb{R}^p_+}(x^k) = 0$) we have

$$\langle x - x^k, y\rangle \geq \langle x - x^k, g^k\rangle \quad \forall (x,y) \in G(T).$$

Using (5.3) we obtain

$$\lambda_k\langle x-x^k,y\rangle \geq \alpha[H(x,x^k)-H(x,x^{k-1})]. \tag{5.4}$$

Since $T^{-1}(0)\neq\emptyset$, with the choice $(x,y)=(x^*,0)\in G(T)$ in (5.4) we obtain

$$H(x^*,x^k)\leq H(x^*,x^{k-1}), \tag{5.5}$$

and, therefore, $\left\{H(x^*,x^k)\right\}$ is decreasing and $\left\{x^k\right\}$ is bounded, proving (i).

Summing (5.4) over $k=1,\dots,n$, using the definition of σ_n and z^n we then obtain

$$\begin{aligned}\langle z^n-x,y\rangle &\leq \alpha\sigma_n^{-1}[H(x,x^0)-H(x,x^n)]\\ &\leq \alpha\sigma_n^{-1}H(x,x^0).\end{aligned}$$

Since x^n is bounded, so is z^n. Let $z^{n_j}\to z^\infty$. Since $\sigma_n\to\infty$, from the last inequality it follows that $\langle z^\infty-x,y\rangle\leq 0\ \forall(x,y)\in G(T)$, which by the maximal monotonicity of T (which is implied by the assumption dom $U\cap\mathbb{R}^p_{++}\neq\emptyset$) means that $0\in T(z^\infty)$. □

Remark 5.1. The above convergence result is not as strong as the one obtained for the special maximal monotone operator ∂f. Even in the case of the classical quadratic proximal algorithm, if in addition $\sum\lambda_k^2<+\infty$, one can prove only ergodic convergence results as the ones derived in Theorem 5.1, see [3]. This, however, should not be too surprising due to the fact that ∂f enjoys additional properties not shared by arbitrary maximal monotone operators and allows us to derive stronger convergence results; see, for example, Bruck [4] for further results and details in the context of quadratic proximal algorithms. With further assumptions on both T and φ, it is also possible to prove global convergence of the sequence $\{x^k\}$, as shown in the recent work of Burachik [5, Chapter 6]. However, some of the assumptions needed on φ in [5] are unfortunately ruling out some interesting particular realizations of the EPM, such as the choice $\varphi=\varphi_2$. For this special case, global convergence was established in [1] under minimal assumptions.

6. Some applications. To further illustrate the simplicity and usefulness of the analysis developed in section 4, we briefly consider in this section an extension of the EPM to linearly constrained convex programs, as recently proposed by Auslender and Haddou [1].

Thus following [1] we consider the more general convex problem

$$(GP)\quad f_*=\inf\{f(x):\ x\in C\},$$

where C is a polyhedral set given by

$$C=\{x\in\mathbb{R}^p:\ Ax\leq b\}$$

with A an $m\times p$ matrix, $b\in\mathbb{R}^m$, and $m\geq p$. We denote by a_i the rows of the matrix A.

Throughout this section we assume for (GP) that (H1) $f_*>-\infty$ and (H2) A is of maximal rank. (The latter is clearly satisfied when $C=\mathbb{R}^p_+$.)

Assume that int$C\neq\emptyset$ and define

$$l_i(x)=b_i-\langle a_i,x\rangle,\ i=1,\dots,m, \tag{6.1}$$

$$L(x)=(l_1(x),\dots,l_m(x)), \tag{6.2}$$

$$D_\varphi(x,y)=d_\varphi(L(x),L(y)), \tag{6.3}$$

$$D(x,y)=H(L(x),L(y)). \tag{6.4}$$

The EPM (in exact form) to solve (GP), which will be called here the GEPM, is then as follows: start with $x^0 \in \operatorname{int}C$ and generate $\{x^k\} \in \operatorname{int}C$ satisfying

$$x^k = \operatorname{argmin}\{f(x) + \mu_k D_\varphi(x, x^{k-1}) : \, x \in \mathbb{R}^p\}. \tag{6.5}$$

Note that for ease of comparison with the results of [1] we use here $\mu_k := 1/\lambda_k$. Three convergence results of the GEPM were established in [1] under the following three different assumptions:

(H3) $\varphi(t) = \varphi_2(t) = -\log t + t - 1$ and $\exists \mu > 0 : \, 0 < \mu_k \leq \mu$.

(H4) $\varphi \in \Phi_1 \cap C^3(0, +\infty)$, $\exists \mu, \bar{\mu} > 0 : \, \bar{\mu} \leq \mu_k \leq \mu$.

(H5) $\varphi \in \Phi_2$, $\exists \mu > 0 : \, 0 < \mu_k \leq \mu$ and $\sum_{k=1}^{\infty} \mu_k = +\infty$.

As pointed out in [1], (H4) imposes serious restrictions on the choice of μ_k. (Note that in [1] the assumption (H4) should also have required that $\varphi \in \Phi_1 \cap C^3(0, +\infty)$.) (H5) allows for relaxing the condition $\mu_k \geq \bar{\mu}$ but still requires $\sum_{k=1}^{\infty} \mu_k = +\infty$. (H3) is the weakest assumption on $\{\mu_k\}$ but handles only the special choice of $\varphi = \varphi_2$. In this particular case, as shown in [1], it is possible to establish an interesting quadratic convergence result for linear programs. However, Auslender and Haddou [1] were not able to extend such a result for a more general class of functions φ such as the one satisfying (H5).

We show below that the analysis of section 4 can be applied to (GP), allowing us to both relax the hypothesis used in [1] and extend their results on the quadratic rate of convergence for linear programming for more general φ than the one considered in (H3). The key ingredient is once again Lemma 4.1. Indeed, using the optimality conditions for (6.5) we obtain

$$g^k - \mu_k \sum_{i=1}^{m} a_i \varphi'(l_i(x^k)/l_i(x^{k-1})) = 0, \tag{6.6}$$

where here $g^k \in \partial f(x^k)$. Using the definition of the subdifferential for the convex function f and (6.6) we then have $\forall x \in C$:

$$f(x^k) - f(x) \leq \langle g^k, x^k - x \rangle \tag{6.7}$$

$$= \mu_k \sum_{i=1}^{m} \langle a_i, x^k - x \rangle \varphi'(l_i(x^k)/l_i(x^{k-1})). \tag{6.8}$$

Let $\varphi \in \Phi_2$. Applying Lemma 4.1(ii) (in $\mathbb{R}^m$) at $c = l(x), a = l(x^{k-1}), b = l(x^k)$ we obtain using (6.2)–(6.4)

$$\sum_{i=1}^{m} \langle a_i, x^k - x \rangle \varphi'(l_i(x^k)/l_i(x^{k-1})) \leq \alpha[D(x, x^{k-1}) - D(x, x^k)]. \tag{6.9}$$

Combining (6.8) and (6.9) we thus obtain $\forall x \in C$

$$f(x^k) - f(x) \leq \alpha \mu_k [D(x, x^{k-1}) - D(x, x^k)]. \tag{6.10}$$

The latter inequality is the basis from which convergence results for the GEPM easily follow. For example, from (6.10), following the proof of Lemma 4.2 we can derive the global estimate

$$f(x^n) - f(x) \leq \alpha \sigma_n^{-1} D(x, x^0) \quad \forall x \in C. \tag{6.11}$$

With $\sigma_n = \sum_{k=1}^{n} \mu_k^{-1} \to \infty$, the convergence of the GEPM then follows with a similar proof as given in Theorem 4.3 (and using [1, Lemma 2.2], which is an extension of Lemma 2.1 for the polyhedral case C). Note that here instead of the assumption (H5) we only require $\sigma_n \to \infty$, which is obviously satisfied if $\mu_k \in (0, \mu),\ \mu > 0$. A similar analysis allows us to prove convergence in the case $\Phi_1 \cap C^3(0, +\infty)$. We omit the details since this can be done much in the same way it was done in section 4 for EPM in that case.

Now consider the GEPM applied to linear programs, i.e., with $f(x) = \langle c, x\rangle$. Let $\rho(x, X_*) := \inf_{y\in X_*} ||x - y||_2$. The next result extends [1, Theorem 3.2], which was proved only under (H3), i.e., for the special case $\varphi(t) = -\log t + t - 1$, to the more general class $\varphi \in \Phi_2$.

THEOREM 6.1. *Let $\varphi \in \Phi_2$. Assume that (H2) holds, $\mu_k \in (0, \mu)$ for some $\mu > 0$, and that the optimal set X_* of (GP) is nonempty. Let $\nu > 0$ and choose $\mu_k \le \min\{\mu, \nu/\min_{1\le i\le m} l_i(x^{k-1})\}\ \forall k \ge 1$. Then the sequences $\{f(x^k)\}$ and $\{\rho(x^k, X_*)\}$ converge quadratically to f_* and 0, respectively.*

Proof. Since X_* is nonempty, from (6.10) we have

$$f(x^k) - f_* \le \alpha\mu_k[D(x^*, x^{k-1}) - D(x^*, x^k)] \quad \forall x^* \in C.$$

The rest of the proof now follows exactly the one given in [1, Theorem 3.2]. □

Appendix. The following result is due to Silverman and Toeplitz, a proof of which can be found for example in [13, Theorem 2, p. 35].

LEMMA A.1. *Let $\{a_{nk}\}$ be a sequence of real numbers satisfying*

(i) $a_{nk} \ge 0\ \forall n = 1, 2, \dots,\ k = 1, 2, \dots,$

(ii) $\sum_{k=1}^{\infty} a_{nk} = 1\ \forall n = 1, 2, \dots,$ *and* $\lim_{n\to\infty} a_{nk} = 0\ \forall k = 1, 2, \dots.$

If $\{u_k\}$ is a sequence such that $\lim_{k\to\infty} u_k = u$, then $\lim_{n\to\infty} \sum_{k=1}^{n} a_{nk}u_k = u$.

LEMMA A.2. *Let $\{v_k\}$ and $\{\beta_k\}$ be nonnegative sequences of real numbers satisfying* (i) $v_{k+1} \le v_k + \beta_k$, (ii) $\sum_{k=1}^{\infty} \beta_k < \infty$. *Then the sequence $\{v_k\}$ converges.*

Proof. The proof is elementary. See also [20, Chapter 2]. □

Acknowledgment. The author is grateful to two referees for their valuable comments and remarks which contributed greatly to improve the results and presentation.

REFERENCES

[1] A. AUSLENDER AND M. HADDOU, *An interior proximal method for convex linearly constrained problems and its extension to variational inequalities*, Math. Programming, 71 (1995), pp. 77–100.

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

[3] H. BREZIS AND P. L. LIONS, *Produit infinis de resolvantes*, Israel J. Math., 29 (1978), pp. 329–345.

[4] R. D. BRUCK, *An iterative solution of a variational inequality for certain monotone operators in Hilbert space*, Bull. Amer. Math. Soc., 81 (1975), pp. 890–892.

[5] R. S. BURACHIK, *Generalized Proximal Point Methods for the Variational Inequality Problem*, Ph.D. thesis, Instituto de Matemàtica Pura e Aplicada, Rio de Janeiro, Brazil, 1995.

[6] G. CHEN AND M. TEBOULLE, *Convergence analysis of a proximal-like minimization algorithm using Bregman functions*, SIAM J. Optim., 3 (1993), pp. 538–543.

[7] P. P. B. EGGERMONT, *Multiplicatively iterative algorithms for convex programming*, Linear Algebra Appl., 130 (1990), pp. 25–42.

[8] O. GÜLER, *On the convergence of the proximal point algorithm for convex minimization*, SIAM J. Control Optim., 29 (1991), pp. 403–419.

[9] A. N. IUSEM, B. SVAITER, AND M. TEBOULLE, *Entropy-like proximal methods in convex programming*, Math. Oper. Res., 19 (1994), pp. 790–814.

[10] A. 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.

[11] K. C. Kiwiel, *Proximal minimization methods with generalized Bregman functions*, SIAM J. Control Optim., 35 (1997), pp. 1142–1168.

[12] S. Kabbadj, *Méthodes proximales entropiques*, Thèse de Doctorat, Université Montpellier II, 1994.

[13] K. Knopp, *Infinite Sequences and Series*, Dover Publications, Inc., New York, 1956.

[14] B. Lemaire, *The proximal algorithm*, Internat. Ser. Numer. Math. 87, J. P. Penot, ed., Birkhauser-Verlag, Basel, 1989, pp. 73–87.

[15] B. Lemaire, *About the convergence of the proximal method*, Lecture Notes in Econom. and Math. Systems 378, D. Pallaschke, ed., 1992, pp. 39–51.

[16] B. Lemaire, *On the convergence of some iterative methods for convex minimization*, Lecture Notes in Econom. and Math. Systems 429, R. Durier and C. Michelot, eds., 1995, pp. 252–268.

[17] F. Liese and I. Vajda, *Convex Statistical Distances*, Teubner, Leipzig, 1987.

[18] B. Martinet, *Perturbation des Méthodes d'Optimisation*, Application, R. A. I. R. O, Analyse numérique, 12 (1978), pp. 153–171.

[19] J. J. Moreau, *Proximité et dualité dans un espace Hilbertien*, Bull. Soc. Math. France, 93 (1965), pp. 273–299.

[20] B. T. Polyak, *Introduction to Optimization*, Optimization Software, Inc., New York, 1987.

[21] R. A. Polyak, *Modified barrier functions (theory and methods)*, Math. Programming, 54 (1992), pp. 177–222.

[22] R. A. Polyak and M. Teboulle, *Nonlinear rescaling and proximal-like methods in convex optimization*, Math. Programming, 76 (1997), pp. 265–284.

[23] R. T. Rockafellar, *On the maximality of sums of nonlinear monotone operators*, Trans. Amer. Math. Soc., 149 (1970), pp. 75–88.

[24] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[25] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.

[26] M. Teboulle, *Entropic proximal mappings with application to nonlinear programming*, Math. Oper. Res., 17 (1992), pp. 670–690.

[27] P. Tseng and D. Bertsekas, *On the convergence of the exponential multiplier method for convex programming*, Math. Programming, 60 (1993), pp. 1–19.

SIAM J. Optim.
Vol. 7, No. 4, pp. 1084–1102, November 1997

SURROGATE PROJECTION METHODS FOR FINDING FIXED POINTS OF FIRMLY NONEXPANSIVE MAPPINGS[*]

KRZYSZTOF C. KIWIEL[†] AND BOŻENA ŁOPUCH[†]

Abstract. We present methods for finding common fixed points of finitely many firmly nonexpansive mappings on a Hilbert space. At every iteration, an approximation to each mapping generates a halfspace containing its set of fixed points. The next iterate is found by projecting the current iterate on a surrogate halfspace formed by taking a convex combination of the halfspace inequalities. This acceleration technique extends one for *convex feasibility problems* (CFPs), since projection operators onto closed convex sets are firmly nonexpansive. The resulting methods are block iterative and, hence, lend themselves to parallel implementation. We extend to accelerated methods some recent results of Bauschke and Borwein [*SIAM Rev.*, 38 (1996), pp. 367–426] on the convergence of projection methods.

Key words. firmly nonexpansive mappings, successive projections, relaxation methods, convex feasibility problems, surrogate inequalities

AMS subject classifications. 49M45, 90C25, 47H09

PII. S1052623495279569

1. Introduction.

Let D be a closed convex nonempty subset of a real Hilbert space $\mathcal{H}$ with inner product $\langle \cdot, \cdot \rangle$ and norm $\| \cdot \|$. Let $T_i : D \to D$ ($i \in I$, $|I| < \infty$) be a collection of *firmly nonexpansive* mappings, i.e.,

$$\|T_i x - T_i y\|^2 \le \langle T_i x - T_i y, x - y \rangle \quad \forall x, y \in D, \tag{1}$$

with fixed-point sets $F_i = \operatorname{Fix} T_i = \{x \in D : x = T_i x\}$, $i \in I$. Our problem is to

$$\text{find, if possible, any point } x \text{ in } \bigcap\nolimits_{i \in I} F_i. \tag{2}$$

This generalizes the following CFP:

$$\text{find, if possible, any point } x \text{ in } \bigcap\nolimits_{i \in I} C_i, \tag{3}$$

where C_i are closed convex subsets of $\mathcal{H}$, since we may let $T_i = P_{C_i}$, where $P_{C_i} x = \arg\min_{y \in C_i} \|x - y\|$ is the projection mapping onto C_i. In turn, (2) is an instance of (3) since each T_i is *nonexpansive* ($\|T_i x - T_i y\| \le \|x - y\| \ \forall x, y \in D$) and F_i is closed and convex [AuE84, Theorem 5.2.1]; thus, (2) and (3) are essentially *equivalent.* Further, if we let each T_i be the *resolvent* $(\bar{I} + A_i)^{-1}$ of some maximal monotone operator A_i in $\mathcal{H}$, where $\bar{I}$ is the identity mapping, then problem (2) reduces to finding a common zero for the A_is, whereas $P_{C_i} = (\bar{I} + N_{C_i})^{-1}$, where N_{C_i} is the (maximal monotone) normal cone operator of C_i [Eck89, EcB92, Tse92]. Next, since (1) holds if and only if $T_i = \frac{1}{2}(\bar{I} + S_i)$ for some nonexpansive mapping S_i [BaB96, Eck89, EcB92, Roc76], problem (2) reduces to finding a common fixed point of the S_is. The earlier references [BrR77, GoR84] contain fundamental material on firmly nonexpansive mappings and their iterations and fixed points.

[*] Received by the editors January 3, 1995; accepted for publication (in revised form) May 20, 1996. This research was supported by the State Committee for Scientific Research under grants 8T11A00409 and 8T11A02610.

http://www.siam.org/journals/siopt/7-4/27956.html

[†] Systems Research Institute, Newelska 6, 01–447 Warsaw, Poland (kiwiel@ibspan.waw.pl, bozena@ibspan.waw.pl)

The recent survey of [BaB96] lists numerous applications and unifies and improves many algorithms for the CFP; other improvements are given in [Flå95]. Another recent survey of [Kiw95] is restricted to finite-dimensional CFPs but gives strong convergence results for both the "short-step" methods discussed in [BaB96, Flå95] and the accelerated "long-step" methods stemming from [Mer62] (see also [GP93]). Our study [Kiw95] was motivated mainly by possible applications of methods for the CFP in algorithms for convex nondifferentiable optimization [Kiw96a, Kiw96b, Kiw97].

This paper establishes convergence of long-step methods for (2) derived via the acceleration techniques of [Kiw95]. Our results on rate of convergence in the infinite-dimensional setting indicate that the methods of [Kiw95] should work well even when the dimension of the space becomes very large. Our work has benefited greatly from the general framework of [BaB96]; in fact, most of our results parallel ones obtained in [BaB96] for short-step methods, and, hence, we shorten some of our proofs by referring to [BaB96] (the extended version of our report [KiŁ94] provides more details). Yet another recent perspective on long-step methods of [Com95a, Com95b, Com97] follows a different path (cf. Remark 2.4).

The paper is organized as follows. In section 2 we describe a general method and its basic properties. Our general convergence results of sections 3 and 4 hinge on generalizations of the concepts of focusing and linearly focusing algorithms of [BaB96]. These concepts, as well as that of linear regularity, are exploited in section 5 in deriving linear rate of convergence results. In section 6 we discuss some examples of [BaB96]. In section 7 we establish convergence of subgradient algorithms for the case where each C_i is a lower-level set of some convex function. Extensions of the surrogate and scalarized subgradient cuts of [Kiw95, Oet75] are given in sections 8 and 9. In section 10 we specialize the preceding results to the case where each C_i is polyhedral. Finally, section 11 extends some results of [Com95a, Com95b, Com97] for infinitely many constraints.

We use the following notation. $B(x,r) = \{y : \|y-x\| \le r\}$ is the ball with center x and radius r, $H = \{x : \langle a, x\rangle \le b\}$ ($a \in \mathcal{H}$, $b \in \mathbb{R}$) is called a halfspace (including $H = \mathcal{H}$ or $\emptyset$), $d_C = \inf_{c\in C} \|\cdot - c\|$ is the *distance function* of a closed convex $C \subset \mathcal{H}$ (with $P_C = I$ and $d_C(\cdot) = \infty$ if $C = \emptyset$), $\operatorname{int} S$ and $\overline{\operatorname{co}}\, S$ are the *interior* and *closed convex hull* of $S \subset \mathcal{H}$, $\{1{:}N\} = \{1, 2, \ldots, N\}$, $\mathbb{R}^N_+ = \{\lambda \in \mathbb{R}^N : \lambda \ge 0\}$ and $t_+ = \max\{t, 0\}$ $\forall t \in \mathbb{R}$. We shall need the following results.

LEMMA 1.1 (see [BaB96, Lemma 2.4]). *If $T : D \to D$ is firmly nonexpansive and $c \in \operatorname{Fix} T$, then $\langle x - Tx, c\rangle \le \langle x - Tx, Tx\rangle$ $\forall x \in D$.*

COROLLARY 1.2 (see [BaB96, Corollary 2.5]). *If C is a closed convex set and $\alpha \ge 0$, then $R_{C,\alpha} = (1-\alpha)I + \alpha P_C$ satisfies $\|c - R_{C,\alpha}x\|^2 \le \|c - x\|^2 - \alpha(2-\alpha)\|x - P_C x\|^2$ $\forall c \in C$, $x \in \mathcal{H}$.*

2. A general algorithm and its basic properties.

Identifying (2) and (3), let $C_i = F_i$, $i \in I$, $N = |I|$ (so that $I = \{1{:}N\}$), and $C = \cap_{i\in I} C_i$. We first state a general algorithm for (2), without assuming that $C \ne \emptyset$.

ALGORITHM 2.1.

Step 0 (initiation). Select an initial point $x^0 \in D$, a weight threshold $\lambda_{\min} \in (0, \frac{1}{N}]$, and a starting localization radius $\rho_0 \ge d_C(x^0)$. Set the iteration counter to $n = 0$.

Step 1 (working set selection). Choose a nonempty set $\hat{I}^n \subset I$, so that for each $i \in I$, $i \in \hat{I}^n$ for infinitely many n.

Step 2 (halfspace selection). For each $i \in \hat{I}^n$, choose a firmly nonexpansive operator $T_i^{(n)} : D \to D$ with $\operatorname{Fix} T_i^{(n)} \supset C_i$, let $x^{in} = T_i^{(n)} x^n$ and $H_i^n = \{x :$

$\langle a^{in}, x\rangle \le b_{in}\}$ with

$$(a^{in}, b_{in}) = (x^n - x^{in}, \langle x^n - x^{in}, x^{in}\rangle) \quad \forall i \in \hat{I}^n. \tag{4}$$

Step 3 (surrogate construction). Find a *weight* vector $\lambda^n \in \mathbb{R}_+^{|I|}$, $\lambda_i^n = 0$, $i \notin \hat{I}^n$, $\sum_{i\in I} \lambda_i^n = 1$, for the *surrogate inequality* $\langle \hat{a}^n, x\rangle \le \hat{b}_n$ with

$$(\hat{a}^n, \hat{b}_n) = \sum_{i\in\hat{I}^n} \lambda_i^n (a^{in}, b_{in}) = \sum_{i\in\hat{I}^n} \lambda_i^n (x^n - x^{in}, \langle x^n - x^{in}, x^{in}\rangle) \tag{5}$$

such that the *surrogate halfspace* $\hat{H}^n = \{x : \langle \hat{a}^n, x\rangle \le \hat{b}_n\}$ satisfies

$$d_{\hat{H}^n}(x^n) \ge \lambda_{\min} \max_{i\in\hat{I}^n} d_{H_i^n}(x^n). \tag{6}$$

If $\hat{H}^n = \emptyset$, print "$C = \emptyset$" and stop.

Step 4 (relaxation). Select a relaxation parameter $\alpha_n \in (0, 2]$ and set (cf. Corollary 1.2)

$$x^{n+1} = R_{\hat{H}^n,\alpha_n} x^n = x^n + \alpha_n (P_{\hat{H}^n} x^n - x^n), \tag{7}$$

$$\sigma_n = \alpha_n(2 - \alpha_n) d^2_{\hat{H}^n}(x^n). \tag{8}$$

Step 5 (infeasibility detection). If $\rho_n^2 < \sigma_n$, print "$C = \emptyset$" and stop.

Step 6 (update locality radius). Set $\rho_{n+1} = (\rho_n^2 - \sigma_n)^{\frac{1}{2}}$, increase n by 1 and go to Step 1.

Define the set of *active indices* $I^n = \{i \in \hat{I}^n : \lambda_i^n > 0\}$. Suppose $x^n \in D$. At Step 2, by Lemma 1.1 with $T = T_i^{(n)}$, we have $C_i \subset \operatorname{Fix} T_i^{(n)} \subset H_i^n$ $\forall i \in \hat{I}^n$. By (4)–(5),

$$\langle a^{in}, x^n\rangle - b_{in} = \|a^{in}\|^2 = \|x^n - x^{in}\|^2 \quad \forall i \in \hat{I}^n, \tag{9}$$

$$\langle \hat{a}^n, x^n\rangle - \hat{b}_n = \sum_{i\in I^n} \lambda_i^n \|x^n - x^{in}\|^2 \quad \text{and} \quad \|\hat{a}^n\| = \|\sum_{i\in I^n} \lambda_i^n (x^n - x^{in})\|, \tag{10}$$

so $x^{in} = P_{H_i^n}(x^n)$, $d_{H_i^n}(x^n) = \|x^n - x^{in}\| = \|a^{in}\|$, $\forall i \in \hat{I}^n$. Since $\cap_{i\in I^n} C_i \subset \cap_{i\in I^n} H_i^n \subset \hat{H}^n$ from $\lambda^n \ge 0$, by Corollary 1.2 and (8),

$$\begin{aligned} \|c - x^{n+1}\|^2 &\le \|c - x^n\|^2 - \alpha_n(2 - \alpha_n) d^2_{\hat{H}^n}(x^n) \\ &= \|c - x^n\|^2 - \sigma_n \quad \forall c \in \cap_{i\in I^n} C_i. \end{aligned} \tag{11}$$

Thus progress toward the solution is measured by $d_{\hat{H}^n}(x^n)$, and we shall be interested in *deep* cuts that have $d_{\hat{H}^n}(x^n)$ as large as possible.

Remark 2.2. Inequality (6) holds if $\lambda_{\hat{\imath}}^n \ge \lambda_{\min}$ for some $\hat{\imath} \in I_{\max}^n := \operatorname{Arg\,max}_{i\in\hat{I}^n} \|x^n - x^{in}\|$, since

$$d_{\hat{H}^n}(x^n) = \frac{\langle \hat{a}^n, x^n\rangle - \hat{b}_n}{\|\hat{a}^n\|} \ge \lambda_{\min} \frac{\max_{i\in\hat{I}^n}(\langle a^{in}, x^n\rangle - b_{in})}{\max_{i\in\hat{I}^n} \|a^{in}\|} = \lambda_{\min} \max_{i\in\hat{I}^n} d_{H_i^n}(x^n)$$

from $\langle \hat{a}^n, x^n\rangle - \hat{b}_n \ge \lambda_{\min} \|a^{\hat{\imath}n}\|^2$ and $\|\hat{a}^n\| \le \sum_i \lambda_i^n \max_i \|a^{in}\|$ by the convexity of $\|\cdot\|$.

Remark 2.3. Other ways of finding cuts are given in [Kiw95, Example 3.1] and section 8; e.g., $\lambda_i^n = \|a^{in}\|^\gamma / \sum_j \|a^{jn}\|^\gamma$, $i \in \hat{I}^n$, are admissible if $\gamma \geq 0$, since $\max_{i \in I^n_{\max}} \lambda_i^n \geq 1/|\hat{I}^n|$. The *deepest* surrogate cut that maximizes $d_{\hat{H}^n}(x^n)$ is obtained for weights that solve

$$\max \left\{ \frac{\sum_{i\in\hat{I}^n} \lambda_i \|a^{in}\|^2}{\|\sum_{i\in\hat{I}^n} \lambda_i a^{in}\|} : \lambda_i \geq 0, i \in \hat{I}^n, \sum_{i\in\hat{I}^n} \lambda_i = 1 \right\} \tag{12}$$

(cf. (9)–(10)), in which case $P_{\hat{H}^n} x^n = P_{\cap_{i\in\hat{I}^n} H_i^n} x^n$. "Cheap" approximate solutions to (12) (that satisfy (6) with $\lambda_{\min} = 1$) are discussed in [Kiw95, Kiw96b, Kiw97].

Remark 2.4. Conditions like $\lambda_i^n \geq \lambda_{\min}$ for *all* $i \in I^n$ (cf. Remark 2.2) simplify the convergence analysis [Com97], but they need not hold for (approximate) solutions of (12).

Remark 2.5. Let $\hat{c}_n = (\langle \hat{a}^n, x^n\rangle - \hat{b}_n)/\|\hat{a}^n\|^2$. Then (cf. (5), (7)) $P_{\hat{H}^n}(x^n) - x^n = -\hat{c}_n \hat{a}^n$ and $x^{n+1} = x^n - \alpha_n \hat{c}_n \hat{a}^n$ if $\hat{a}^n \neq 0$, whereas the method of [BaB96] would produce

$$\tilde{x}^{n+1} = x^n + \alpha_n \left(\sum_{i\in I^n} \lambda_i^n x^{in} - x^n \right) = x^n - \alpha_n \hat{a}^n. \tag{13}$$

Both the *long-step* method (7) and the *short-step* method (13) use the same direction, but the former can take a much longer step when some tangent cone to $\cap_{i\in I^n} H_i^n$ is "flat" and $\hat{c}_n \gg 1$ [Kiw95, Lemma 4.3]. Of course, both steps coincide if $|\hat{I}^n| = 1$, so the expected improvements will crystalize for parallel methods.

Remark 2.6. The short-step method (13) is replaced in [BaB96] by $\tilde{x}^{n+1} = x^n + \sum_{i\in I^n} \tilde{\lambda}_i^n \alpha_i^{(n)} (x^{in} - x^n)$ with stepsizes $\alpha_i^{(n)} \in [0, 2]$ and weights $\tilde{\lambda}_i^n \geq 0$, $\sum_{i\in I} \tilde{\lambda}_i^n = 1$. This iteration is *equivalent* to (13) with $\alpha_n = \sum_{i\in I} \tilde{\lambda}_i^n \alpha_i^{(n)} \in [0, 2]$ and $\lambda_i^n = \tilde{\lambda}_i^n \alpha_i^{(n)} / \alpha_n$.

By (11), the sequence $\{x^k\}$ is Fejér monotone with respect to $C = \cap_i C_i$

$$\|c - x^{n+1}\|^2 \leq \|c - x^n\|^2 - \sigma_n \leq \|c - x^0\|^2 - \sum_{j=0}^{n} \sigma_j \leq \|c - x^0\|^2 \quad \forall c \in C, \forall n. \tag{14}$$

Using $\rho_n^2 = \rho_0^2 - \sum_{j=0}^{n-1} \sigma_j$, one may prove the following lemma as in [Kiw95], so we omit its proof. Lemma 2.7 can be used to detect $C = \emptyset$.

LEMMA 2.7 (the nested ball principle). *If* $(\rho_0 - \|x^{n+1} - x^0\|)^2 > \rho_n^2 - \sigma_n$ *then* $C = \emptyset$.

3. Basic convergence results. From now on, unless stated otherwise, we *assume that* $C \neq \emptyset$ *and* $\{x^n\} \subset D$. The second condition is assumed *implicitly* in [BaB96]; it holds, e.g., if $D = \mathcal{H}$ or $B(c, \|c - x^0\|) \subset D$ for some $c \in C$ (cf. (14)). Let $\underline{\alpha}_\infty = \underline{\lim}_n \alpha_n$ and $\overline{\alpha}_\infty = \overline{\lim}_n \alpha_n$.

LEMMA 3.1.

(i) $\|c - x^n\|^2 - \|c - x^{n+1}\|^2 \geq \alpha_n(2 - \alpha_n) d^2_{\hat{H}^n}(x^n)$ *if* $c \in \cap_{i\in I^n} C_i$.

(ii) $\|c - x^n\|^2 - \|c - x^m\|^2 \geq \sum_{l=n}^{m-1} \alpha_l(2 - \alpha_l) d^2_{\hat{H}^l}(x^l)$ *if* $c \in \cap_{l=n}^{m-1} \cap_{i\in I^l} C_i$, $m \geq n \geq 0$.

(iii) $\{x^n\}$ *is Fejér monotone with respect to* C (*hence bounded*) *and*

$$\sum_{l=0}^{\infty} \alpha_l (2 - \alpha_l) d^2_{\hat{H}^l}(x^l) < \infty.$$

Proof. (i) repeats (11). (ii)–(iii) follow from (i). □

COROLLARY 3.2.

(i) *If* $\operatorname{int} C \neq \emptyset$*, then* $\{x^n\}$ *converges in norm to some* $x \in D$.

(ii) *If* $\underline{\lim}_n d_C(x^n) \to 0$*, then* $\{x^n\}$ *converges in norm to some* $x \in C$.

(iii) $\{x^n\}$ *has at most one weak cluster point in* C.

Proof. This follows from Lemma 3.1(iii) and [BaB96, Theorem 2.16]. □

COROLLARY 3.3. *If* $\operatorname{int} C \neq \emptyset$*, then* $\sum_n \|x^{n+1} - x^n\| = \sum_n \alpha_n d_{\hat{H}^n}(x^n) < \infty$.

Proof. Fix $c \in \operatorname{int} C$ and get $\epsilon > 0$ such that (s.t.) $B(c, \epsilon) \subset C$. Suppose $x^{n+1} \neq x^n$. Let $y = (x^n - x^{n+1})/\|x^n - x^{n+1}\|$, so that $c + \epsilon y \in C$ and, by Fejér monotonicity, $\|c + \epsilon y - x^{n+1}\|^2 \leq \|c + \epsilon y - x^n\|^2$. Expanding yields $2\epsilon\|x^{n+1} - x^n\| \leq \|c - x^n\|^2 - \|c - x^{n+1}\|^2$, so $\sum_n \|x^{n+1} - x^n\| < \infty$, with $\|x^{n+1} - x^n\| = \alpha_n d_{\hat{H}^n}(x^n)$ by (7). □

COROLLARY 3.4. *If* $\overline{\alpha}_\infty < 2$*, then the algorithm is* regular*, i.e.,* $\|x^{n+1} - x^n\| \to 0$.

Proof. By Lemma 3.1(iii), $\sum_n \alpha_n d^2_{\hat{H}^n}(x^n) < \infty$, so, since

$$\|x^{n+1} - x^n\| = \alpha_n d_{\hat{H}^n}(x^n)$$

(cf. (7)) and $\alpha_n \leq 2$, $\sum_n \|x^{n+1} - x^n\|^2 < \infty$. □

COROLLARY 3.5. *If* $0 < \underline{\alpha}_\infty, \overline{\alpha}_\infty < 2$*, then* $d_{\hat{H}^n}(x^n) \to 0$ *and* $\lim_{n: i \in \hat{I}^n} d_{H_i^n}(x^n) = 0$ $\forall i \in I$.

Proof. This follows from Lemma 3.1(iii) and (6). □

As in [BaB96], from now on we assume that the algorithm is *focusing.*

DEFINITION 3.6. *We say the algorithm is* focusing *if for each* $i \in I$ *and subsequence* $\{x^{n_k}\}$*, the conditions* $x^{n_k} \rightharpoonup x$, $d_{H_i^{n_k}}(x^{n_k}) \to 0$ (*i.e.,* $x^{n_k} - T_i^{(n_k)} x^{n_k} \to 0$) *and* $i \in \hat{I}^{n_k}$ $\forall k$ *imply* $x \in C_i$*, where* $\to$ *and* $\rightharpoonup$ *stand for norm and weak convergence respectively.*

FACT 3.7 (see [BaB96, Proposition 3.16]). *Suppose for each* $i \in I$, $\{T_i^{(n)}\}$ converges actively pointwise *to* T_i*, i.e.,* $\lim_{n: i \in \hat{I}^n} T_i^{(n)} x = T_i x$ *for every* $x \in D$*. Then the algorithm is focusing.*

THEOREM 3.8. *If* $0 < \underline{\alpha}_\infty, \overline{\alpha}_\infty < 2$*, then* $\{x^n\}$ *either converges in norm to some point in* C *or has no norm cluster points at all.*

Proof. Use Corollary 3.5 in the proof of [BaB96, Theorem 3.10]. □

Remark 3.9. If there exists $\epsilon > 0$ s.t. $\{\alpha_n\} \subset [\epsilon, 2 - \epsilon]$, then $0 < \underline{\alpha}_\infty, \overline{\alpha}_\infty < 2$.

DEFINITION 3.10. *We say the algorithm is* intermittent *or* p-intermittent *if there is a positive integer* p *s.t.* $i \in \hat{I}^n \cup \hat{I}^{n+1} \cup \cdots \cup \hat{I}^{n+p-1}$ *for each* $i \in I$ *and all* n.

THEOREM 3.11.

(i) *Suppose the algorithm is intermittent and* $0 < \underline{\alpha}_\infty, \overline{\alpha}_\infty < 2$*. Then* $\{x^n\}$ *is regular and converges weakly to some point in* C.

(ii) *Suppose the algorithm is* p*-intermittent, and* $\hat{\nu}_n = \min\{\alpha_l(2 - \alpha_l) : np \leq l < (n+1)p\}$ $\forall n \geq 0$*. If* $\sum_n \hat{\nu}_n = \infty$*, then* $\{x^n\}$ *has a unique weak cluster point in* C *and a subsequence* $x^{n_k p} \rightharpoonup x \in C$ *s.t.*

$$\sum_{l=n_k p}^{(n_k+1)p-1} \left\{ \max_{i \in \hat{I}^l} d^2_{H_i^l}(x^l) + \|x^{l+1} - x^l\|^2 \right\} \to 0. \tag{15}$$

(iii) *Suppose* $\{x^n\}$ *converges weakly to some point* x*. If* $\sum_{n: i \in \hat{I}^n} \alpha_n(2 - \alpha_n) = \infty$ *for some* $i \in I$*, then* $x \in C_i$*. Consequently,* $x \in C$ *if* $\sum_{n: i \in \hat{I}^n} \alpha_n(2 - \alpha_n) = \infty$ *for* every $i \in I$.

Proof. (i) Use Corollary 3.4 and Corollary 3.5 in the proof of [BaB96, Theorem 3.20(i)]. (ii) Let $c \in C$. By Lemma 3.1(ii), (6) and the definition of $\hat{\nu}_n$, for all $n \geq 0$,

$$\|x^{np} - c\|^2 - \|x^{(n+1)p} - c\|^2 \geq \hat{\nu}_n \sum_{l=np}^{(n+1)p-1} d^2_{\hat{H}^l}(x^l) \geq \hat{\nu}_n \sum_{l=np}^{(n+1)p-1} \lambda^2_{\min} \max_{i \in \hat{I}^l} d^2_{H^l_i}(x^l),$$

where $2d_{\hat{H}^l}(x^l) \geq \|x^{l+1} - x^l\|$ by (7). The conclusion follows as in the proofs of [BaB96, Theorem 3.20(ii)] and Theorem 3.15. (iii) $\sum_{n:i \in \hat{I}^n} \alpha_n(2-\alpha_n)\lambda^2_{\min} d^2_{H^n_i}(x^n) < \infty$ (cf. Lemma 3.1(iii) and (6)) yields $\underline{\lim}_{n:i\in\hat{I}^n} d_{H^n_i}(x^n) = 0$, so $x \in C_i$ because the algorithm is focusing. □

Remark 3.12. If $0 < \underline{\alpha}_\infty, \overline{\alpha}_\infty < 2$, then $\sum_{n:i\in\hat{I}^n} \alpha_n(2-\alpha_n) = \infty$ for all i.

THEOREM 3.13.

(i) *If* $\sum_{n:\hat{I}^n = I} \alpha_n(2-\alpha_n) = \infty$, *then* $\{x^n\}$ *has a unique cluster point* x *in* C *and a subsequence* $\{x^{n_k}\}$ *s.t.* $x^{n_k} \rightharpoonup x$ *and* $\max_{i\in I} d_{H^{n_k}_i}(x^{n_k}) \to 0$.

(ii) *If there exist* $\epsilon > 0$ *and a subsequence* $\{x^{n_k}\}$ *s.t.* $\epsilon \leq \alpha_{n_k} \leq 2 - \epsilon$ *and* $\hat{I}^{n_k} = I$ *for all* k, *then* $x^{n_k} \rightharpoonup x$ *for some* $x \in C$ *and* $\max_{i\in I} d_{H^{n_k}_i}(x^{n_k}) \to 0$.

Proof. (i) By Lemma 3.1(iii), $\sum_n \alpha_n(2-\alpha_n) d^2_{\hat{H}^n}(x^n) < \infty$, so

$$\underline{\lim}_{n} \max_{i \in I} d_{H^n_i}(x^n) = 0$$

by (6). Thus we can extract a subsequence $\{x^{n_k}\}$ and fix x s.t. $\max_{i\in I} d_{H^{n_k}_i}(x^{n_k}) \to 0$ and $x^{n_k} \rightharpoonup x$. Since the algorithm is focusing, $x \in C$ (cf. Definition 3.6). By Corollary 3.2(iii), $\{x^n\}$ has at most one weak cluster point in C, so (i) holds. (ii) is proved similarly. □

DEFINITION 3.14. *The algorithm is* quasi cyclic *if there is an increasing sequence of integers* $\{\tau_k\}_{k=0}^\infty$ *s.t.* $\tau_0 = 0$, $\sum_{k=0}^\infty (\tau_{k+1} - \tau_k)^{-1} = \infty$ *and* $I = \bigcup_{l=\tau_k}^{\tau_{k+1}-1} \hat{I}^l$ $\forall k$.

THEOREM 3.15. *Suppose the algorithm is quasi cyclic and* $0 < \underline{\alpha}_\infty, \overline{\alpha}_\infty < 2$. *Then* $\{x^n\}$ *has a unique weak cluster point in* C *and a subsequence* $x^{\tau_{k'}} \rightharpoonup x \in C$ *s.t.*

$$\sum_{l=\tau_{k'}}^{\tau_{k'+1}-1} \left\{ \max_{i\in\hat{I}^l} d_{H^l_i}(x^l) + \|x^{l+1} - x^l\| \right\} \to 0. \tag{16}$$

Proof. Fix $c \in C$ and $\bar{k}$ s.t. $\alpha_l(2-\alpha_l) \geq \epsilon > 0$ $\forall l \geq \bar{k}$. By Lemma 3.1(ii), $\forall k \geq \bar{k}$,

$$\|x^{\tau_k} - c\|^2 - \|x^{\tau_{k+1}} - c\|^2 \geq \epsilon \sum_{l=\tau_k}^{\tau_{k+1}-1} d^2_{\hat{H}^l}(x^l) \geq \epsilon(\tau_{k+1} - \tau_k)^{-1} \sum_{l=\tau_k}^{\tau_{k+1}-1} d_{\hat{H}^l}(x^l),$$

using the Cauchy–Schwarz inequality. Summing and invoking Definition 3.14 yields the existence of a subsequence $\{x^{\tau_{k'}}\}$ s.t. $\sum_{l=\tau_{k'}}^{\tau_{k'+1}-1} d_{\hat{H}^l}(x^l) \to 0$. Then (16) follows from (6) and the fact that $2d_{\hat{H}^l}(x^l) \geq \|x^{l+1} - x^l\|$ (cf. (7)). Extracting a subsequence if necessary, assume $x^{\tau_{k'}} \rightharpoonup x \in D$. To see that $x \in C$, for any $i \in I$, pick $n_{k'} \in \{\tau_{k'} : \tau_{k'+1} - 1\}$ s.t. $i \in \hat{I}^{n_{k'}}$ to get $x \in C_i$, since the algorithm is focusing, $d_{H^{n_{k'}}_i}(x^{n_{k'}}) \to 0$ and $x^{n_{k'}} \rightharpoonup x$ by (16). By Corollary 3.2(iii), x is the unique weak cluster point of $\{x^n\}$ in C. □

4. Projection setting. At Step 2 we have $x^{in} = T_i^{(n)} x^n = P_{H_i^n} x^n$ with $H_i^n \supset C_i$ for every index i and all n. Hence, as in the setting of [BaB96, section 4], one might assume that $T_i^{(n)}$ is the projection onto some closed convex set C_i^n containing C_i: $T_i^{(n)} = P_{C_i^n}$ and $C_i^n \supset C_i$ for all i and n.

Definition 4.1. *Let $\hat{d}_i(x) = \|x - T_i x\|$ be the* defect *function of T_i, $i \in I$, so that $\hat{d}_i(x) = 0$ iff $x \in C_i$. We say the algorithm is* linearly focusing *if there is $\beta > 0$ s.t. $\beta \hat{d}_i(x^n) \leq d_{H_i^n}(x^n)$ for all large n and every $i \in \hat{I}^n$; it is* strongly focusing *if for every index i and every subsequence $\{x^{n_k}\}$, the conditions $x^{n_k} \rightharpoonup x$, $d_{H_i^{n_k}}(x^{n_k}) \to 0$ and $i \in \hat{I}^{n_k}$ imply $\hat{d}_i(x^{n_k}) \to 0$, and hence $x \in C_i$ by the demiclosedness principle* [Opi67, Lemma 2]. *Thus* (*cf. Definition* 3.6)*: linearly focusing ⇒ strongly focusing ⇒ focusing.*

Remark 4.2. For each $i \in I$, $\hat{d}_i$ is nonexpansive (so is $\bar{I} - T_i$ from $\|(x - T_i x) - (y - T_i y)\|^2 \leq \|x - y\|^2 - \|T_i x - T_i y\|^2$; cf. [Eck89, section 3.2.4]).

Corollary 4.3. *If the algorithm* uses constant operators, *i.e., $T_i^{(n)} = T_i$ for all $n \geq 0$ and $i \in I$, then the algorithm is linearly focusing.*

Corollary 4.4. *If the set $S = \{x^n : n \geq 0\}$ is relatively compact, then the algorithm is strongly focusing. In particular, this holds whenever $\mathcal{H}$ is finite-dimensional or* $\operatorname{int} C \neq \emptyset$.

Proof. Use continuity of T_i in the proof of [BaB96, Corollary 4.12]. □

Corollary 4.5. *Suppose the algorithm is linearly focusing,* $0 < \underline{\alpha}_\infty, \overline{\alpha}_\infty < 2$ *and either $\mathcal{H}$ is finite dimensional or* $\operatorname{int} C \neq \emptyset$. *Then $\{x^n\}$ converges in norm to some point in C.*

Proof. If $\operatorname{int} C \neq \emptyset$, then $\{x^n\}$ converges in norm (Corollary 3.2(i)). If $\dim \mathcal{H} < \infty$, then $\{x^n\}$ has a norm cluster point. The result follows from Theorem 3.8. □

Theorem 4.6. *Suppose* $\operatorname{int} C \neq \emptyset$, *so that $\{x^n\}$ converges to some x* (*Corollary* 3.2(i)). *If $\sum_{n:i \in \hat{I}^n} \alpha_n = \infty$ for some $i \in I$, then $x \in C_i$. Thus, $x \in C$ if $\sum_{n:i \in \hat{I}^n} \alpha_n = \infty$ for all $i \in I$.*

Proof. Since $\sum_{n:i \in \hat{I}^n} \alpha_n d_{\hat{H}^n}(x^n) < \infty$ by Corollary 3.3 with

$$d_{\hat{H}^n}(x^n) \geq \lambda_{\min} d_{H_i^n}(x^n)$$

if $i \in \hat{I}^n$ by (6), we have $\underline{\lim}_n d_{H_i^n}(x^n) = 0$. Since the algorithm is focusing, $x \in C_i$. □

Corollary 4.7. *Suppose $\{x^n\}$ has a subsequence $\{x^{n'}\}$ s.t. $\underline{\lim}_{n'} \alpha_{n'} > 0$ and $\hat{I}^{n'} = I$ for all n'. If* $\operatorname{int} C \neq \emptyset$, *then $\{x^n\}$ converges in norm to some point in C.*

Remark 4.8. Note that the last theorem works especially when $\alpha_n \equiv 2$, in which case none of the previous results are applicable. The result of [BaB96, Theorem 4.22] corresponding to Corollary 4.7 assumes additionally that $\lambda_i^{n'} \to \lambda_i \in (0, 1]$ for $i \in I$.

Definition 4.9. *We say the algorithm* considers remotest sets *if $\hat{I}^n$ contains some* active remotest index $i_n \in \hat{I}^n_{\mathrm{rem}} := \{i : \hat{d}_i(x^n) = \max_{j \in I} \hat{d}_j(x^n)\}$ *for all n.*

Theorem 4.10. *Suppose the algorithm is strongly focusing and considers remotest sets.*

(i) *If $\sum_n \alpha_n(2 - \alpha_n) = \infty$, then $\{x^n\}$ has a unique cluster point in C and a subsequence $x^{n_k} \rightharpoonup x \in C$ s.t.* $\max_{i \in I} \hat{d}_i(x^{n_k}) \to 0$.

(ii) *If* $0 < \underline{\alpha}_\infty, \overline{\alpha}_\infty < 2$, *then $x^n \rightharpoonup x$ for some $x \in C$ and* $\max_{i \in I} \hat{d}_i(x^n) \to 0$.

Proof. (i) By Lemma 3.1(iii), $\sum_n \alpha_n(2 - \alpha_n) d^2_{\hat{H}^n}(x^n) < \infty$, so $\underline{\lim}_n d_{H_{i_n}^n}(x^n) = 0$ by (6). Thus we can extract a subsequence $\{x^{n_k}\}$ and fix x and $i \in I$ s.t. $d_{H_i^{n_k}}(x^{n_k}) \to 0$, $i_{n_k} \equiv i$ and $x^{n_k} \rightharpoonup x$. Since the algorithm considers remotest sets and is strongly focusing, we deduce $\max_{j\in I}\hat{d}_j(x^{n_k}) = \hat{d}_i(x^{n_k}) \to 0$ and $x \in C$ (cf. Definition 4.1). By Corollary 3.2(iii), $\{x^n\}$ has at most one weak cluster point in C, so (i) holds. (ii) is proved similarly. □

Remark 4.11. Theorem 4.10 also holds for the *approximate remotest set control* which demands that

$$\max_{i\in I}\hat{d}_i(x^{n_k}) \to 0 \qquad \text{whenever} \quad x^{n_k} \rightharpoonup x \quad \text{and} \quad \max_{i\in \hat{I}^{n_k}} d_{H_i^{n_k}}(x^{n_k}) \to 0.$$

This control was used in [GPR67, Kiw95] with $\hat{d}_j$ replaced by d_{C_j}.

5. Norm convergence and bounded regularity. Guaranteeing norm convergence requires further assumptions.

DEFINITION 5.1. $\{T_i\}_{i\in I}$ *is* regular *if* $\forall \epsilon > 0\ \exists \delta > 0\ \forall x \in \mathcal{H} : \max_{i\in I}\hat{d}_i(x) \le \delta \Rightarrow d_C(x) \le \epsilon$, *and* boundedly regular *if* $\forall$ *bounded* $S \subset \mathcal{H}\ \forall \epsilon > 0\ \exists \delta > 0\ \forall x \in S : \max_{i\in I}\hat{d}_i(x) \le \delta \Rightarrow d_C(x) \le \epsilon$. $\{C_i\}_{i\in I}$ *is* (boundedly) regular *if* $\{T_i = P_{C_i}\}_{i\in I}$ (*then* $\hat{d}_i = d_{C_i}\ \forall i \in I$) *is also.*

THEOREM 5.2. *Suppose the algorithm is strongly focusing and p-intermittent, and* $\hat{\nu}_n = \min\{\alpha_l(2-\alpha_l) : np \le l < (n+1)p\}\ \forall n \ge 0$. *If* $\sum_n \hat{\nu}_n = \infty$ *and* $\{T_i\}_{i\in I}$ *is boundedly regular, then* $\{x^n\}$ *converges in norm to some point in* C.

Proof. Use Theorem 3.11(ii) and the proofs of [BaB96, Theorem 5.2] or Theorem 5.5. □

THEOREM 5.3. *Suppose the algorithm is strongly focusing and considers remotest sets, and* $\{T_i\}_{i\in I}$ *is boundedly regular. If* $\sum_n \alpha_n(2-\alpha_n) = \infty$, *then* $\{x^n\}$ *converges in norm to some point in* C. *In particular, this happens whenever* $0 < \underline{\alpha}_\infty, \overline{\alpha}_\infty < 2$.

Proof. Use Theorem 4.10(i) and the proof of [BaB96, Theorem 5.3]. □

THEOREM 5.4. *Suppose the algorithm is strongly focusing and* $\{T_i\}_{i\in I}$ *is boundedly regular. If* $\sum_{n:\hat{I}^n = I}\alpha_n(2-\alpha_n) = \infty$, *then* $\{x^n\}$ *converges in norm to some point in* C.

Proof. By Theorem 3.13(i), $x^{n_k} \rightharpoonup x$ and $\max_{i\in I} d_{H_i^{n_k}}(x^{n_k}) \to 0$ imply that $\max_{i\in I}\hat{d}_i(x^{n_k}) \to 0$ (strong focusing), so $d_C(x^{n_k}) \to 0$ by bounded regularity; apply Corollary 3.2(ii). □

THEOREM 5.5. *Suppose the algorithm is strongly focusing and quasi cyclic, and* $0 < \underline{\alpha}_\infty, \overline{\alpha}_\infty < 2$. *If* $\{T_i\}_{i\in I}$ *is boundedly regular, then* $x^n \to x$ *for some* $x \in C$.

Proof. Using Definition 3.14 and (16) with $x^{\tau_{k'}} \rightharpoonup x \in C$, for any $i \in I$ pick $n_{k'} \in \{\tau_{k'} : \tau_{k'+1} - 1\}$ s.t. $i \in \hat{I}^{n_{k'}}$, $d_{H_i^{n_{k'}}}(x^{n_{k'}}) \to 0$, $x^{n_{k'}} - x^{\tau_{k'}} \to 0$, and $x^{n_{k'}} \rightharpoonup x$. As the algorithm is strongly focusing, $\hat{d}_i(x^{n_{k'}}) \to 0$. But $\hat{d}_i$ is nonexpansive (Remark 4.2) and $x^{n_{k'}} - x^{\tau_{k'}} \to 0$, so $\hat{d}_i(x^{\tau_{k'}}) \to 0$. Since i is arbitrary, $\max_{i\in I}\hat{d}_i(x^{\tau_{k'}}) \to 0$. Now $\{T_i\}_{i\in I}$ is boundedly regular and $\{x^n\}$ is bounded, so $d_C(x^{\tau_{k'}}) \to 0$. Hence (Corollary 3.2(ii)), $x^n \to x$. □

As in [BaB96], guaranteeing *linear* convergence requires *linear* regularity.

DEFINITION 5.6. $\{T_i\}_{i\in I}$ *is* linearly regular *if* $\exists \kappa > 0\ \forall x \in \mathcal{H}$: $d_C(x) \le \kappa \max_{i\in I}\hat{d}_i(x)$, *and* boundedly linearly regular *if* $\forall$ *bounded* $S \subset \mathcal{H}\ \exists \kappa_S > 0\ \forall x \in S$: $d_C(x) \le \kappa_S \max_{i\in I}\hat{d}_i(x)$. $\{C_i\}_{i\in I}$ *is* (boundedly) linearly regular *if* $\{T_i = P_{C_i}\}_{i\in I}$ *(then* $\hat{d}_i = d_{C_i}\ \forall i \in I$*) is also.*

We say $\{x^n\}$ *converges linearly with rate* $\beta \in [0,1)$ if there are $x \in \mathcal{H}$ and $\alpha \ge 0$ s.t. $\|x^n - x\| \le \alpha\beta^n$ for all n.

THEOREM 5.7. *Suppose the algorithm is linearly focusing and p-intermittent,* $\{\alpha_n\} \subset [\epsilon, 2-\epsilon]$ *for some* $\epsilon > 0$, *and* $\{T_i\}_{i\in I}$ *is boundedly linearly regular. Then* $\{x^n\}$

converges linearly to some point in C; the rate of convergence is independent of the starting point whenever $\{T_i\}_{i\in I}$ is linearly regular.

Proof. Fix any $i \in I$. For all $k \geq 0$, get $m_k \in \{kp\colon (k+1)p-1\}$ with $i \in \hat{I}^{m_k}$. Since $\hat{d}_i$ is nonexpansive (Remark 4.2),

$$\hat{d}_i(x^{kp}) \leq \|x^{kp} - x^{m_k}\| + \hat{d}_i(x^{m_k}) \leq \sum_{l=kp}^{m_k-1} \|x^{l+1} - x^l\| + \hat{d}_i(x^{m_k}),$$

so

$$\hat{d}_i^2(x^{kp}) \leq (m_k + 1 - kp)\left(\sum_{l=kp}^{m_k-1} \|x^{l+1} - x^l\|^2 + \hat{d}_i^2(x^{m_k})\right)$$

by the Cauchy–Schwarz inequality. Fix $c \in C$. Since $\min_{\alpha\in(0,2-\epsilon]}(2-\alpha)/\alpha > \epsilon/2$, Lemma 3.1(i) with $\alpha_n \leq 2-\epsilon$ and $\|x^{n+1} - x^n\| = \alpha_n d_{\hat{H}^n}(x^n)$ (cf. (7)) yield

$$\|x^n - c\|^2 - \|x^{n+1} - c\|^2 \geq \frac{2-\alpha_n}{\alpha_n}\|x^{n+1} - x^n\|^2 \geq \frac{\epsilon}{2}\|x^{n+1} - x^n\|^2,$$

so

$$\frac{\epsilon}{2}\sum_{l=kp}^{m_k-1} \|x^{l+1} - x^l\|^2 \leq \|x^{kp} - c\|^2 - \|x^{m_k} - c\|^2 \leq \|x^{kp} - c\|^2 - \|x^{(k+1)p} - c\|^2.$$

Since the algorithm is linearly focusing, there is $\beta > 0$ s.t. $\beta\hat{d}_j(x^n) \leq d_{H_j^n}(x^n)$ for all $j \in \hat{I}^n$ and large n, whereas $\max_{j\in\hat{I}^n} d_{H_j^n}(x^n) \leq d_{\hat{H}^n}(x^n)/\lambda_{\min}$ by (6), and $\epsilon^2 d_{\hat{H}^n}^2(x^n) \leq \|x^n - c\|^2 - \|x^{n+1} - c\|^2$ from (11) with $\alpha_n(2-\alpha_n) \geq \epsilon^2$. Hence, $\beta^2\epsilon^2\lambda_{\min}^2\hat{d}_i^2(x^{m_k}) \leq \|x^{m_k} - c\|^2 - \|x^{m_k+1} - c\|^2 \leq \|x^{kp} - c\|^2 - \|x^{(k+1)p} - c\|^2$. Thus,

$$\hat{d}_i^2(x^{kp}) \leq p\left(\frac{2}{\epsilon} + \frac{1}{\beta^2\epsilon^2\lambda_{\min}^2}\right)\left(\|x^{kp} - c\|^2 - \|x^{(k+1)p} - c\|^2\right). \tag{17}$$

The conclusion follows as in the proof of [BaB96, Theorem 5.7]. □

THEOREM 5.8. *Suppose the algorithm is linearly focusing, considers remotest sets, there is $\epsilon > 0$ s.t. $\epsilon \leq \alpha_n \leq 2-\epsilon$ for all large n, and $\{T_i\}_{i\in I}$ is boundedly linearly regular. Then $\{x^n\}$ converges linearly to some point in C; the rate of convergence is independent of the starting point whenever $\{T_i\}_{i\in I}$ is linearly regular.*

Proof. Use (17) with $p = 1$, $i = i_n$ (omit $\frac{2}{\epsilon}$) as in [BaB96, Proof of Theorem 5.8]. □

Remark 5.9. As in [BaB96], one may use $T_i = P_{C_i}$ $\forall i \in I$ in the preceding results.

6. Examples.

Example 6.1. Suppose the algorithm is linearly focusing, $0 < \underline{\alpha}_\infty, \overline{\alpha}_\infty < 2$, and some T_j is demicompact [Com95a]; e.g., its range is boundedly compact (e.g., so is C_j and $T_j = P_{C_j}$). Then $\{x^n\}$ converges in norm to some point in C.

Proof. By Corollary 3.5, $\lim_{n:j\in\hat{I}^n} d_{H_j^n}(x^n) = 0$. Since the algorithm is linearly focusing, it has a subsequence $\{x^{n'}\}$ with $j \in \hat{I}^{n'}$ and $\|x^{n'} - T_j x^{n'}\| = \hat{d}_j(x^{n'}) \to 0$. Passing to a subsequence if necessary, we can assume that $\{T_j x^{n'}\}$ is norm convergent. Hence, $\{x^{n'}\}$ has a norm cluster point. The result follows from Theorem 3.8. □

Example 6.2. Our framework covers the examples of [BaB96, Examples 6.1, 6.3, 6.5, 6.6, 6.8, 6.11, 6.15, 6.16, 6.17, 6.18, 6.21, 6.22, 6.24, 6.35, 6.43, and 6.44].

Example 6.3. Our framework provides "long-step" versions (cf. Remark 2.5) of the examples of [BaB96, Examples 6.13, 6.20, 6.27, 6.34, 6.37, 6.39, 6.40, 6.42, and 6.50].

Remark 6.4. Our framework does not cover Examples 6.30 and 6.47 of [BaB96] because Theorems 6.29 and 6.46 of [BaB96] hinge on a certain regularity property of short-step methods.

7. Subgradient algorithms. Throughout this section and sections 8 and 9 we make the following standing assumption.

Assumption 7.1. $C_i = \{x : f_i(x) \le 0\}$ for every $i \in I$ $(= \{1\colon N\})$, where $f_i : \mathcal{H} \to \mathbb{R}$ is a convex function whose *subdifferential* ∂f_i, defined by

$$\partial f_i(\hat{x}) = \{g \in \mathcal{H} : f_i(x) \ge f_i(\hat{x}) + \langle g, x - \hat{x}\rangle \ \ \forall x \in \mathcal{H}\} \quad \forall \hat{x} \in \mathcal{H},$$

is nonempty and uniformly bounded on bounded sets, so that f_i is bounded and Lipschitz continuous on bounded sets (cf. [BaB96, Proposition 7.8]).

DEFINITION 7.2. *The algorithm is called a* subgradient algorithm *if, for all* n *and* $i \in \hat{I}^n$, *Step* 2 *chooses* $T_i^{(n)} = P_{\check{H}_i^n}$, *where* $\check{H}_i^n = \{x : f_i(x^n) + \langle g^{in}, x - x^n\rangle \le 0\}$ *for some* $g^{in} \in \partial f_i(x^n)$, *so that* $P_{\check{H}_i^n}(x^n) = x^n - \frac{f_i^+(x^n)}{\|g^{in}\|^2} g^{in}$, *where* $f_i^+ = \max\{f_i, 0\}$ *and* $\frac{0}{0} := 0$.

THEOREM 7.3. *Let* $\hat{c} \in C$ *and* $L_i = \sup\{\|g\| : g \in \partial f_i(x), x \in B(\hat{c}, \|\hat{c} - x^0\|)\}$, $i \in I$. *Then* $\{x^n\} \subset B(\hat{c}, \|\hat{c} - x^0\|)$ *and* $\|g^{in}\| \le L_i$ *for all* $i \in \hat{I}^n$ *and* $n \ge 0$. *Further,*

(i) *the subgradient algorithm is focusing for* $T_i = P_{C_i}$, $i \in I$ (*cf. Definition* 3.6); *and*

(ii) *if there is some* Slater point $\check{x}$ *s.t.* $\sup_{i \in I} f_i(\check{x}) < 0$, *then*

$$d_{C_i}(x^n) \le \frac{\|\check{x} - x^0\|}{-f_i(\check{x})} f_i^+(x^n) \le \frac{L_i \|\check{x} - x^0\|}{-f_i(\check{x})} d_{\check{H}_i^n}(x^n)$$

for all $i \in \hat{I}^n$ *and* $n \ge 0$, *where* L_i *corresponds to* $\hat{c} = \check{x}$. *Thus, the algorithm is linearly focusing with* $\beta = \inf_{i \in I} \frac{-f_i(\check{x})}{L_i \|\check{x} - x^0\|}$ (*cf. Definition* 4.1).

Proof. Use the proofs of [BaB96, Theorems 7.7 and 7.12]. □

We only give two translations of the preceding convergence results.

THEOREM 7.4.

(i) *Suppose the subgradient algorithm is intermittent, and* $0 < \underline{\alpha}_\infty$ *and* $\overline{\alpha}_\infty < 2$. *Then* $\{x^n\}$ *is regular and converges weakly to some point in* C.

(ii) *Suppose the subgradient algorithm is* p*-intermittent and* $\hat{\nu}_n = \min\{\alpha_l(2 - \alpha_l) : np \le l < (n+1)p\}$ $\forall n \ge 0$. *If* $\sum_n \hat{\nu}_n = \infty$, *then* $\{x^n\}$ *has a unique weak cluster point in* C.

Proof. Combine Theorems 3.11 and 7.3. □

THEOREM 7.5. *Suppose for some* $\check{x} \in \mathcal{H}$ *and* $\check{\epsilon} > 0$, $\sup_{i \in I} f_i(\check{x}) \le -\check{\epsilon}$. *Then* $\check{x} \in \operatorname{int} C$, $\{C_i\}_{i \in I}$ *is boundedly linearly regular and* $\{x^n\}$ *converges in norm to some* $x \in \mathcal{H}$.

(i) *If* $\sum_{n : i \in \hat{I}^n} \alpha_n = \infty$ *for every index* i, *then* $x \in C$.

(ii) *If the algorithm is intermittent or considers remotest sets, and* $\{\alpha_n\} \subset [\epsilon, 2 - \epsilon]$ *for some* $\epsilon > 0$, *then* $x \in C$ *and* $\{x^n\}$ converges linearly *to* x.

Proof. If $x \in B(\check{x}, \inf_i\{\frac{\check{\epsilon}}{L_i}, \|\check{x} - x^0\|\})$, $g \in \partial f_i(x)$, then $f_i(\check{x}) \ge f_i(x) + \langle g, \check{x} - x\rangle \ge f_i(x) - L_i \|\check{x} - x\|$ yields $f_i(x) \le 0$, so $\check{x} \in \operatorname{int} C$ and $\{C_i\}_{i \in I}$ is boundedly linearly regular [GPR67, Equation (11)]. By Theorem 7.3, the algorithm is linearly focusing, so (i) follows from Theorem 4.6, and (ii) follows from Theorems 5.7 and 5.8; cf. Remark 5.9. □

Example 7.6. Suppose $0 < \underline{\alpha}_\infty, \overline{\alpha}_\infty < 2$ and the level set $\{x : f_j(x) \le 1\}$ of some f_j is boundedly compact. Then $\{x^n\}$ converges in norm to some point in C.

Proof. By Corollary 3.5, Definition 7.2, and Theorem 7.3, $\lim_{n:j\in\hat{I}^n} d_{H_j^n}(x^n) = 0$ yields $\lim_{n:j\in\hat{I}^n} f_j^+(x^n) = 0$, so there is a subsequence $\{x^{n'}\}$ with $j \in \hat{I}^{n'}$ and $f_j(x^{n'}) \le 1$. Hence $\{x^{n'}\}$ has a norm cluster point. The result follows from Theorem 3.8. □

Remark 7.7. If $f_i = d_{C_i}$ for some i, then $\partial f_i(x) = \frac{x-P_{C_i}x}{\|x-P_{C_i}x\|}$ if $x \notin C_i$, $\partial f_i(x) = \{g \in \mathcal{H} : \|g\| \le 1, \langle g, y-x\rangle \le 0 \ \forall y \in C_i\}$ if $x \in C_i$ [BaB96, Remark 7.6], and $P_{C_i} = P_{H_i^n}$ for all n.

THEOREM 7.8. *Let $I_\partial \subset I$ and $I'_\partial = I \setminus I_\partial$ be s.t. for some $\check{x} \in \cap_{i\in I'_\partial} C_i$, $\sup_{i\in I_\partial} f_i(\check{x}) < 0$, $f_i = d_{C_i}$ $\forall i \in I'_\partial$ and $\{C_i\}_{i\in I'_\partial}$ is boundedly linearly regular. Then $\{C_i\}_{i\in I}$ is boundedly linearly regular and the subgradient algorithm is linearly focusing (for $T_i = P_{C_i}$, $i \in I$). If $\{\alpha_n\} \subset [\epsilon, 2-\epsilon]$ for some $\epsilon > 0$, and the algorithm is either intermittent or considers remotest sets, then $\{x^n\}$* converges linearly *to some $x \in C$.*

Proof. Let $C' = \cap_{i\in I'_\partial} C_i$. By bounded linear regularity, for any bounded $S \subset \mathcal{H}$ there exists $\kappa'_S > 0$ s.t. $d_{C'}(x) \le \kappa'_S \max_{i\in I'_\partial} d_{C_i}(x)$ for all $x \in S$. Since $\check{x} \in C' \cap \operatorname{int} \bigcap_{i\in I_\partial} C_i$, C' and $\{C_i\}_{i\in I_\partial}$ are boundedly linearly regular [GPR67, Equation (11)], i.e., there is $\check{\kappa}_S > 0$ s.t. $d_C(x) \le \check{\kappa}_S \max\{d_{C'}(x), \max_{i\in I_\partial} d_{C_i}(x)\}$ for all $x \in S$. Hence, $d_C(x) \le \kappa_S \max_{i\in I} d_{C_i}(x)$ with $\kappa_S = \check{\kappa}_S \max\{\kappa'_S, 1\}$, i.e., $\{C_i\}_{i\in I}$ is boundedly linearly regular. By Theorem 7.3, $\beta d_{C_i}(x^n) \le d_{H_i^n}(x^n)$ $\forall i \in \hat{I}^n \cap I_\partial$ with $\beta > 0$, and $d_{C_i}(x^n) = d_{H_i^n}(x^n)$ $\forall i \in \hat{I}^n \cap I'_\partial$ (cf. Remark 7.7), so the algorithm is linearly focusing. Invoke Theorems 5.7 and 5.8. □

Remark 7.9. Theorems 7.4, 7.5, and 7.8 provide "long-step" extensions of the results of [BaB96, Theorems 7.15, 7.18, 7.22, and 7.27 and the associated examples].

To cover more examples, we now extend the subgradient framework as in [Kiw95].

DEFINITION 7.10. *We say the subgradient algorithm uses* analytic surrogates *if at Step 2, $\check{H}_i^n = \{x : \langle \check{a}^{in}, x\rangle \le \check{b}_{in}\}$ with $(\check{a}^{in}, \check{b}_{in}) = (g^{in}, \langle g^{in}, x^n\rangle - f_i(x^n))$ for $i \in \hat{I}^n$, whereas Step 3 finds a weight vector $\lambda^n \in \mathbb{R}_+^{|I|}$, $\lambda_i^n = 0$, $i \notin \hat{I}^n$, $\sum_{i\in I} \lambda_i^n = 1$ for the* surrogate inequality *$\langle \hat{a}^n, x\rangle \le \hat{b}_n$ with $(\check{a}^n, \check{b}_n) = \sum_{i\in\hat{I}^n} \lambda_i^n (\check{a}^{in}, \check{b}_{in})$ s.t. $\hat{H}^n = \{x : \langle \hat{a}^n, x\rangle \le \hat{b}_n\}$ satisfies*

$$(18) \qquad d_{\hat{H}^n}(x^n) \ge \lambda_{\min} \frac{\max_{i\in\hat{I}^n} f_i^+(x^n)}{\max_{i\in\hat{I}^n} \|g^{in}\|} = \lambda_{\min} \frac{\max_{i\in\hat{I}^n} (\langle \check{a}^{in}, x^n\rangle - \check{b}_{in})_+}{\max_{i\in\hat{I}^n} \|\check{a}^{in}\|}.$$

DEFINITION 7.11. *We say the subgradient algorithm* considers most violated constraints *if $\hat{I}^n$ contains some index $i_n \in \check{I}^n_{\mathrm{rem}} := \{i : f_i^+(x^n) = \max_{j\in I} f_j^+(x^n)\}$ for all n.*

Remark 7.12. Choices of weights satisfying (18) are discussed in [Kiw95] and in section 8.

Remark 7.13. The Fejér estimates (11) and (14) remain valid for analytic surrogates, since $\cap_{i\in I^n} C_i \subset \cap_{i\in I^n} \check{H}_i^n \subset \hat{H}^n$ from convexity and $\lambda^n \ge 0$. Hence, Lemmas 2.7 and 3.1 and Theorem 7.3 are also true. By (18) and Theorem 7.3,

$$(19) \qquad d_{\hat{H}^n}(x^n) \ge \lambda_{\min} \max_{i\in\hat{I}^n} f_i^+(x^n)/L \quad \text{with} \quad L = \max_{i\in I} L_i \qquad \text{for all } n \ge 0.$$

Hence, the results of section 3 are easily extended to the subgradient algorithm with analytic surrogates by replacing $d_{H_i^n}(x^n)$ with $f_i^+(x^n)/L$ in (6); "focusing" is replaced by the fact that $x^{n_k} \rightharpoonup x$ and $f_i^+(x^{n_k})/L \to 0$ imply $x \in C_i$ by weak lower semicontinuity of f_i. A similar argument extends Corollaries 4.4 and 4.5, Theorem 4.6, Corollary 4.7 (with $T_i = P_{C_i}$; cf. Remark 5.9), and Example 7.6. An extension of Theorem 4.10 is given below.

THEOREM 7.14. *Suppose the subgradient algorithm uses analytic surrogates and considers most violated constraints.*

(i) *If* $\sum_n \alpha_n(2-\alpha_n) = \infty$, *then* $\{x^n\}$ *has a unique weak cluster point in* C.

(ii) *If* $0 < \underline{\alpha}_\infty, \overline{\alpha}_\infty < 2$, *then* $x^n \rightharpoonup x$ *for some* $x \in C$ *and* $\max_{i\in I} f_i^+(x^n) \to 0$.

Proof. (i) By Lemma 3.1(iii), $\sum_n \alpha_n(2-\alpha_n) d^2_{\hat{H}^n}(x^n) < \infty$, so

$$\varliminf_n \max_{i\in I} f_i^+(x^n) = 0$$

by (19) with $\max_{i\in \hat{I}^n} f_i^+(x^n) = \max_{i\in I} f_i^+(x^n)$. Thus, we can find a subsequence $\{x^{n_k}\}$ and x s.t. $x^{n_k} \rightharpoonup x$ and $\max_{i\in I} f_i^+(x^{n_k}) \to 0$, so $x \in C$. By Corollary 3.2(iii), $\{x^n\}$ has at most one weak cluster point in C; so (i) holds. (ii) is proved similarly. □

We now extend [BaB96, Theorem 7.33(ii)] by specializing Theorems 5.7 and 5.8.

THEOREM 7.15. *Let* $I_\partial \subset I$ *and* $I'_\partial = I \setminus I_\partial$ *be s.t. for some* $\check{x} \in \cap_{i\in I'_\partial} C_i$, $\sup_{i\in I_\partial} f_i(\check{x}) < 0$, $f_i = d_{C_i}$ $\forall i \in I'_\partial$ *and* $\{C_i\}_{i\in I'_\partial}$ *is boundedly linearly regular. If the subgradient algorithm uses analytic surrogates, is either intermittent or considers most violated constraints, and* $\{\alpha_n\} \subset [\epsilon, 2-\epsilon]$ *for some* $\epsilon > 0$, *then* $\{x^n\}$ converges linearly *to some* $x \in C$.

Proof. $\{C_i\}_{i\in I}$ is boundedly linearly regular (Theorem 7.8). By Theorem 7.3, there is $\beta > 0$ s.t. $\beta d_{C_i}(x^n) \le f_i^+(x^n)/L$ for all $i \in \hat{I}^n \cap I_\partial$, and $d_{C_i}(x^n) = f_i^+(x^n)$ for all $i \in \hat{I}^n \cap I'_\partial$, so replacing β and L by $\min\{\beta, 1\}$ and $\max\{L, 1\}$, respectively, we have $\beta d_{C_i}(x^n) \le f_i^+(x^n)/L$ for all $i \in \hat{I}^n$ and $n \ge 0$. Hence, (19) may be used in the proofs of Theorems 5.7 and 5.8. □

8. Surrogate subgradient cuts.

We now show how to satisfy the requirements of Definition 7.10. To ease notation, let $H_i = \{x \in \mathcal{H} : \langle a^i, x\rangle \le b_i\}$, $a^i \in \mathcal{H} \setminus \{0\}$, $b_i \in \mathbb{R}$, $i \in J$, where $|J| < \infty$. Let $\hat{x} \notin \mathcal{P} := \cap_{i\in J} H_i$, $r_i = (\langle a^i, \hat{x}\rangle - b_i)_+$, $i \in J$, and $J_{\max} = \operatorname{Arg\,max}_i r_i$. In view of (18), our task is to find weights $\lambda_i \ge 0$, $\sum_i \lambda_i = 1$ for the surrogate inequality $\langle \hat{a}, x\rangle \le \hat{b}$ with $(\hat{a}, \hat{b}) = \sum_i \lambda_i (a^i, b_i)$ such that $\langle \hat{a}, \hat{x}\rangle > \hat{b}$ and $\hat{H} = \{x : \langle \hat{a}, x\rangle \le \hat{b}\}$ satisfies

$$(20) \quad d_{\hat{H}}(\hat{x}) = \frac{\langle \hat{a}, \hat{x}\rangle - \hat{b}}{\|\hat{a}\|} = \frac{\sum_{i\in J} \lambda_i(\langle a^i, \hat{x}\rangle - b_i)}{\|\sum_{i\in J} \lambda_i a^i\|} \ge \lambda_{\min} \frac{\max_{i\in J}(\langle a^i, \hat{x}\rangle - b_i)}{\max_{i\in J} \|a^i\|}.$$

Sometimes it is convenient to use the following condition that implies (20):

$$(21) \qquad d_{\hat{H}}(\hat{x}) \ge \lambda_{\min} \max_{i\in J} \frac{\langle a^i, \hat{x}\rangle - b_i}{\|a^i\|} = \lambda_{\min} \max_{i\in J} d_{H_i}(\hat{x}).$$

Example 8.1. First, (20) holds if $\lambda_{\hat{\imath}} \ge \lambda_{\min}$ for some $\hat{\imath} \in J_{\max}$ and $\lambda_i = 0$ whenever $\langle a^i, \hat{x}\rangle < b_i$, since then $\langle \hat{a}, \hat{x}\rangle - \hat{b} = \sum_i \lambda_i r_i$ and $\|\hat{a}\| \le \sum_i \lambda_i \max_i \|a^i\|$ by convexity of $\|\cdot\|$.

Example 8.2. Let $\psi : \mathbb{R}_+ \to \mathbb{R}_+$ be nondecreasing with $\psi(t) > 0$ for $t \ge \max_i r_i$. Choose $\mu \in \mathbb{R}_+^{|J|}$ with $\sum_i \mu_i = 1$ and $\mu_{\hat{\imath}} \ge \lambda_{\min}$ for some $\hat{\imath} \in J_{\max}$. Let $\lambda_i =$

$\mu_i\psi(r_i)/\sum_j \mu_j\psi(r_j)$, $i \in J$. Then $\lambda_{\hat\imath} \ge \mu_{\hat\imath}/\sum_i \mu_i \ge \lambda_{\min}$, so (20) holds by Example 8.1. Examples include $\psi(t) = t^\gamma$ with $\gamma \ge 0$, and $\psi(t) = 0$ if $t < \max_i r_i$, 1 otherwise.

Example 8.3. Choose ψ as in Example 8.2 and $m_{ij} \ge 0$, $i, j \in J$, s.t.

$$m_{\hat\imath\hat\imath}/\sum_{i,j} m_{ij} \ge \lambda_{\min}$$

for some $\hat\imath \in J_{\max}$. Let $\lambda_i = \sum_j m_{ij}\psi(r_j)/\sum_{k,j} m_{kj}\psi(r_j)$, $i \in J$. Then $\lambda_{\hat\imath} \ge m_{\hat\imath\hat\imath}\psi(r_{\hat\imath})/\sum_{k,j} m_{kj}\psi(r_{\hat\imath}) \ge \lambda_{\min}$, so (20) holds by Example 8.1.

Example 8.4. Choosing ψ and μ as in Example 8.2, let

$$\lambda_i = (\mu_i\psi(r_i)/\|a^i\|)/\sum_j \mu_j\psi(r_j)/\|a^j\|, \qquad i \in J.$$

Then (20) holds, since $d_{H_{\hat\imath}}(\hat x) = r_{\hat\imath}/\|a^{\hat\imath}\| \ge r_{\hat\imath}/\max_i \|a^i\|$ and

$$d_{\hat H}(\hat x) = \frac{\sum_i \mu_i\psi(r_i)d_{H_i}(\hat x)}{\|\sum_i \mu_i\psi(r_i)a^i/\|a^i\|\|} \ge \frac{\sum_i \mu_i\psi(r_i)d_{H_i}(\hat x)}{\sum_i \mu_i\psi(r_i)} \ge \frac{\mu_{\hat\imath}\psi(r_{\hat\imath})d_{H_{\hat\imath}}(\hat x)}{\sum_i \mu_i\psi(r_{\hat\imath})} = \mu_{\hat\imath}d_{H_{\hat\imath}}(\hat x).$$

Example 8.5. Let $\lambda_i = (r_i/\|a^i\|^2)/\sum_j(r_j/\|a^j\|^2)$, $i \in J$. Then (21) holds if $\lambda_{\min} \le |J|^{-1/2}$, since by convexity of $\|\cdot\|^2$

$$d^2_{\hat H}(\hat x) = \frac{\sum_i \|r_i a^i/\|a^i\|^2\|^2}{\|\sum_i r_i a^i/\|a^i\|^2\|^2} \sum_i \frac{r_i^2}{\|a^i\|^2} \ge \frac{1}{|J|}\max_i d^2_{H_i}(\hat x).$$

Example 8.6. The *deepest* surrogate cut that maximizes $d_{\hat H}(\hat x)$ has weights $\check\lambda_i$ that solve

$$\max\left\{\frac{\sum_i \lambda_i(\langle a^i, \hat x\rangle - b_i)}{\|\sum_i \lambda_i a^i\|} : \lambda_i \ge 0, i \in J, \sum_i \lambda_i = 1\right\}.$$

Following [Kiw96b, section 5], we may equivalently find

$$(22) \qquad \hat\lambda \in \operatorname{Arg\,min}\left\{\left\|\sum_i \lambda_i a^i\right\|^2/2 + \sum_i \lambda_i(b_i - \langle a^i, \hat x\rangle) : \lambda \ge 0\right\},$$

and let $\check\lambda_j = \hat\lambda_i/\sum_j \hat\lambda_j$, $i \in J$ (then $P_{\mathcal P}(\hat x) = \hat x - \sum_i \hat\lambda_i a^i$ and $\hat\lambda$ is a Lagrange multiplier for $\min\{\|x - \hat x\|^2/2 : \langle a^i, x\rangle \le b_i, i \in J\}$). If the Gram matrix G with entries $\langle a^i, a^j\rangle$ is available, (22) can be solved by standard active-set QP methods. The restricted active-set method of [Kiw95, section 8] can find, using little storage [Kiw94], in just two iterations an approximate solution to (22) for which (21) holds with $\lambda_{\min} = 1$; see also [Kiw97].

Remark 8.7. The preceding examples also show how to find surrogates in the geometric framework of section 2. To this end, note that for $J = \hat I^n$, $(a^i, b_i) = (a^{in}, b_{in})$ and $\lambda_i = \lambda_i^n$, $i \in \hat I^n$, (6) and (20)–(21) are *equivalent* due to (9). In particular, Example 8.2 yields a "long-step" extension of [BaB96, Example 6.32].

9. Scalarized subgradient cuts.

Maintaining Assumption 7.1, we now extend the scalarized subgradient cuts of [Kiw95, Example 3.6], which generalized those of [Oet75].

For $y, z \in \mathbb{R}^N$, let $|y| = (|y_1|, \dots, |y_N|)$ and $\langle y, z\rangle = \sum_{i=1}^N y_i z_i$. Let $\|\cdot\|$ be any *monotone* norm on $\mathbb{R}^N$ s.t. $|y| \le |z| \Rightarrow \|y\| \le \|z\|$, e.g., $\|y\| = \|y\|_p := (\sum_{i=1}^N |y_i|^p)^{1/p}$, $1 \le p \le \infty$. The *dual* norm defined by $\|z\|_* = \max_{\|y\|\le 1} \langle y, z\rangle$ satisfies $\|y\| = \max_{\|z\|_*\le 1} \langle y, z\rangle$ and is monotone, since $\||y|\| = \|y\|$ and $\max_{\|y\|\le 1} \langle y, z\rangle = \max_{\|y\|\le 1} \langle |y|, |z|\rangle$. Hence, $\|y\| = \max_{z\in Z} \langle y, z\rangle$ if $y \ge 0$, where $Z = \{z \in \mathbb{R}^N_+ : \|z\|_* \le 1\}$. For each $x \in \mathcal{H}$, let $y(x) := (f_1^+(x), \dots, f_N^+(x)) \ge 0$ and

$$f(x) := \|y(x)\| = \max\{\langle y(x), z\rangle : z \in Z\}. \tag{23}$$

Theorem 9.1. *f is convex and Lipschitz continuous on bounded sets. For each $\hat{x} \in \mathcal{H}$,*

$$\partial f(\hat{x}) = \left\{ \sum_{i\in I} z_i \partial f_i^+(\hat{x}) : z \in Z(\hat{x}) \right\} = \left\{ \sum_{i\in \check{I}(\hat{x})} z_i \partial f_i(\hat{x}) : z \in Z(\hat{x}) \right\}, \tag{24}$$

where $Z(\hat{x}) = \operatorname{Arg\,max}_{z\in Z} \langle y(\hat{x}), z\rangle$, $\check{I}(\hat{x}) = \{i : f_i(\hat{x}) \ge 0\}$, $\partial f_i^+(\hat{x}) = \partial f_i(\hat{x})$ if $f_i(\hat{x}) > 0$, $\partial f_i^+(\hat{x}) = \cup_{0\le\lambda\le 1} \lambda \partial f_i(\hat{x})$ if $f_i(\hat{x}) = 0$, $\partial f_i^+(\hat{x}) = \{0\}$ if $f_i(\hat{x}) < 0$. Consequently, $\sup\{\|g\| : g \in \partial f(\hat{x})\} \le \sup_{\cdot} \frac{\|\cdot\|_1}{\|\cdot\|_*} \sup\{\|g\| : g \in \cup_{i\in I} \partial f_i(\hat{x})\}$.

Proof. Since $Z \subset \mathbb{R}^N_+$ is compact and each f_i is convex and Lipschitz continuous on bounded sets (cf. Assumption 7.1), so are f_i^+, $\langle y(\cdot), z\rangle$ for any $z \in Z$, and f. Since $z \ge 0$ and f_i^+ are continuous, $\partial \langle y(\cdot), z\rangle (\hat{x}) = \sum_i z_i \partial f_i^+(\hat{x})$ (cf. [IoT74, Theorem 4.2.1]). Therefore, [IoT74, Theorem 4.2.3], $\partial f(\hat{x}) = \overline{\operatorname{co}}\, G(\hat{x})$, where $G(\hat{x}) = \{\sum_{i=1}^N z_i \partial f_i^+(\hat{x}) : z \in Z(\hat{x})\}$. By a similar argument using $f_i^+ = \max_{\lambda\in[0,1]} \lambda f_i$, we have $\partial f_i^+(\hat{x}) = \partial f_i(\hat{x})$ if $f_i(\hat{x}) > 0$, $\partial f_i^+(\hat{x}) = \{0\}$ if $f_i(\hat{x}) < 0$, and $\partial f_i^+(\hat{x}) = \overline{\operatorname{co}} \cup_{0\le\lambda\le 1} \lambda \partial f_i(\hat{x})$ if $f_i(\hat{x}) = 0$. It is easy to see that if $\tilde{Z} \subset \mathbb{R}^m_+$ and $A_i \subset \mathcal{H}$, $i = 1\colon m$, are convex, then so is $A = \cup_{\tilde{z}\in\tilde{Z}} \sum_i \tilde{z}_i A_i$ (cf. [STF86, Lemma 3.4.1]), whereas if $\tilde{Z}$ is compact and each A_i is bounded and weakly closed, so is A. But each $\partial f_i(\hat{x})$ is bounded, convex, and weakly closed [Phe93, Proposition 1.11]. Hence $\overline{\operatorname{co}}\, G(\hat{x}) = G(\hat{x})$ and $\partial f_i^+(\hat{x}) = \cup_{0\le\lambda\le 1} \lambda \partial f_i(\hat{x})$ if $f_i(\hat{x}) = 0$. If $z \in Z(\hat{x})$ and $g = \sum_i z_i \lambda_i g^i$ with $g^i \in \partial f_i(\hat{x})$, $\lambda_i = 1$ if $f_i(\hat{x}) > 0$, $\lambda_i \in [0, 1]$ if $f_i(\hat{x}) = 0$, $\lambda_i = 0$ if $f_i(\hat{x}) < 0$, and $\check{z}_i = z_i \lambda_i$ for all i, then $|\check{z}| \le |z|$, $\|\check{z}\|_* \le \|z\|_* \le 1$, $\check{z} \ge 0$, and $\langle y(\hat{x}), \check{z}\rangle = \langle y(\hat{x}), z\rangle$, i.e., $\check{z} \in Z(\hat{x})$, and $g = \sum_{i\in\check{I}(\hat{x})} \check{z}_i g^i$. Finally, $\|g\| \le \hat{L} \sum_i z_i = \hat{L}\|z\|_1 \le \hat{L} \sup_{\cdot} \frac{\|\cdot\|_1}{\|\cdot\|_*}$, where $\hat{L} = \sup\{\|g\| : g \in \cup_{i=1}^N \partial f_i(\hat{x})\}$. □

Definition 9.2. *The algorithm is called a* scalarized subgradient algorithm *if Step* 3 *sets $\hat{H}^n = \{x : f(x^n) + \langle g^n, x - x^n\rangle \le 0\}$ for some $g^n \in \partial f(x^n)$.*

Corollary 9.3. *The scalarized subgradient algorithm uses analytic surrogates in the sense that if $g^n = \sum_{i\in\check{I}(x^n)} z_i^n g^{in}$ with $z^n \in Z(x^n)$ and $g^{in} \in \partial f_i(x^n)$, then for λ^n defined by $\lambda_i^n = z_i^n / \sum_{j\in\check{I}(x^n)} z_j^n$, $i \in \check{I}(x^n)$, $\lambda_i^n = 0$, $i \notin \check{I}(x^n)$,* (18) *holds with $\hat{I}^n = I$ and*

$$\lambda_{\min} = \left(\sup_{\cdot} \frac{\|\cdot\|_1}{\|\cdot\|_*} \sup_{\cdot} \frac{\|\cdot\|_\infty}{\|\cdot\|} \right)^{-1}. \tag{25}$$

Proof. Clearly,

$$d_{\hat{H}^n}(x^n) = \frac{f(x^n)}{\|g^n\|} = \frac{\|y(x^n)\| / \sum_{i \in \check{I}(x^n)} z_i^n}{\| \sum_i \lambda_i^n g^{in} \|},$$

where

$$\|y(x^n)\| \geq \|y(x^n)\|_\infty \inf_{\cdot} \frac{\| \cdot \|}{\| \cdot \|_\infty},$$

$$\sum_{i \in \check{I}(x^n)} z_i^n \leq \|z^n\|_1 \leq \|z^n\|_* \sup_{\cdot} \frac{\| \cdot \|_1}{\| \cdot \|_*} \leq \sup_{\cdot} \frac{\| \cdot \|_1}{\| \cdot \|_*},$$

and $\| \sum_i \lambda_i^n g^{in} \| \leq \max_i \|g^{in}\|$. □

Remark 9.4. If $\| \cdot \| = \| \cdot \|_p$, $\gamma = p - 1 \geq 0$ and q satisfies $\frac{1}{p} + \frac{1}{q} = 1$, then $z_i^n = f_i^+(x^n)^\gamma / (\sum_j f_j^+(x^n)^p)^{1/q}$ and $\lambda_i^n = f_i^+(x^n)^\gamma / (\sum_j f_j^+(x^n)^\gamma)$, $i \in I$, as in Example 8.2.

Remark 9.5. Since scalarized subgradient algorithms use analytic surrogates and consider most violated constraints ($\hat{I}^n \equiv I$), their convergence is described by Remark 7.13 and Theorems 7.14 and 7.15 (with $I'_\partial = \emptyset$). Alternatively, since $C = \{x : f(x) \leq 0\}$, they may be viewed as subgradient algorithms with N and f_1 replaced by 1 and f, respectively. Of course, f cannot have a Slater point, but this is not really necessary.

THEOREM 9.6. *If there is some Slater point $\check{x}$ s.t. $f_i(\check{x}) < 0$, $\forall i \in I$, then $d_C(x) \leq \frac{\|\check{x} - x\|}{-\max_{i \in I} f_i(\check{x})} \sup_{\cdot} \frac{\| \cdot \|_\infty}{\| \cdot \|} f(x)$ for any x.*

Proof. Let $\epsilon = -\max_i f_i(\check{x})$, $t = \min_i \frac{\epsilon}{\epsilon + f_i^+(x)}$ and $y = (1-t)\check{x} + tx$. Then for each i, $f_i(y) \leq (1-t) f_i(\check{x}) + t f_i(x) \leq 0$, so $y \in C_i$ and $d_C(x) \leq \|x - y\| = (1-t)\|\check{x} - x\|$. But $1 - t = \max_i \frac{f_i^+(x)}{\epsilon + f_i^+(x)} \leq \max_i \frac{f_i^+(x)}{\epsilon}$ and $\|y(x)\|_\infty \leq f(x) \sup_{\cdot} \frac{\| \cdot \|_\infty}{\| \cdot \|}$. □

10. Polyhedral framework.

We now consider the case where each f_i is a *polyhedral* function of the form $\phi(x) = \max_{j=1}^m (\langle a^j, x \rangle - b_j)$ with $a^j \in \mathcal{H}$, $b_j \in \mathbb{R}$, $m < \infty$. Then $\partial \phi(x) = \mathrm{co}\{a^j : \langle a^j, x \rangle - b_j = \phi(x)\}$ (cf. Theorem 9.1) and $\|g\| \leq L_\phi$ for each $g \in \partial \phi(x)$, where $L_\phi = \max_j \|a^j\|$ is the Lipschitz constant of ϕ. We shall need the following version of Hoffman's lemma [Hof52].

LEMMA 10.1. *Consider a nonempty polyhedron $\mathcal{P} = \{x \in \mathcal{H} : \langle a^i, x \rangle \leq b_i, i = 1{:}\, m\}$. There exists $\alpha > 0$ s.t. $\phi(x) := \max_{i=1}^m (\langle a^i, x \rangle - b_i)_+ \geq \alpha d_{\mathcal{P}}(x)$ for all $x \in \mathcal{H}$.*

Proof. Suppose $\mathcal{P} \neq \mathcal{H}$ (otherwise let $\alpha = 1$). For any $\hat{I} \subset \{1{:}\, m\}$ s.t. $\{a^i\}_{i \in \hat{I}}$ are positively independent, let $\alpha_{\hat{I}} = \min\{\| \sum_{i \in \hat{I}} \mu_i a^i \| : \mu_i \geq 0, \sum_{i \in \hat{I}} \mu_i = 1\}$, and let α be the minimum of such $\alpha_{\hat{I}}$. Let $\hat{x} \notin \mathcal{P}$ and $\bar{x} = \arg\min\{\|x - \hat{x}\|^2/2 : \langle a^i, x \rangle \leq b_i, i = 1{:}\, m\}$. By the optimality conditions [Lau72, section 2.3], there exists $\lambda \in \mathbb{R}_+^m$ s.t. $\bar{x} - \hat{x} + \sum_i \lambda_i a^i = 0$ and $\sum_i \lambda_i (\langle a^i, \bar{x} \rangle - b_i) = 0$. Since $\bar{x} \neq \hat{x}$, $\sum_i \lambda_i a^i \neq 0$. By a classic reduction argument (cf. [IoT74, section 3.5.1]), we may assume that $\{a^i : \lambda_i > 0\}$ are positively independent. Let $\mu = \lambda / \sum_i \lambda_i$. Clearly, $g := \sum_i \mu_i a^i \in \partial \phi(\bar{x})$ and $\phi(\bar{x}) = 0$, so $\phi(\hat{x}) \geq \langle g, \hat{x} - \bar{x} \rangle$. Since $d_{\mathcal{P}}(\hat{x}) = \|\hat{x} - \bar{x}\|$ and $\langle g, \frac{\hat{x} - \bar{x}}{\|\hat{x} - \bar{x}\|} \rangle = \|g\| \geq \alpha$, the conclusion follows. □

COROLLARY 10.2. *If each f_i is polyhedral, then there exist $\alpha_i > 0$ and L_i s.t. $\alpha_i d_{C_i}(x) \leq f_i^+(x) \leq L_i d_{C_i}(x)$ for all $x \in \mathcal{H}$. Further, there exist $\alpha > 0$ and $L < \infty$ s.t. $\alpha d_C(x) \leq \max_{i \in I} f_i^+(x) \leq L \max_{i \in I} d_{C_i}(x)$ for all x. In particular, $\{C_i\}_{i \in I}$ is linearly regular. Moreover* (*cf.* (23)), *$\alpha \inf_{\cdot} \frac{\| \cdot \|}{\| \cdot \|_\infty} d_C(x) \leq f(x)$ for all x.*

Proof. For each i, let L_i be the Lipschitz constant of f_i, so that $f_i^+(x) = |f_i^+(x) - f_i^+(P_{C_i}x)| \leq L_i d_{C_i}(x)$ for all x, let $L = \max_i L_i$ and invoke Lemma 10.1. ∎

THEOREM 10.3. *Suppose the subgradient algorithm uses analytic surrogates, is intermittent or considers most violated constraints, $\{\alpha_n\} \subset [\epsilon, 2-\epsilon]$ for some $\epsilon > 0$, and each f_i is polyhedral. Then $\{x^n\}$ converges linearly to some point in C with a rate independent of the starting point.*

Proof. $\{C_i\}_{i \in I}$ is linearly regular and there are $\beta > 0$ and L s.t. $\beta d_{C_i}(x^n) \leq f_i^+(x^n)/L$ and $\|g^{in}\| \leq L$ for all $i \in \hat{I}^n$ and $n \geq 0$ (Corollary 10.2). Hence, (cf. (18)) (19) may be used in the proofs of Theorems 5.7 and 5.8 (with $T_i = P_{C_i}$ for all i). ∎

THEOREM 10.4. *Suppose the subgradient algorithm uses analytic surrogates, is intermittent or considers most violated constraints, $\{\alpha_n\} \subset [\epsilon, 2-\epsilon]$ for some $\epsilon > 0$, and $f_i = d_{C_i}$ with C_i polyhedral for $i \in I$. Then $\{x^n\}$ converges linearly to some point in C with a rate independent of the starting point.*

Proof. $\{C_i\}_{i \in I}$ is linearly regular (Corollary 10.2) and $d_{C_i}(x^n) = f_i^+(x^n)$ with $\|g^{in}\| \leq 1$ (cf. Remark 7.7) for all $i \in \hat{I}^n$ and $n \geq 0$. Hence, (cf. (18)) (19) with $L = 1$ may be used in the proofs of Theorems 5.7 and 5.8 (with $T_i = P_{C_i}$ for all i). ∎

THEOREM 10.5. *Suppose the scalarized subgradient algorithm generates $\{\alpha_n\} \subset [\epsilon, 2-\epsilon]$ for some $\epsilon > 0$ and each f_i is polyhedral. Then $\{x^n\}$ converges linearly to some point in C with a rate independent of the starting point.*

Proof. There are $\beta > 0$ and L_f s.t. $\beta d_C(x^n) \leq f(x^n)/L_f$ (Corollary 10.2), $\|g^n\| \leq L_f$ (Theorem 9.1) and hence $d_{\hat{H}^n}(x^n) \geq f(x^n)/L_f$ (cf. Definition 9.2) for all $n \geq 0$. This suffices for modifying the proof of Theorem 5.8 (with $T_1 = P_{C_1}$, and C_1 and N replaced by C and 1). ∎

Remark 10.6. Theorems 10.4 and 10.5 extend ones in [BaB96, Theorem 7.36] and [Oet75, p. 48].

11. Infinite constraint sets. We now consider the case where I is countably infinite. The control of Step 1 is *chaotic*, i.e., $I = \limsup_n \check{I}^n$. One may select $M \geq 1$ and choose $\hat{I}^n$ s.t. $|\hat{I}^n| \leq M$; see [Com95b] for examples of chaotic and admissible controls. If $|\hat{I}^n| = \infty$, it suffices to find a finite $\check{I}^n \subset \hat{I}^n$ s.t. $\max_{i \in \check{I}^n} d_{H_i^n}(x^n) \geq \frac{1}{2}\sup_{i \in \hat{I}^n} d_{H_i^n}(x^n)$; then $\check{I}^n$ may replace $\hat{I}^n$ in (5) to get

$$d_{\hat{H}^n}(x^n) \geq \lambda_{\min} \sup_{i \in \hat{I}^n} d_{H_i^n}(x^n)$$

for any $\lambda_{\min} \in (0, \frac{1}{2}]$. We still say the algorithm considers remotest sets if

$$\forall n \exists i_n \in \hat{I}^n : \qquad \hat{d}_{i_n}(x^n) \geq \frac{1}{2} \sup_{i \in I} \hat{d}_i(x^n);$$

more generally, *coercive* control demands that

$$\sup_{i \in I} \hat{d}_i(x^{n_k}) \to 0 \quad \text{whenever} \quad \hat{d}_{i_{n_k}}(x^{n_k}) \to 0 \quad \text{with} \quad i_{n_k} \in \hat{I}^{n_k}.$$

Remark 11.1. The results of sections 3–6 extend easily to $|I| = \infty$, except for the following: Theorem 3.8, Corollary 4.5, Theorems 4.10 and 5.2–5.5, and Example 6.1.

THEOREM 11.2. *Theorems* 4.10 *and* 5.2–5.5 *hold for $|I| = \infty$ with "strongly focusing" replaced by "linearly focusing"* (*and* $\max_{i \in I} \hat{d}_i$ *by* $\sup_{i \in I} \hat{d}_i$).

Proof. Theorem 4.10: For (i), again $\underline{\lim}_n d_{H^n_{i_n}}(x^n) = 0$ and $\beta\hat{d}_{i_n}(x^n) \le d_{H^n_{i_n}}(x^n)$ (Definition 4.1) yield $\underline{\lim}_n \sup_{i\in I} \hat{d}_i(x^n) = 0$, so there is a subsequence $x^{n_k} \rightharpoonup x$ s.t. $\sup_{i\in I} \hat{d}_i(x^{n_k}) \to 0$; the rest follows as before. (ii) is proved similarly. Theorem 5.2: By Definition 3.10 and (15) with $x^{n_k p} \rightharpoonup x \in C$, $\forall\epsilon > 0\ \exists\bar{k}\ \forall k \ge \bar{k}$ $\forall i \in I\ \exists m_k \in \{kp{:}\,(k+1)p-1\}$: $i \in \hat{I}^{m_k}$, $d_{H_i^{m_k}}(x^{m_k}) < \epsilon$ and $\|x^{m_k} - x^{n_k p}\| \le \sum_{l=n_k p}^{(n_k+1)p-1} \|x^{l+1} - x^l\| < \epsilon$. But $\beta\hat{d}_i(x^{m_k}) \le d_{H_i^{m_k}}(x^{m_k})$ for all large k (Definition 4.1) and $\hat{d}_i$ is nonexpansive (Remark 4.2), so $\sup_{i\in I} \hat{d}_i(x^{n_k p}) \to 0$, $d_C(x^{n_k p}) \to 0$ (bounded regularity) and $x^n \to x$ (Corollary 3.2(ii)). Theorem 5.3: By Theorem 4.10(i), $\sup_{i\in I} \hat{d}_i(x^{n_k}) \to 0$ and bounded regularity yield $d_C(x^{n_k}) \to 0$, so $x^n \to x$ (Corollary 3.2(ii)). Theorem 5.4: By Theorem 3.13(i) and Definition 4.1, $\beta \sup_{i\in I} \hat{d}_i(x^{n_k}) \le \sup_{i\in I} d_{H_i^{n_k}}(x^{n_k}) \to 0$ yields $d_C(x^{n_k}) \to 0$ by bounded regularity; apply Corollary 3.2(ii). Theorem 5.5: By Definition 3.14 and (16) with $x^{\tau_{k'}} \rightharpoonup x \in C$, $\forall\epsilon > 0\ \exists\bar{k}\ \forall k' \ge \bar{k}\ \forall i \in I\ \exists n_{k'} \in \{\tau_{k'}{:}\,\tau_{k'+1} - 1\}$: $i \in \hat{I}^{n_{k'}}$, $d_{H_i^{n_{k'}}}(x^{n_{k'}}) < \epsilon$ and $\|x^{n_{k'}} - x^{\tau_{k'}}\| \le \sum_{l=\tau_{k'}}^{\tau_{k'+1}-1} \|x^{l+1} - x^l\| < \epsilon$. But $\beta\hat{d}_i(x^{\tau_{k'}}) \le d_{H_i^{\tau_{k'}}}(x^{\tau_{k'}})$ for all large k' (Definition 4.1) and $\hat{d}_i$ is nonexpansive (Remark 4.2), so $\sup_{i\in I} \hat{d}_i(x^{\tau_{k'}}) \to 0$, $d_C(x^{\tau_{k'}}) \to 0$ (bounded regularity) and $x^n \to x$ (Corollary 3.2(ii)). □

Remark 11.3. Theorems 4.10 and 5.3 hold under approximate remotest set control (Remark 4.11); under coercive control, "strongly focusing" should be replaced by "linearly focusing."

THEOREM 11.4. *Suppose the algorithm is linearly focusing,* $0 < \underline{\alpha}_\infty, \overline{\alpha}_\infty < 2$ *and the control is* chaotically coercive, *i.e.,* $\exists \hat{\imath}_k \in \hat{I}^{n_k}$: $\hat{d}_{\hat{\imath}_k}(x^{n_k}) \to 0 \Rightarrow \sup_{i\in I} \hat{d}_i(x^{n_k}) \to 0$. *Then the following hold.*

(i) $\{x^n\}$ *converges weakly to some point in* C, *and* $\sup_{i\in I} \hat{d}_i(x^{n_k}) \to 0$.

(ii) *If* $\{T_i\}_{i\in I}$ *is boundedly regular, then* $\{x^n\}$ *converges in norm to some point in* C.

(iii) *If some* T_j *is demicompact* (*cf. Example* 6.1) *and* $|\{n : j \in \hat{I}^n\}| = \infty$, *then* $\{x^n\}$ *converges in norm to some point in* C.

Proof. (i) By Definition 4.1 and Corollary 3.5, $\beta\hat{d}_{\hat{\imath}_k}(x^{n_k}) \le d_{H^{n_k}_{\hat{\imath}_k}}(x^{n_k}) \to 0$ yields $\sup_{i\in I} \hat{d}_i(x^{n_k}) \to 0$, so if $x^{n_{k'}} \rightharpoonup x$, then $x \in C$; the result follows from Corollary 3.2(iii). (ii) Bounded regularity and $\sup_{i\in I} \hat{d}_i(x^{n_k}) \to 0$ yield $d_C(x^{n_k}) \to 0$; apply Corollary 3.2(ii). (iii) By the proof of Example 6.1, $\{x^n\}$ has a norm cluster point, which must lie in C by (i); the result follows from [BaB96, Theorem 2.16(v)]. □

THEOREM 11.5. *Suppose* $0 < \underline{\alpha}_\infty, \overline{\alpha}_\infty < 2$, *and the control is* admissible, *i.e., there exist positive integers* $\{M_i\}_{i\in I}$ *s.t.* $\forall i \in I\ \forall n \ge 0$: $i \in \bigcup_{l=n}^{n+M_i-1} I^l$. *Then the following hold.*

(i) $\{x^n\}$ *converges weakly to some point in* C, *and*

$$\sum_{l=n}^{n+M_i-1} \left\{ \max_{i\in \hat{I}^l} d_{H_i^l}(x^l) + \|x^{l+1} - x^l\| \right\} \to 0 \qquad \forall i \in I. \tag{26}$$

(ii) *If the algorithm is linearly focusing, then* $\sup_{i\in I} \hat{d}_i(x^n) \to 0$, *so that if* $\{T_i\}_{i\in I}$ *is boundedly regular, then* $\{x^n\}$ *converges in norm to some point in* C.

(iii) *If the algorithm is linearly focusing and some* T_j *is demicompact* (*cf. Example* 6.1), *then* $\{x^n\}$ *converges in norm to some point in* C.

Proof. Use the proofs of Theorem 3.15 for (i) and Theorem 11.2 for (ii) with $\tau_k = n$, $\tau_{k+1} = n + M_i$, $k = k' = n$ for $i \in I$. (iii) Use the proof of Theorem

11.4(iii). ∎

Remark 11.6. When $|I| = \infty$, Assumption 7.1 should require that ∂f_i be nonempty and uniformly bounded with respect to bounded sets and all $i \in I$, so that $L = \sup_{i\in I} L_i < \infty$ (cf. Theorem 7.3). We still say the algorithm considers most violated constraints if $\forall n \exists i_n \in \hat{I}^n$: $f_{i_n}^+(x^n) \geq \frac{1}{2} \sup_{i\in I} f_i^+(x^n)$. Again, the results of section 7 extend easily, except for Example 7.6 and the part of Remark 7.13 related to the exceptions of Remark 11.1; Theorems 11.4(i) and 11.5(i) with $\hat{d}_i$ replaced by f_i^+ extend to the subgradient algorithm with analytic surrogates as in Remark 7.13.

Acknowledgments. We would like to thank the two anonymous referees for their valuable comments, which enabled us to improve our results.

REFERENCES

[AuE84] J. -P. Aubin and I. Ekeland, *Applied Nonlinear Analysis*, Wiley, New York, 1984.

[BaB96] H. H. Bauschke and J. M. Borwein, *On projection algorithms for solving convex feasibility problems*, SIAM Rev., 38 (1996), pp. 367–426.

[BrR77] R. E. Bruck, Jr. and S. Reich, *Nonexpansive projections and resolvents of accretive operators in Banach spaces*, Houston J. Math., 3 (1977), pp. 459–470.

[Com95a] P. L. Combettes, *Construction d'un point fixe commun á une famille de contractions fermes*, C. R. Acad. Sci. Paris, Sér. I Math., 320 (1995), pp. 1385–1390.

[Com95b] P. L. Combettes, *Convex Set Theoretic Image Recovery by Extrapolated Iterations of Parallel Subgradient Projections*, Tech. report, Dept. of Electrical Eng., City College and Graduate School, City Univ. of New York, New York, 1995.

[Com97] P. L. Combettes, *Hilbertian convex feasibility problem: Convergence of projection methods*, Appl. Math. Optim., 35 (1997), pp. 311–330.

[EcB92] J. Eckstein and D. P. Bertsekas, *On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Math. Programming, 55 (1992), pp. 293–318.

[Eck89] J. Eckstein, *Splitting Methods for Monotone Operators with Applications to Parallel Optimization*, Ph.D. thesis, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, MA, June 1989. Report LIDS-TH-1877, Laboratory for Information and Decision Sciences, MIT.

[Flå95] S. D. Flåm, *Successive averages of firmly nonexpansive mappings*, Math. Oper. Res., 20 (1995), pp. 497–512.

[GoR84] K. Goebel and S. Reich, eds., *Uniform Convexity, Hyperbolic Geometry and Nonexpansive Mappings*, Marcel Dekker, New York, 1984.

[GP93] U. García-Palomares, *Parallel projected aggregation methods for solving the convex feasibility problem*, SIAM J. Optim., 3 (1993), pp. 882–900.

[GPR67] L. G. Gurin, B. T. Polyak, and E. V. Raik, *The method of projections for finding a common point of convex sets*, Zh. Vychisl. Mat. i Mat. Fiz., 7 (1967), pp. 1211–1228 (in Russian). English transl. in U. S. S. R. Comput. Math. and Math. Phys., 7 (1967), pp. 1–24.

[Hof52] A. J. Hoffman, *On approximate solutions of systems of linear inequalities*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 263–265.

[IoT74] A. D. Ioffe and V. M. Tikhomirov, *Theory of Extremal Problems*, Nauka, Moscow, 1974 (in Russian). English transl., North–Holland, Amsterdam, 1979.

[KiŁ94] K. C. Kiwiel and B. Łopuch, *Surrogate Projection Methods for Finding Fixed Points of Firmly Nonexpansive Mappings*, Tech. report, Systems Research Institute, Warsaw, November 1994. Revised January 1996.

[Kiw94] K. C. Kiwiel, *A Cholesky dual method for proximal piecewise linear programming*, Numer. Math., 68 (1994), pp. 325–340.

[Kiw95] K. C. Kiwiel, *Block-iterative surrogate projection methods for convex feasibility problems*, Linear Algebra Appl., 215 (1995), pp. 225–260.

[Kiw96a] K. C. Kiwiel, *The efficiency of subgradient projection methods for convex optimization, part* I*: General level methods*, SIAM J. Control Optim., 34 (1996), pp. 660–676.

[Kiw96b] K. C. Kiwiel, *The efficiency of subgradient projection methods for convex optimization, part* II*: Implementations and extensions*, SIAM J. Control Optim., 34 (1996), pp. 677–697.

[Kiw97] K. C. Kiwiel, *Monotone Gram matrices and deepest surrogate inequalities in accelerated relaxation methods for convex feasibility problems*, Linear Algebra Appl., 252 (1997), pp. 27–33.

[Lau72] P. J. Laurent, *Approximation et Optimisation*, Hermann, Paris, 1972.

[Mer62] Yu. I. Merzlyakov, *On a relaxation method of solving systems of linear inequalities*, Zh. Vychisl. Mat. i Mat. Fiz., 2 (1962), pp. 482–487 (in Russian). English transl. in U. S. S. R. Comput. Math. and Math. Phys., 2 (1962), pp. 504–510.

[Oet75] W. Oettli, *Symmetric duality, and a convergent subgradient method for discrete, linear, constrained approximation problems with arbitrary norms appearing in the objective function and in the constraints*, Approx. Theory Appl., 14 (1975), pp. 43–50.

[Opi67] Z. Opial, *Weak convergence of the sequence of successive approximations for nonexpansive mappings*, Bull. Amer. Math. Soc. (N.S.), 73 (1967), pp. 591–597.

[Phe93] R. R. Phelps, *Convex Functions, Monotone Operators and Differentiability*, second ed., Lecture Notes in Mathematics 1364, Springer-Verlag, Berlin, 1993.

[Roc76] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.

[STF86] A. G. Sukharev, A. V. Timokhov, and V. V. Fedorov, *A Course on Optimization Methods*, Nauka, Moscow, 1986 (in Russian).

[Tse92] P. Tseng, *On the convergence of the products of firmly nonexpansive maps*, SIAM J. Optim., 2 (1992), pp. 425–434.

SIAM J. OPTIM.
Vol. 7, No. 4, pp. 1103–1137, November 1997

ON GENERIC ONE-PARAMETRIC SEMI-INFINITE OPTIMIZATION [*]

H. TH. JONGEN[†] AND O. STEIN[‡]

Abstract. We consider differentiable semi-infinite optimization problems depending on a real parameter. For generic one-parametric families we classify the corresponding set of generalized critical points into eight types. Five of these types also occur in problems with a finite number of inequality constraints, whereas the other three types are typical for the semi-infinite case. We discuss types 7 and 8 in detail. While at points of type 6, the singularity is due to the fact that in the associated lower level problem a Lagrange multiplier corresponding to an active inequality constraint vanishes, at points of type 7 and 8, the gradients of the active constraints in the lower level problem are linearly dependent. If the total number of active constraints in the lower level problem does not exceed the lower level dimension, the point is of type 7; otherwise, it is of type 8. Moreover, we distinguish between points of type 8a and 8b, where a point is of type 8a if the Mangasarian–Fromovitz constraint qualification holds in the lower level problem, and of type 8b otherwise. At points of type 8a, the set of generalized critical points is not smooth, but it does not exhibit a turning point. The linear and quadratic indices remain constant when passing along a point of type 8a. Points of type 7 and type 8b are (relative) boundary points of the set of generalized critical points.

Key words. parametric optimization, semi-infinite optimization, generalized critical point, critical point, singularity, index, jump

AMS subject classifications. 90C31, 90C34

PII. S1052623495289094

1. Introduction.

In this paper we study the solution sets of semi-infinite optimization problems depending on a real parameter. The description of the problems is as follows:

$$SIP(t) \qquad \text{minimize } f(\cdot,t) \text{ on the feasible set } M(t),$$

where

$$\begin{aligned} M(t) &= \{x \in \mathbb{R}^n \mid h^i(x,t)=0,\ i\in I,\ g(x,t,y)\ge 0,\ y\in Y(t)\},\\ Y(t) &= \{y\in\mathbb{R}^m \mid u^k(t,y)=0,\ k\in K,\ v^l(t,y)\ge 0,\ l\in L\},\end{aligned}$$

and $t \in \mathbb{R}$.

The defining functions $f, h^i : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$, $g : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}$, and u^k, v^l : $\mathbb{R} \times \mathbb{R}^m \to \mathbb{R}$ are supposed to be r-times continuously differentiable, $r \ge 1$ to be specified later on ($r \ge 3$ will always be sufficient). Moreover, the cardinalities of the index sets I, K, L satisfy the inequalities $|I| < n$, $|K| < m$, and $|L| < \infty$.

For a fixed value $t = \bar{t}$ of the parameter, $SIP(\bar{t})$ is a semi-infinite problem because the feasible set $M(\bar{t})$ is described by, in general, infinitely many constraints according to infinite index sets $Y(\bar{t})$.

[*] Received by the editors July 14, 1995; accepted for publication (in revised form) July 7, 1996. http://www.siam.org/journals/siopt/7-4/28909.html

[†] RWTH Aachen, Department of Mathematics, D-52056 Aachen, Germany (jongen@rwth-aachen.de).

[‡] RWTH Aachen, Department of Mathematics, D-52056 Aachen, Germany (stein@mathc.rwth-aachen.de). This author was supported by the “Deutsche Forschungsgemeinschaft” through the Graduiertenkolleg “Mathematische Optimierung” at the University of Trier.

For an introduction to semi-infinite programming (SIP) problems we refer to the extensive survey by R. Hettich and K. O. Kortanek [7]. The present research about the generic structure of the solution set related to $SIP(t)$ is based on several preliminary studies. Fundamental results concerning *finite* one-parametric optimization problems are due to M. Kojima and R. Hirabayashi [18], H. Th. Jongen, P. Jonker, and F. Twilt [13, 14], as well as to A. B. Poore and C. A. Tiahrt [20, 24]. The generic local structure of *parameter-free* semi-infinite optimization problems has been studied by H. Th. Jongen and G. Zwier [17, 25]. A stability analysis for the feasible set of one-parametric semi-infinite optimization problems has been performed by H. Th. Jongen, J.-J. Rückmann, and G.-W. Weber [15], while results about the solution set of $SIP(t)$ in case that the index set $Y(t)$ is endowed with a special structure have been derived by T. Rupp [21, 22] and R. Hettich, H. Th. Jongen, and O. Stein [6].

Throughout the paper we make the following assumptions on the index set $Y(t)$.

Assumption 1. The set $Y(t) \subset \mathbb{R}^m$ is compact for all $t \in \mathbb{R}$.

Assumption 2. The set-valued mapping $t \to Y(t)$ is upper semicontinuous at each $t \in \mathbb{R}$.

Note that Assumptions 1 and 2 coincide with Berge's notion of upper semicontinuity for the mapping Y on $\mathbb{R}$ (cf. [1]).

For a given point $(x,t) \in \mathbb{R}^{n+1}$, we define the set of active inequality constraints by

$$Y_0(x,t) = \{y \in Y(t) \mid g(x,t,y) = 0\}.$$

Note that $Y_0(x,t)$ is compact for all $(x,t) \in \mathbb{R}^{n+1}$ by Assumption 1 and by the continuity of $g(x,t,\cdot)$. Of course, $Y_0(x,t)$ need not consist of isolated points.

If $F, G \in C^1$ we denote by $F_x(x,y)$ (column vector) the partial derivative of F and by $\frac{d}{dx}G(x)$ the total derivative of G with respect to x.

Definition 1.1. *A point $\bar{x} \in M(\bar{t})$ is called a* generalized critical point *(in short, g.c. point) for $SIP(\bar{t})$ if the set of vectors*

$$\{f_x(\bar{x},\bar{t}),\ h_x^i(\bar{x},\bar{t}),\ i \in I,\ g_x(\bar{x},\bar{t},y),\ y \in Y_0(\bar{x},\bar{t})\}$$

is linearly dependent.

In case of a g.c. point, there exist a finite subset $\{\bar{y}^1,\ldots,\bar{y}^s\} \subset Y_0(\bar{x},\bar{t})$ ($s \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$) and real numbers $\kappa,\ \lambda_i,\ i \in I,\ \mu_j,\ j \in \{1,\ldots,s\}$, not all vanishing, such that

$$\kappa f_x(\bar{x},\bar{t}) = \sum_{i\in I} \lambda_i h_x^i(\bar{x},\bar{t}) + \sum_{j=1}^{s} \mu_j g_x(\bar{x},\bar{t},\bar{y}^j). \tag{1}$$

Hence, Definition 1.1 relaxes the first-order necessary condition for a point $\bar{x}$ to be a local minimum for $SIP(\bar{t})$ (cf. [5, 8]), where the multipliers κ and μ_j are assumed to be nonnegative.

Definition 1.2. *A point $\bar{x} \in M(\bar{t})$ is called a* stationary point *for $SIP(\bar{t})$ if equation* (1) *holds with $\kappa = 1$ and $\mu_j \geq 0,\ j \in \{1,\ldots,s\}$.*

Definition 1.3. *The* g.c. point set $\Sigma \subset \mathbb{R}^{n+1}$ *is defined to be the set*

$$\Sigma \ = \ \left\{(x,t) \in \mathbb{R}^{n+1} \;\middle|\; x \in M(t) \ \textit{ is a g.c. point for } SIP(t)\right\} .$$

In the case where the index set $Y(t)$ is a constant finite set, i.e., the standard case of finitely many inequality constraints, the generic structure of the set Σ is completely characterized in [13, 14]. In particular, it turns out that each point of Σ belongs to one of five specific types. However, in the semi-infinite case, three additional types are coming into play.

Let the linear space of real valued C^r-functions on $\mathbb{R}^N$, which is denoted by $C^r(\mathbb{R}^N,\mathbb{R})$, be topologized by means of the (strong or Whitney) C^r_s-topology (cf. [9, 12]). A typical base neighborhood W^r_ε of the zero function is generated by means of a continuous positive function $\varepsilon : \mathbb{R}^N \longrightarrow \mathbb{R}$:

$$W^r_\varepsilon = \left\{\psi \in C^r(\mathbb{R}^N,\mathbb{R}) \,\middle|\, |\partial^\alpha \psi(y)| < \varepsilon(y) \text{ for all } y \in \mathbb{R}^N, \text{ for all } |\alpha| \le r \right\}.$$

A typical C^r_s-neighborhood of $F \in C^r(\mathbb{R}^N,\mathbb{R})$ has the form $F + W^r_\varepsilon$. The C^r_s-topology for a finite product of spaces is defined by the corresponding product topology.

Definition 1.4. *The set CUSC (compact upper semicontinuous) is defined as the following subset of $C^3(\mathbb{R}^{m+1},\mathbb{R})^{|K|+|L|}$: $(\ \dots,\ u^k,\ \dots,\ v^l,\ \dots)$ belongs to CUSC if and only if Assumptions* 1 *and* 2 *are satisfied.*

Theorem 1.5. *There exists a C^3_s-open dense subset*

$$\mathcal{F} \ \subset\ C^3(\mathbb{R}^{n+1},\mathbb{R})^{|I|+1} \times C^3(\mathbb{R}^{n+m+1},\mathbb{R}) \times CUSC$$

such that for $(f,\ \dots,\ h^i,\ \dots,\ g,\ \dots,\ u^k,\ \dots,\ v^l,\ \dots) \in \mathcal{F}$ we have the following: each point of the corresponding generalized critical point set Σ is one of eight types.

It is the aim of this paper to give a complete classification of the eight types that generically appear in the semi-infinite case, and to present results concerning the local structure of Σ at points of each type. Thereby, we will enhance a result of our previous paper [6] which focussed on points of type 6.

The paper is organized as follows. In section 2 we discuss the case that $SIP(t)$ can locally be reduced to a one-parametric finite optimization problem, and thus, the above mentioned types 1–5 generically occur. Section 3 deals with generic violations of this reduction approach which lead to the new types 6–8. In section 4, we sketch the genericity part of the proof of Theorem 1.5, and section 5 contains remarks about jumps at singular points of type 6–8 and about certain generalizations of our concept.

2. The reducible case.

Since for given $(\bar{x},\bar{t}) \in \mathbb{R}^{n+1}$ with $\bar{x} \in M(\bar{t})$, any $y \in Y_0(\bar{x},\bar{t})$ is a global minimum of $g(\bar{x},\bar{t},\cdot)$ on $Y(\bar{t})$, the elements of $Y_0(\bar{x},\bar{t})$ are solutions of the finite multiparametric optimization problem

$$Q(x,t) \qquad \text{minimize } \ g(x,t,\cdot) \ \text{ on the feasible set } Y(t)$$

at the parameter value $(x,t) = (\bar{x},\bar{t})$. By this observation, $SIP(t)$ is a two-level optimization problem, where the upper level consists of optimizing the objective function f, whereas the lower level is concerned with the corresponding active index set Y_0 of inequality constraints.

In order to exploit the finite structure of the lower level problem $Q(x,t)$, as well as to describe the locally reduced finite upper level problem (cf. section 2.2), we recall some definitions and results for finite parametric optimization problems.

2.1. Finite optimization problems.

Consider the problem

$$P(\tau) \qquad \text{minimize } \ F(\cdot,\tau) \ \text{ on the feasible set } \mathcal{M}(\tau)\ ,$$

where

$$\mathcal{M}(\tau) \ = \ \{z \in \mathbb{R}^n \ \big|\ H^i(z,\tau) = 0\ ,\ i \in \mathcal{I},\ G^j(z,\tau) \ge 0\ ,\ j \in \mathcal{J}\}$$

and $\tau \in \mathbb{R}^k$. The functions $F, H^i, G^j : \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}$ are assumed to be $C^r,\ r \geq 2$, and the cardinalities of $\mathcal{I}, \mathcal{J}$ satisfy $|\mathcal{I}| < n,\ |\mathcal{J}| < \infty$.

In the following, we make use of two constraint qualifications.

DEFINITION 2.1. *Let $\bar{\tau} \in \mathbb{R}^k$ be fixed. The* linear independence constraint qualification *(in short, LICQ) is said to hold at $\bar{z} \in \mathcal{M}(\bar{\tau})$ if the set of vectors*

$$\{H^i_z(\bar{z}, \bar{\tau}),\ i \in \mathcal{I},\ G^j_z(\bar{z}, \bar{\tau}),\ j \in \mathcal{J}_0(\bar{z}, \bar{\tau})\}$$

is linearly independent; here, $\mathcal{J}_0(\bar{z}, \bar{\tau})$ denotes the index set of active inequality constraints.

Mangasarian–Fromovitz constraint qualification *(in short, MFCQ) is said to hold at $\bar{z} \in M(\bar{\tau})$ if both the set of vectors $\{H^i_z(\bar{z}, \bar{\tau})\ ,\ i \in \mathcal{I}\}$ is linearly independent and if there exists a vector $\xi \in \mathbb{R}^n$ satisfying*

$$\begin{aligned} \xi^\top H^i_z(\bar{z}, \bar{\tau}) &= 0\ ,\ i \in \mathcal{I}, \\ \xi^\top G^j_z(\bar{z}, \bar{\tau}) &> 0\ ,\ j \in \mathcal{J}_0(\bar{z}, \bar{\tau}). \end{aligned}$$

It is well known that LICQ implies MFCQ; i.e., MFCQ is a weaker constraint qualification.

DEFINITION 2.2. *For $\bar{z} \in \mathcal{M}(\bar{\tau})$ let the Lagrange function $\mathcal{L}$ be defined as follows:*

$$\mathcal{L}^{(\bar{z},\bar{\tau})}(z, \lambda, \mu, \tau) = F(z, \tau) - \sum_{i \in \mathcal{I}} \lambda_i H^i(z, \tau) - \sum_{j \in \mathcal{J}_0(\bar{z}, \bar{\tau})} \mu_j G^j(z, \tau)\ .$$

The point $\bar{z}$ is called a critical point *for $F(\cdot, \bar{\tau})|_{\mathcal{M}(\bar{\tau})}$ if LICQ holds and if there exist real numbers $\bar{\lambda}_i,\ i \in \mathcal{I},\ \bar{\mu}_j,\ j \in \mathcal{J}_0(\bar{z}, \bar{\tau})$ (called Lagrange multipliers) satisfying*

$$\mathcal{L}^{(\bar{z},\bar{\tau})}_z(\bar{z}, \bar{\lambda}, \bar{\mu}, \bar{\tau}) = 0. \tag{2}$$

Note that, by LICQ, the Lagrange multipliers of a critical point are uniquely determined.

DEFINITION 2.3. *Let $\bar{z} \in M(\bar{\tau})$ be a critical point with Lagrange multipliers $\bar{\lambda}_i,\ i \in \mathcal{I},\ \bar{\mu}_j,\ j \in \mathcal{J}_0(\bar{z}, \bar{\tau})$. Then, $\bar{z}$ is called* nondegenerate *if the following conditions hold:*

$$\begin{aligned} &ND1: \quad \bar{\mu}_j \neq 0,\ j \in \mathcal{J}_0(\bar{z}, \bar{\tau}), \\ &ND2: \quad \mathcal{L}^{(\bar{z},\bar{\tau})}_{zz}(\bar{z}, \bar{\lambda}, \bar{\mu}, \bar{\tau})\ |_{T_{\bar{z}}\mathcal{M}(\bar{\tau})} \ \text{ is nonsingular.} \end{aligned}$$

A critical point is called a nondegenerate local mimimum *if it is both a nondegenerate critical point and a local minimum.*

In Definition 2.3, the symbol $T_{\bar{z}}\mathcal{M}(\bar{\tau})$ stands for the tangent space of $\mathcal{M}(\bar{\tau})$ at $\bar{z}$, i.e.,

$$T_{\bar{z}}\mathcal{M}(\bar{\tau}) = \left\{ \xi \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \xi^\top H^i_z(\bar{z}, \bar{\tau}) = 0,\ i \in \mathcal{I}, \\ \xi^\top G^j_z(\bar{z}, \bar{\tau}) = 0,\ j \in \mathcal{J}_0(\bar{z}, \bar{\tau}) \end{array} \right\}\ .$$

Moreover, $\mathcal{L}_{zz}$ denotes the matrix of second-order partial derivatives (Hessian) with respect to z. Finally, $\mathcal{L}_{zz}|_{T_{\bar{z}}\mathcal{M}(\bar{\tau})}$ stands for the matrix $V^\top \mathcal{L}_{zz} V$, where V may be any matrix of n-vectors which form a basis for the tangent space $T_{\bar{z}}\mathcal{M}(\bar{\tau})$.

Definition 2.4. *Let $\bar{z} \in \mathcal{M}(\bar{\tau})$ be a nondegenerate critical point. The* linear index/linear coindex *(LI/LCI) of $\bar{z}$ is defined to be the number of negative/positive numbers $\bar{\mu}_j$ in ND1 (cf. Definition 2.3). The* quadratic index/quadratic coindex *(QI,QCI) of $\bar{z}$ is defined to be the number of negative/positive eigenvalues of $\mathcal{L}_{zz}^{(\bar{z},\bar{\tau})}(\bar{z},\bar{\lambda},\bar{\mu},\bar{\tau})|_{T_{\bar{z}}\mathcal{M}(\bar{\tau})}$ in ND2.*

Note that the numbers QI and QCI are independent of the incidental choice of the matrix V (by Sylvester's law of inertia). The indices LI, LCI, QI, QCI completely determine the local behavior of the objective function $F(\cdot,\tau)$ on the feasible set $\mathcal{M}(\tau)$ (cf. [11]). In particular, a nondegenerate critical point is a local minimum if and only if LI+QI=0. For convenience we will refer to the number of negative and positive eigenvalues of a symmetric matrix A by QI(A) and QCI(A), resp.

By defining $\zeta = (z,\lambda,\mu)$, the Karush–Kuhn–Tucker equations for a critical point $\bar{z} \in \mathcal{M}(\bar{\tau})$ read

$$\mathcal{L}_{\zeta}^{(\bar{z},\bar{\tau})}(\bar{\zeta},\bar{\tau}) \;=\; 0.$$

Since for a nondegenerate critical point it is easily shown that the matrix $\mathcal{L}_{\zeta\zeta}^{(\bar{z},\bar{\tau})}(\bar{\zeta},\bar{\tau})$ is nonsingular, the implicit function theorem yields the existence of a locally unique C^{r-1}-function ζ satisfying $\zeta(\bar{\tau}) = \bar{\zeta}$ and

$$\mathcal{L}_{\zeta}^{(\bar{z},\bar{\tau})}(\zeta(\tau),\tau) \;\equiv\; 0\ . \tag{3}$$

Hence, in a neighborhood of $\bar{\tau}$ we may define the marginal function

$$\Phi(\tau) = F(z(\tau),\tau)\ .$$

Equation (3) immediately yields the following well-known lemma.

Lemma 2.5. *Φ is of differentiability class C^r, and locally around $\bar{\tau}$ we have*

$$\tfrac{d}{d\tau}\,\Phi(\tau) \;\equiv\; \mathcal{L}_{\tau}^{(\bar{z},\bar{\tau})}(\zeta(\tau),\tau)\ .$$

2.2. The reduced upper level problem.

In this section, we study generalized critical points $\bar{x}$ of $SIP(\bar{t})$ with the additional property that all elements of $Y_0(\bar{x},\bar{t})$ are *nondegenerate* global minima of $Q(\bar{x},\bar{t})$. Then, $Y_0(\bar{x},\bar{t})$ is a discrete set. Since $Y_0(\bar{x},\bar{t})$ is also a closed subset of the compact set $Y(\bar{t})$ (cf. Assumption 1), it follows that $Y_0(\bar{x},\bar{t})$ is a finite set, say $Y_0(\bar{x},\bar{t}) = \{\bar{y}^j,\ j \in J\}$, where $J = \{1,\dots,s\}$ and $s \in \mathbb{N}_0$.

Now we treat (x,t) as parameters and apply the implicit function theorem as in section 2.1. In this way we obtain for (x,t) near $(\bar{x},\bar{t})$ and for any $j \in J$ a nondegenerate local minimum $y^j(x,t)$ of $Q(x,t)$, where the function $y^j(x,t)$ depends C^{r-1} on the parameters (x,t). Thus, we may introduce the (locally defined) marginal functions $\varphi^j(x,t) = g(x,t,\ y^j(x,t))$, $j \in J$. Using the corresponding Lagrangians

$$\begin{aligned}\mathcal{L}^{(\bar{x},\bar{t},\bar{y}^j)}(x,t,y^j,\beta^j,\gamma^j) = g(x,t,y^j)\ &- \sum_{k\in K}\beta_k^j u^k(t,y^j)\\ &- \sum_{l\in L_0(\bar{x},\bar{t},\bar{y}^j)}\gamma_l^j v^l(t,y^j)\ ,\quad j \in J,\end{aligned}$$

we obtain from Lemma 2.5 the following.

Corollary 2.6. *The marginal functions φ^j, $j \in J$, are of differentiability class C^r, and locally around $(\bar{x}, \bar{t})$ we have*

$$\varphi^j_x(x,t) \;\equiv\; g_x(x,t,y^j(x,t)).$$

The next lemma is a straight generalization of the "reduction lemma" in [8]. Here, we need both Assumption 1 and 2.

Lemma 2.7. *Let $(\bar{x}, \bar{t}) \in \Sigma$ for $SIP(t)$ and let all elements of $Y_0(\bar{x}, \bar{t})$ be nondegenerate global minima of $Q(\bar{x}, \bar{t})$. Define the unfolded feasible set $Z = \{M(t) \times \{t\} \mid t \in \mathbb{R}\}$ as well as the set*

$$M^{(\bar{x},\bar{t})}(t) \;=\; \left\{x \in \mathbb{R}^n \mid h^i(x,t) = 0,\ i \in I,\ \varphi^j(x,t) \geq 0,\ j \in J\right\}$$

and its unfolding $Z^{(\bar{x},\bar{t})} = \{M^{(\bar{x},\bar{t})}(t) \times \{t\} \mid t \in \mathbb{R}\}$. Then, there exists a neighborhood U of $(\bar{x}, \bar{t})$ with $Z \cap U \;=\; Z^{(\bar{x},\bar{t})} \cap U$.

By Lemma 2.7 we obtain a local reduction of the semi-infinite optimization problem to an optimization problem with finitely many constraints, namely,

$$P^{(\bar{x},\bar{t})}(t) \qquad \text{minimize } f(\cdot,t) \text{ on the feasible set } M^{(\bar{x},\bar{t})}(t)\ .$$

For finite one-parametric optimization problems $P^{(\bar{x},\bar{t})}(t)$ the generic structure of Σ (which is defined as in Definition 1.3, with the obvious specifications) has been studied in [13, 14], where all defining functions are supposed to be C^3. We emphasize that continuous derivatives of second order are sufficient for all situations under consideration apart from singular points of type 3 (cf. Definition 2.11), where we locally need third-order derivatives in order to treat the vanishing eigenvalue of an associated Hessian. Therefore we require the following.

Assumption 3. $SIP(t)$ is defined by C^3-functions.

By Assumption 3 and Corollary 2.6, all defining functions of $P^{(\bar{x},\bar{t})}(t)$ are of differentiability class C^3, and hence, we can reformulate the definitions and results from [13, 14] for the locally reduced upper level problem $P^{(\bar{x},\bar{t})}(t)$. Thereby we obtain that, generically, each point of Σ belongs to one of precisely five different types. The remainder of this section deals with the definitions of these types and their related characteristic numbers as well as with short descriptions of Σ in a neighborhood of each type. Note that these definitions and results are immediate consequences of [13, 14] and, being familiar with these works, the reader may proceed with section 3. However, in the remainder of this paper we will refer to the following facts frequently and in detail.

Recall that, by Corollary 2.6, we have

$$\begin{aligned} &\varphi^j_x(\bar{x},\bar{t}) = g_x(\bar{x},\bar{t},\bar{y}^j), \\ &\varphi^j_{xt}(\bar{x},\bar{t}) = g_{xt}(\bar{x},\bar{t},\bar{y}^j) \;+\; g_{xy}(\bar{x},\bar{t},\bar{y}^j) \cdot y^j_t(\bar{x},\bar{t}), \\ \text{and}\quad &\varphi^j_{xx}(\bar{x},\bar{t}) = g_{xx}(\bar{x},\bar{t},\bar{y}^j) \;+\; g_{xy}(\bar{x},\bar{t},\bar{y}^j) \cdot y^j_x(\bar{x},\bar{t}). \end{aligned}$$

Also note that, by the definitions of φ^j and J, *all* indices in J correspond to inequalities being active at $\bar{z}$, i.e., $J_0(\bar{z}) = J = \{1, \ldots, s\}$. Furthermore, we let $I = \{1, \ldots, q\}$ with $q \in \mathbb{N}_0$.

Definition 2.8. *Let Γ be a one-dimensional manifold in $\mathbb{R}^{n+1}$ and $\bar{z} = (\bar{x}, \bar{t}) \in \Gamma$. If the function $\Phi(x,t) \equiv t$, restricted to Γ, possesses a local extremum at $\bar{z}$,*

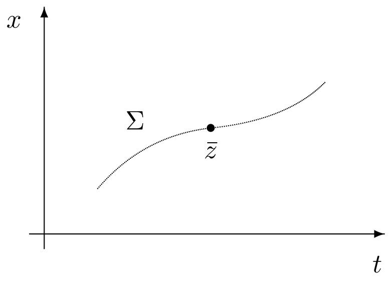


Fig. 2.1. *A g.c. point of type* 1.

we call $\bar{z}$ *a* turning point *with respect to* t *for* Γ. *If, additionally,* Γ *is locally a* C^2*-manifold and the extremum of* $\Phi(x,t)$ *is nondegenerate, we call* $\bar{z}$ *a* quadratic turning point *with respect to* t.

Note that in case of a quadratic turning point $\bar{z}$, the set Γ can be approximated by means of a parabola in a neighborhood of $\bar{z}$.

Definition 2.9 (type 1). *A point* $\bar{z} = (\bar{x}, \bar{t}) \in \Sigma$ *is of* type 1 *if the following conditions* (1.1) *and* (1.2) *are fulfilled.*

(1.1) *All elements of* $Y_0(\bar{x}, \bar{t})$ *are nondegenerate global minima for* $Q(\bar{x}, \bar{t})$.

(1.2) $\bar{x}$ *is a nondegenerate critical point for* $P^{(\bar{x},\bar{t})}(\bar{t})$.

Characteristic numbers: LI, LCI, QI, QCI (cf. Definition 2.4*).*

If $\bar{z}$ is of type 1, the set Σ can be parametrized by means of the parameter t in a neighborhood of $\bar{z}$ (compare Figure 2.1). Since f, h^i, φ^j are C^3-functions, Σ is a C^2-manifold around $\bar{z}$. Locally, the indices LI, LCI, QI, and QCI remain constant along Σ.

Definition 2.10 (type 2). *A point* $\bar{z} = (\bar{x}, \bar{t}) \in \Sigma$ *is of* type 2 *if the following conditions* (2.1)–(2.7) *are fulfilled.*

(2.1) *All elements of* $Y_0(\bar{x}, \bar{t})$ *are nondegenerate global minima for* $Q(\bar{x}, \bar{t})$.

(2.2) $\bar{x}$ *is a critical point for* $P^{(\bar{x},\bar{t})}(\bar{t})$.

(2.3) $s > 0$.

By condition (2.2) *and Definition* 2.2 *we have*

$$f_x(\bar{z}) \;=\; \sum_{i=1}^{q} \bar{\lambda}_i h_x^i(\bar{z}) \;+\; \sum_{j=1}^{s} \bar{\mu}_j \varphi_x^j(\bar{z}). \tag{4}$$

(2.4) *In* (4), *exactly one of the Lagrange multipliers* $\bar{\mu}_j$ *vanishes.*

After renumbering, we assume that $\bar{\mu}_s = 0$. *Put*

$$T = \bigcap_{i \in I} \mathrm{Ker}\ (h_x^i(\bar{z}))^\top \;\cap \bigcap_{j \in J} \mathrm{Ker}\ (\varphi_x^j(\bar{z}))^\top, \tag{5}$$

$$\tilde{T} = \bigcap_{i \in I} \mathrm{Ker}\ (h_x^i(\bar{z}))^\top \;\cap \bigcap_{j \in J \setminus \{s\}} \mathrm{Ker}\ (\varphi_x^j(\bar{z}))^\top,$$

$$L(z) = f(z) \;-\; \sum_{i=1}^{q} \bar{\lambda}_i h^i(z) \;-\; \sum_{j=1}^{s} \bar{\mu}_j \varphi^j(z), \tag{6}$$

where $\mathrm{Ker}(A)$ *denotes the zero space of the matrix* A.

(2.5) $L_{xx}(\bar{z})|_T$ *is nonsingular.*

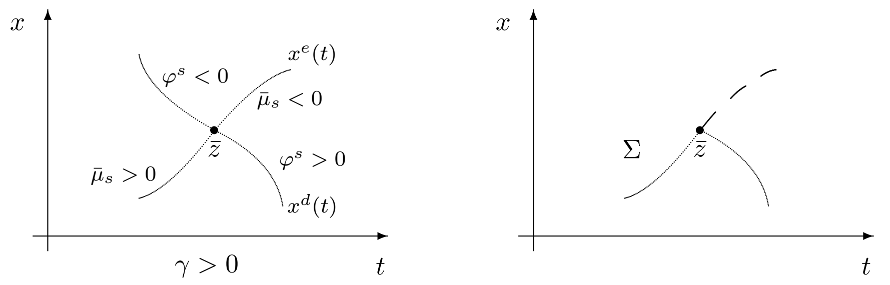


FIG. 2.2. *A g.c. point of type* 2.

(2.6) *$L_{xx}(\bar{z})|_{\tilde{T}}$ is nonsingular.*
Let B be an $n \times r$ matrix of rank r. By $B^\dagger$ we denote the matrix $(B^\top B)^{-1}B^\top$. In fact, $B^\dagger$ is the Moore–Penrose inverse of B. Now, let W be a basis matrix for the linear space $\tilde{T}$, put $\psi = (h^1, \ldots, h^q, \varphi^1, \ldots, \varphi^{s-1})$, and define

$$\begin{aligned} \alpha &= -((\psi_x)^\dagger)^\top \cdot (\psi_t)^\top, \\ \beta &= -W(W^\top L_{xx} W)^{-1} W^\top \left(L_{xx}\ \alpha + L_{xt}\right), \\ \gamma &= (\varphi^s_x)^\top(\alpha+\beta)\ +\ \varphi^s_t, \end{aligned}$$

all partial derivatives being evaluated at $\bar{z}$.
(2.7) $\gamma \neq 0$.
We put $\delta = QI(L_{xx}(\bar{z})|_{\tilde{T}}) - QI(L_{xx}(\bar{z})|_T)$ and obtain the characteristic numbers sign(γ) and δ.

A point of type 2 is a degenerate critical point; however, only the strict complementary condition (ND1 in Definition 2.3) is violated. Let $P_e^{(\bar{x},\bar{t})}(t)$ (resp., $P_d^{(\bar{x},\bar{t})}(t)$) denote the parametric optimization problem which differs from $P^{(\bar{x},\bar{t})}(t)$ in the sense that the *inequality* constraint φ^s is turned into an *equality* constraint (resp., *deleted* as a constraint). Then, $\bar{x}$ is a nondegenerate critical point both for $P_e^{(\bar{x},\bar{t})}(\bar{t})$ and $P_d^{(\bar{x},\bar{t})}(\bar{t})$. As a consequence, the set Σ is (locally) the union of the two C^2-curves $t \longmapsto (x^e(t), t)$ and $t \longmapsto (x^d(t), t)$ as far as they are feasible points; here, $x^e(t)$ and $x^d(t)$ are the critical points near $\bar{x}$ for $P_e^{(\bar{x},\bar{t})}(t)$ and $P_d^{(\bar{x},\bar{t})}(t)$, respectively. It can be shown that $x_t^d(\bar{t}) = \alpha + \beta$ and hence, if we follow the points $(x^d(t), t)$ for increasing t, we enter (leave) the feasible set $M^{(\bar{x},\bar{t})}(t)$ according to sign$(\gamma) = +1$ (-1); see Figure 2.2 for a typical situation, where the labels "$x^e(t)$", "$x^d(t)$" denote the graphs of x^e and x^d, respectively. The part of Σ consisting of *nonstationary* points is represented by a dashed curve.

DEFINITION 2.11 (type 3). *A point $\bar{z} = (\bar{x}, \bar{t}) \in \Sigma$ is of* type 3 *if the following conditions* (3.1)–(3.5) *are fulfilled.*
(3.1) *All elements of $Y_0(\bar{x}, \bar{t})$ are nondegenerate global minima for $Q(\bar{x}, \bar{t})$.*
(3.2) *$\bar{x}$ is a critical point for $P^{(\bar{x},\bar{t})}(\bar{t})$.*
By condition (3.2)*, the critical point relation* (4) *holds.*
(3.3) *In* (4)*, we have $\bar{\mu}_j \neq 0,\ j \in J$.*
Let the Lagrange function L be defined as in (6) *and let the tangent space T be as in* (5)*.*
(3.4) *Exactly one eigenvalue of $L_{xx}(\bar{z})|_T$ vanishes.*

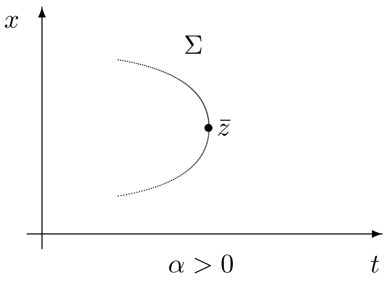


FIG. 2.3. *A g.c. point of type* 3.

Let V be a basis matrix for the tangent space T. According to (3.4)*, let w be a nonvanishing vector such that $V^\top L_{xx}(\bar{z})Vw = 0$, and put $v = Vw$. Now let $\psi = (h^1, \ldots, h^q, \varphi^1, \ldots, \varphi^s)$ and define (the symbol † again denotes the Moore-Penrose inverse):*

$$\alpha_1 = L_{xxx}(v,v,v) \; - \; 3v^\top L_{xx} \cdot ((\psi_x)^\dagger)^\top (v^\top \psi_{xx} v), \tag{7}$$

$$\alpha_2 = L_{xt}^\top v \; - \; \psi_t (\psi_x)^\dagger L_{xx} v, \tag{8}$$

where

$$L_{xxx}(v,v,v) = \sum_{i,j,k=1}^{n} \tfrac{\partial^3}{\partial x_i \partial x_j \partial x_k} L \cdot v_i v_j v_k,$$
$$v^\top \psi_{xx} v = (v^\top h_{xx}^1 v, \ldots, v^\top \varphi_{xx}^s v)^\top,$$

all partial derivatives being evaluated at $\bar{z}$. In the case that $I = J = \emptyset$, we have $T = \mathbb{R}^n$ and we omit all entries of ψ in (7) *and* (8)*. Next, we define $\alpha = \alpha_1 \cdot \alpha_2$.*

(3.5) $\alpha \neq 0$.

We put $\beta = QI(L_{xx}(\bar{z})|_T)$ and obtain the characteristic numbers sign(α) *and β.*

A point of type 3 is a degenerate critical point; however, only condition ND2 in Definition 2.3 is violated. In a neighborhood of $\bar{z}$ the index set of active inequality constraints for points on Σ remains constant (hence, equal to J). Locally around $\bar{z}$ the set Σ is a one-dimensional C^2-manifold, and the function $\Phi(x,t) \equiv t$, restricted to Σ, has a nondegenerate local maximum (minimum) at $\bar{z}$ according to sign(α) = +1 (−1). Consequently, the set Σ has a quadratic turning point at $\bar{z}$. In view of condition (3.3), the indices LI and LCI do not change when passing the point $\bar{z}$ along Σ. However, the quadratic index QI changes from β to $\beta + 1$, or vice versa. A typical situation for a g.c. point of type 3 is sketched in Figure 2.3.

DEFINITION 2.12 (type 4). *A point $\bar{z} = (\bar{x}, \bar{t}) \in \Sigma$ is of* type 4 *if the following conditions* (4.1)–(4.7) *are fulfilled.*

(4.1) *All elements of $Y_0(\bar{x}, \bar{t})$ are nondegenerate global minima for $Q(\bar{x}, \bar{t})$.*

(4.2) $q + s > 0$.

(4.3) $q + s - 1 < n$.

Define the $n \times (q+s)$ matrix $M = \big(h_x^1(\bar{z}), \ldots, h_x^q(\bar{z}), \varphi_x^1(\bar{z}), \ldots, \varphi_x^s(\bar{z})\big)$.

(4.4) rank(M) $= q + s - 1$.

From condition (4.4) *we see that* Ker(M) *is a one-dimensional space. Let $(\bar{\lambda}, \bar{\mu})$ be a*

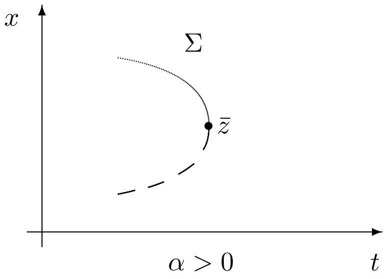


FIG. 2.4. *A g.c. point of type* 4.

generator for $\mathrm{Ker}(M)$ *and define*

$$L(z) \;=\; \sum_{i=1}^{q} \bar{\lambda}_i h^i(z) \;+\; \sum_{j=1}^{s} \bar{\mu}_j \varphi^j(z)$$

and $T = \mathrm{Ker}(M^\top)$. *Furthermore, let* W *be a basis matrix for* T *and define*

$$A \;=\; L_t(\bar{z})\; W^\top L_{xx}(\bar{z}) W.$$

(4.5) *A is nonsingular.*
Finally, let

$$\alpha \;=\; f_x^\top(\bar{z}) W A^{-1} W^\top f_x(\bar{z}).$$

We note that α is independent of the choice of the matrix W.
(4.6) $\alpha \neq 0$.
In the case $s > 0$ we require, additionally,
(4.7) $\bar{\mu}_j \neq 0,\ j \in J$,
and we normalize the $\bar{\mu}_j$'s by setting $\bar{\mu}_s = 1$. Let γ be the number of negative $\bar{\mu}_j$, $j \in \{1, \ldots, s-1\}$, and put $\delta = L_t(\bar{z})$, $\beta = QCI(A)$. Then, we have the characteristic numbers sign(α), β as well as (corresponding to $\bar{\mu}_s = 1$) γ and sign(δ).

For specific details about this type we refer to [13] and [14]. Here, we only mention that locally around $\bar{z}$ the set Σ is a one-dimensional C^2-manifold and the function $\Phi(x,t) \equiv t$, restricted to Σ, has a nondegenerate local maximum (minimum) at $\bar{z}$ according to $\mathrm{sign}(\alpha) = +1$ (-1). Consequently, the set Σ has a quadratic turning point at $\bar{z}$. When passing the point $\bar{z}$ along Σ, the linear index LI changes from γ to $s-\gamma$, and the quadratic index QI changes from $\beta - 1$ to $n-q-s-\beta+1$ or from β to $n-q-s-\beta$ (or vice versa), according to the values of sign(α) and sign(δ). See Figure 2.4 for an example with $J \neq \emptyset$, where the dashed part of Σ stands for nonstationary points.

DEFINITION 2.13 (type 5). *A point $\bar{z} = (\bar{x}, \bar{t}) \in \Sigma$ is of* type 5 *if the following conditions* (5.1)–(5.5) *are fulfilled.*
(5.1) *All elements of $Y_0(\bar{x}, \bar{t})$ are nondegenerate global minima for $Q(\bar{x}, \bar{t})$.*
(5.2) $q + s = n + 1$.
(5.3) *The set $\{h_z^i(\bar{z}),\ i \in I,\ \varphi_z^j(\bar{z}),\ j \in J\}$ is linearly independent.*
Since we assume $q = |I| < n$ throughout, condition (5.2) *implies that $s \geq 2$. From conditions* (5.2) *and* (5.3) *we see that there exist $\lambda_i,\ i \in I$, and $\mu_j,\ j \in J$, not all*

vanishing (and unique up to a common multiple) such that

$$\sum_{i=1}^{q} \lambda_i h_x^i(\bar{z}) \ + \ \sum_{j=1}^{s} \mu_j \varphi_x^j(\bar{z}) \ = \ 0. \tag{9}$$

(5.4) *In* (9) *we have* $\mu_j \neq 0, \ j \in J$.
From conditions (5.2) *and* (5.3) *it follows that there exist unique numbers* $\alpha_i, \ i \in I$, *and* $\beta_j, \ j \in J$, *such that*

$$f_z(\bar{z}) \ = \ \sum_{i=1}^{q} \alpha_i h_z^i(\bar{z}) \ + \ \sum_{j=1}^{s} \beta_j \varphi_z^j(\bar{z}). \tag{10}$$

Put

$$\Delta_{ij} \ = \ \beta_i \ - \ \beta_j \frac{\mu_i}{\mu_j} \ , \quad i, j \in J,$$

and let Δ *be the* $s \times s$ *matrix with* Δ_{ij} *as its* (i, j)*th element.*
(5.5) *All off-diagonal elements of* Δ *are unequal to zero.*
Put

$$L(z) \ = \ \sum_{i=1}^{q} \lambda_i h^i(z) \ + \ \sum_{j=1}^{s} \mu_j \varphi^j(z),$$

where $\lambda_i, \ \mu_j$ *satisfy* (9). *From condition* (5.3) *we see that* $L_t(\bar{z}) \neq 0$. *We define*

$$\gamma_j \ = \ sign(\mu_j \cdot L_t(\bar{z})) \ , \quad j \in J.$$

Moreover, let δ_j *denote the number of negative entries in the* j*th column of* $\Delta, \ j \in J$. *Thereby, we obtain the characteristic numbers* $\gamma_j, \ \delta_j, \ j \in J$.

A combination of (9), (10), and condition (5.5), together with the linear independence of the set $\{h_x^i(\bar{z}), i \in I, \varphi_x^j(\bar{z}), j \in J \setminus \{k\}\}$ for any $k \in J$, yields that $\bar{z}$ is a nondegenerate critical point if we delete φ^k as a constraint. For $k \in J$ put

$$M_k = \{z \in \mathbb{R}^{n+1} \mid h^i(z) = 0, \ i \in I, \ \varphi^j(z) = 0, \ j \in J \setminus \{k\}\},$$
$$M_k^+ = \{z \in M_k \mid \varphi^k(z) \geq 0\}.$$

From conditions (5.2), (5.3) and the fact that the h^i, φ^j are C^3-functions it follows that, locally around $\bar{z}$, the set M_k is a one-dimensional C^3-manifold, $k = 1, \ldots, s$. Furthermore, in a neighborhood of $\bar{z}$, the set Σ is the union of the sets M_k^+, $k = 1, \ldots, s$. The indices (LI, LCI, QI, QCI) along $M_k^+ \setminus \{\bar{z}\}$ are equal to $(\delta_k, s-1-\delta_k, 0, 0)$. As t increases and passes the value $\bar{t}$, the set M_k^+ emanates from $\bar{z}$ (ends at $\bar{z}$) according to $\gamma_k = +1$ (-1). If MFCQ is violated at $\bar{z}$, there is exactly one k with $\delta_k = 0$, i.e., there is exactly one branch M_k of local minima. Furthermore, if $\bar{z}$ satisfies MFCQ, then there are branches of local minima if and only if $\bar{z}$ is a local minimum itself. In the latter case, there are exactly two indices k_1 and k_2 with $\delta_{k_1} = \delta_{k_2} = 0$, and these indices satisfy $\gamma_{k_1} = -\gamma_{k_2}$ (thus, the set of local minima of $P^{(\bar{x}, \bar{t})}$ does not exhibit a turning point at $(\bar{x}, \bar{t})$). Figure 2.5 shows two examples for Σ around g.c. points of type 5.

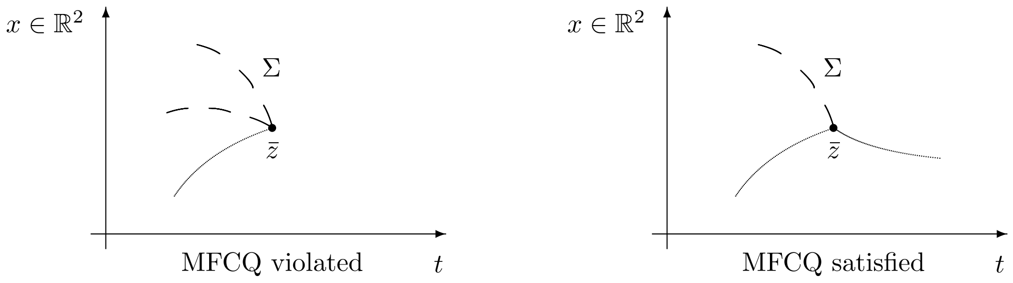


FIG. 2.5. *G.c. points of type* 5.

3. The nonreducible case. In this section, we deal with generic violations of the reduction approach in section 2.2, where we assumed that for given $\bar{t}$ and a g.c. point $\bar{x}$ of $SIP(\bar{t})$, all elements of $Y_0(\bar{x}, \bar{t})$ are nondegenerate global minima for $Q(\bar{x}, \bar{t})$. By Assumption 1, $Y_0(\bar{x}, \bar{t})$ turned out to be a finite set. Now we *require* $Y_0(\bar{x}, \bar{t})$ to be finite, i.e.,

$$Y_0(\bar{x}, \bar{t}) = \{\bar{y}^j,\ j \in J\},\ J = \{1, \dots, s\},\ s \in \mathbb{N}_0,$$

and we assume that exactly one of these points is degenerate. After renumbering, we may assume that $\bar{y}^1, \dots, \bar{y}^{s-1}$ are nondegenerate global minima of $Q(\bar{x}, \bar{t})$, whereas $\bar{y}^s$ is a degenerate global minimum. Let Assumption 3 be fulfilled also in this section; i.e., all defining functions of $SIP(t)$ are of differentiability class C^3.

In the generic case, only the degenerate types discussed in section 2.2 play a role for $\bar{y}^s$. Since $\bar{y}^s$ is a local minimum, and since we have only one parameter t at hand, the restricted Hessian of the corresponding Lagrangian $\mathcal{L}_{yy}(\bar{x}, \bar{t}, \bar{y}^s)|_{T_{\bar{y}}Y(\bar{t})}$ is nonsingular generically. In fact, the first such singularity (in one dimension) takes the form y^4, which has singularity-codimension two. Hence, the singularities generically occurring at $\bar{y}^s$ are related to the types 2, 4, and 5, which leads to the three additional types 6, 7, and 8 for the semi-infinite case.

The following subsections treat the definitions of the three new types as well as results about the corresponding local structure of Σ. Proofs will be given as far as they have not been published in [6], where g.c. points of type 6 are treated extensively.

3.1. Points of type 6. A g.c. point of type 6 can be roughly characterized by the fact that the degeneracy of the minimum $\bar{y}^s$ of $Q(\bar{x}, \bar{t})$ is due to the vanishing of exactly one Lagrange multiplier (i.e., ND1 in Definition 2.3 is violated), whereas LICQ is satisfied at $\bar{y}^s$. In order to improve readability, we will not formulate all conditions in original variables, but we will apply sequential simplification. For explicit formulations within original coordinates we refer to [23].

DEFINITION 3.1 (type 6). *A point* $\bar{z} = (\bar{x}, \bar{t}) \in \Sigma$ *is of* type 6 *if the following conditions* (6.1)–(6.7)* *are fulfilled.*

(6.1) *There exists an* $s \in \mathbb{N}_0$ *with* $Y_0(\bar{x}, \bar{t}) = \{\bar{y}^j,\ j \in J\},\ J = \{1, \dots, s\}$, *and the points* $\bar{y}^1, \dots, \bar{y}^{s-1}$ *are nondegenerate global minima for* $Q(\bar{x}, \bar{t})$.

(6.2) *The set of vectors* $\{h_x^i(\bar{z}),\ i \in I,\ g_x(\bar{z}, \bar{y}^j),\ j \in J\}$ *is linearly independent.*

Condition (6.2) *implies the existence of unique real numbers* $\bar{\lambda}_i,\ i \in I$, *and* $\bar{\mu}_j,\ j \in J$, *satisfying*

$$f_x(\bar{z}) \ = \ \sum_{i \in I} \bar{\lambda}_i h_x^i(\bar{z}) \ + \ \sum_{j \in J} \bar{\mu}_j g_x(\bar{z}, \bar{y}^j). \tag{11}$$

(6.3) *In* (11)*, we have* $\bar{\mu}_j \neq 0,\ j \in J$*.*
Now, consider the one-parametric finite optimization problem

$Q(\bar{x}, t)$ *minimize* $g(\bar{x}, t, \cdot)$ *on the feasible set* $Y(t)$*.*

(6.4) *The point* $(\bar{t}, \bar{y}^s)$ *satisfies conditions* (2.2)–(2.6) *(cf. Definition* 2.10*) for* $Q(\bar{x}, t)$*.*

(6.4.1) $\bar{y}^s$ *is a critical point for* $Q(\bar{x}, \bar{t})$*.*

(6.4.2) $L_0(\bar{x}, \bar{t}, \bar{y}^s) \neq \emptyset$*.*

After renumbering, we assume that $L_0(\bar{x}, \bar{t}, \bar{y}^s) = \{1, \ldots, p\},\ p \geq 1$*. Then, we have (cf. Definition* 2.2*):*

$$g_y(\bar{x}, \bar{t}, \bar{y}^s) \ = \ \sum_{k \in K} \bar{\beta}_k u_y^k(\bar{t}, \bar{y}^s) \ + \ \sum_{l=1}^{p} \bar{\gamma}_l v_y^l(\bar{t}, \bar{y}^s). \tag{12}$$

(6.4.3) *In* (12)*, exactly one of the Lagrange multipliers* $\bar{\gamma}_l$ *vanishes. After renumbering, we assume that* $\bar{\gamma}_p = 0$*. Define* T*,* $\tilde{T}$*, and* $\mathcal{L}(t, y)$ *as in Definition* 2.10*.*

(6.4.4) $\mathcal{L}_{yy}(\bar{x}, \bar{t}, \bar{y}^s)|_T$ *is nonsingular.*

(6.4.5) $\mathcal{L}_{yy}(\bar{x}, \bar{t}, \bar{y}^s)|_{\tilde{T}}$ *is nonsingular.*

Note that we do not require a transversality condition corresponding to condition (2.7) *here. However, condition* (6.7)* *(see below) implies transversality in the lower level problem (compare Theorem* 3.2(iv)*).*

Before we state the next conditions we reduce the problem locally by coordinate transformation. For $j \in \{1, \ldots, s-1\}$, we introduce the marginal functions (compare condition (6.1) and section 2.2) $\varphi^j(x, t) \ = \ g(x, t, y^j(x, t))$. In view of conditions (6.2) and (6.3) we can treat the equality constraint functions $h^i,\ i \in I$, and the inequality constraint functions $\varphi^j,\ j = 1, \ldots, s-1$, as new coordinates. Since $\bar{\mu}_j \neq 0,\ j = 1, \ldots, s-1$ (cf. condition (6.3)), we see that φ^j is a binding constraint. Hence, in the new coordinates we can delete the constraints $h^i,\ i \in I$, and $\varphi^j,\ j = 1, \ldots, s-1$. Subsequently, in view of conditions (6.4.1) and (6.4.3), the constraints $u^k,\ k \in K$, and $v^l,\ l \in \{1, \ldots, p-1\}$, are binding near the point $\bar{y}_s$. Hence, around $\bar{y}_s$, we can use the functions $u^k,\ k \in K$, and $v^l,\ l \in \{1, \ldots, p-1\}$, as new coordinates and they can be deleted in our further considerations.

The preceding observations show that we may proceed locally with the following simplified system $SIP(t)^*$ (in new coordinates and dimensions), where $I = \emptyset,\ K = \emptyset$, and $|L| = 1$:

$SIP(t)^*$ minimize $f(\cdot, t)$ on the feasible set $M(t)$,

where

$$\begin{aligned} M(t) &= \{x \in \mathbb{R}^n \mid g(x, t, y) \geq 0 \ , \ y \in Y(t)\}, \\ Y(t) &= \{y \in \mathbb{R}^m \mid v(t, y) \geq 0 \ \}, \end{aligned}$$

and $Y_0(\bar{x}, \bar{t}) = \{\bar{y}\}$. The corresponding lower level problem is

$Q(x, t)^*$ min $g(x, t, y)$ subject to $v(t, y) \geq 0$.

In order to study the feasible set $M(t)$ for t near $\bar{t}$, we merely have to focus on the behavior of the functions g and v around $(\bar{x}, \bar{t}, \bar{y})$.

From condition (6.4) it follows for $Q(x,t)^*$ that

$$g_y(\bar{x},\bar{t},\bar{y}) \;=\; \bar{\gamma}\; v_y(\bar{t},\bar{y}) \tag{13}$$

with $\bar{\gamma}=0$ and $v_y(\bar{t},\bar{y})\neq 0$. Hence, (13) can be read as a critical point relation either treating v as an equality constraint, i.e., for

$$Q(x,t)^*_e \qquad \min\; g(x,t,y) \;\text{ subject to }\; v(t,y)=0\;,$$

or just deleting v as a constraint, i.e., for

$$Q(x,t)^*_d \qquad \min\; g(x,t,y)\;.$$

By conditions (6.4.4) and (6.4.5), $\bar{y}$ is a nondegenerate critical point for both $Q(\bar{x},\bar{t})^*_e$ and $Q(\bar{x},\bar{t})^*_d$. Thus, there exist locally unique C^2-functions $y^e(x,t)$ and $\gamma^e(x,t)$, where $y^e(x,t)$ is the unique critical point near $\bar{y}$ for $Q(\bar{x},t)^*_e$ with t near $\bar{t}$ (with Lagrange multiplier $\gamma^e(x,t)$), as well as a C^2-function $y^d(x,t)$ with the analogous property. Now define the problems

$$SIP(t)^*_e \qquad \min\; f(x,t) \;\text{ subject to }\; g(x,t,y^e(x,t))\geq 0$$

and

$$SIP(t)^*_d \qquad \min\; f(x,t) \;\text{ subject to }\; g(x,t,y^d(x,t))\geq 0\;.$$

Convention: In the following we mark conditions (6.5)–(6.7) with a star (*) in order to underline that they are conditions in terms of the simplified system.

Definition 3.1 (type 6, continued).

(6.5)* *$\bar{x}$ is a nondegenerate critical point for $SIP(\bar{t})^*_e$.*

(6.6)* *$\bar{x}$ is a nondegenerate critical point for $SIP(\bar{t})^*_d$.*

By condition (6.5), there exist locally unique C^2-functions $x^e(t)$ and $\mu^e(t)$, where $x^e(t)$ is the unique critical point near $\bar{x}$ for $SIP(t)^*_e$ with t near $\bar{t}$ (with Lagrange multiplier $\mu^e(t)$). In a similar way, condition (6.6)* gives rise to C^2-functions $x^d(t)$ and $\mu^d(t)$.*

Finally, we define the values α and δ by

$$\alpha = \frac{d}{dt}\; v\left(t,\; y^d(x^d(t),t)\right)|_{t=\bar{t}}\;,$$
$$\delta = \frac{d}{dt}\; \gamma^e\left(x^e(t),t\right)|_{t=\bar{t}}\;.$$

(6.7)* $\alpha\neq 0$.

The characteristic numbers are sign(α) and sign(δ).

Locally around a point of type 6, Σ is composed by means of the C^2-curves $t\longmapsto (x^e(t),t)$ and $t\longmapsto (x^d(t),t)$. From the next theorem, we obtain exactly six possibilities according to the local structure of Σ, four of which are depicted in Figure 3.1. We include a proof of this theorem in the appendix.

Theorem 3.2. *Let $\bar{z}=(\bar{x},\bar{t})$ be a point of type 6. Then, the following holds:*

(i) *if $\dot{x}^e(\bar{t})-\dot{x}^d(\bar{t})$ vanishes, then the value $\alpha\cdot\delta$ is negative (where the dot denotes derivation with respect to t),*

(ii) *sign(δ) does not vanish,*

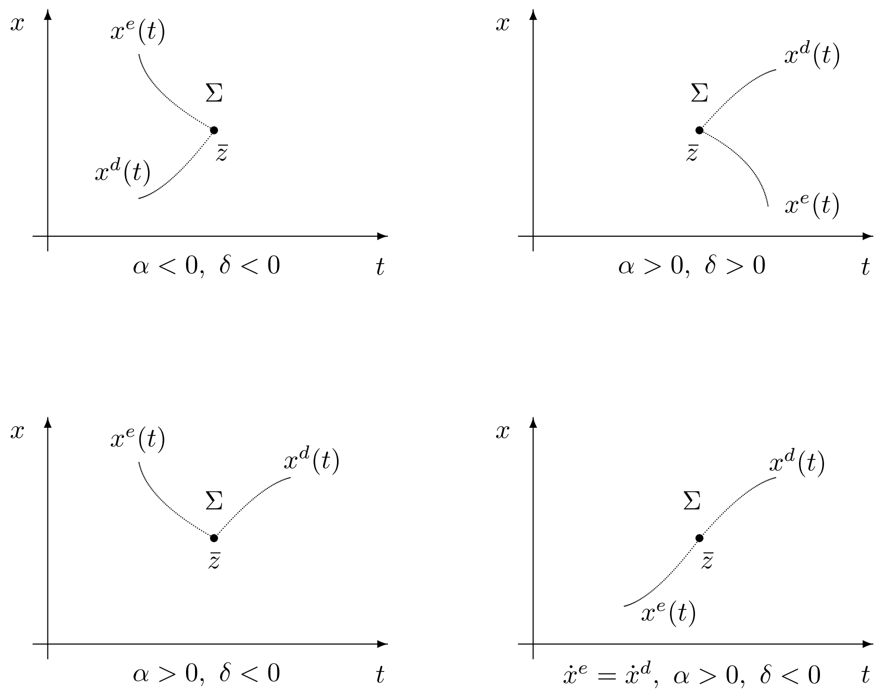


FIG. 3.1. *G.c. points of type* 6.

(iii) *for t in a neighborhood of $\bar{t}$, the points $y^d(x^d(t),t)$ and $y^e(x^e(t),t)$ are local minima of the problems $Q(x^d(t),t)^*_d$ and $Q(x^e(t),t)^*_e$, respectively,*

(iv) *the C^2-curves $t \longmapsto (y^e(x^e(t),t),t)$ and $t \longmapsto \big(y^d(x^d(t),t),t\big)$ meet in $(\bar{y},\bar{t})$ under a nonvanishing angle.*

REMARK 3.1. *Altogether, the curves $t \longmapsto (x^e(t), y^e(x^e(t),t),t)$ and $t \longmapsto \big(x^d(t), y^d(x^d(t),t),t\big)$ meet in $(\bar{x},\bar{y},\bar{t})$ under a nonvanishing angle, because this holds particularly for their lower level components by Theorem* 3.2(iv)*. Anyway, the projections $t \longmapsto (x^e(t),t)$ and $t \longmapsto \big(x^d(t),t\big)$ need* not *meet in $(\bar{x},\bar{t})$ under a nonvanishing angle. Equation* (40) *(cf. the appendix) even implies the equality $\dot{x}^e(\bar{t}) = \dot{x}^d(\bar{t})$ whenever x^e and x^d are scalar functions (by Cramer's rule). Thus, condition* (6.10)* *in our previous paper* [6] *has to be deleted.*

The results in parts (ii) and (iii) of Theorem 3.2 together with condition (6.7)* yield that exactly one branch of each graph (emanating at or ending in $\bar{z}$) belongs to Σ, as depicted in Figure 3.1. From this we immediately obtain the following.

COROLLARY 3.3. *Let $\bar{z}$ be a point of type* 6*. Then, $\bar{z}$ is a turning point for Σ if and only if $\alpha \cdot \delta$ is positive.*

Moreover, by part (i) of Theorem 3.2, the set Σ is locally a C^1-manifold if the curves $t \longmapsto (x^e(t),t)$ and $t \longmapsto (x^d(t),t)$ meet under a vanishing angle. Examples show that, in general, Σ is not a C^2-manifold in this case.

Now, we will study the change of the indices (LI, LCI, QI, QCI) when passing $\bar{z}$ along Σ, by comparing the indices at $(x^e(\bar{t}),\bar{t})$ and $(x^d(\bar{t}),\bar{t})$ for the problems $SIP(t)^*_d$ and $SIP(t)^*_e$ as introduced in conditions (6.5)* and (6.6)*, respectively. To this aim, we define the Lagrangians corresponding to $SIP(t)^*_d$ and $SIP(t)^*_e$, as well as their evaluated Hessians by

$$\begin{array}{rclcrcl} L^d(x,t,\mu) &=& L(x,t,\mu,y^d(x,t)) \ , & H_d &=& L^d_{xx}(\bar{x},\bar{t},\bar{\mu}), \\ L^e(x,t,\mu) &=& L(x,t,\mu,y^e(x,t)) \ , & H_e &=& L^e_{xx}(\bar{x},\bar{t},\bar{\mu}), \end{array}$$

where

$$L(x,t,\mu,y) \;=\; f(x,t) - \mu g(x,t,y).$$

The tangent spaces to the feasible sets of both problems are given by

$$T^d \;=\; T^e \;=\; T \;=\; \mathrm{Ker}\ g_x^\top(\bar{x},\bar{t},\bar{y}).$$

Recall that, in order to compute the change of the quadratic index, we have to compare the number of negative eigenvalues of the restricted Hessians $H_d|_T$ and $H_e|_T$. The next lemma is easily checked by using properties of the determinant function, whereas Lemma 3.5 is obvious.

Lemma 3.4. *Consider an $n \times n$ matrix A and a column vector $b \in \mathbb{R}^n \setminus \{0\}$. Let V be an $n \times (n-1)$ matrix whose columns form a basis of* Ker $b^\top$ *and put $\tilde{V} = (V, b)$. Then, we have*

$$\det \begin{pmatrix} A & b \\ b^\top & 0 \end{pmatrix} \;=\; -\frac{||b||_2^4}{\det^2 \tilde{V}} \cdot \det\left(V^\top A V\right).$$

Lemma 3.5. *Let A and B be symmetric $n \times n$ matrices with a positive semidefinite difference $A - B$. Then, if B is positive definite, so is A.*

Now we state our main result concerning index changes at points of type 6.

Theorem 3.6. *Let $\bar{z} = (\bar{x}, \bar{t})$ be a point of type* 6. *When passing the point $\bar{z}$ along Σ, the following holds for the corresponding indices:*

(i) *LI, LCI remain unchanged,*

(ii) *QI either remains constant or changes by one,*

(iii) *QI changes by one if and only if $\bar{z}$ is a turning point for Σ.*

Moreover, we have the following:

(iv) *If $\bar{z}$ is a turning point for Σ and one of the branches of Σ consists of local minimizers, then this is the branch corresponding to the curve $t \longmapsto \left(x^d(t), t\right)$.*

Proof. Part (i). It is an immediate consequence of condition (6.3).

Part (ii). In [6] the equation

$$V^\top H_d V - V^\top H_e V \;=\; \bar{\mu}\vartheta \left(V^\top \gamma_x^e\right)\left(V^\top \gamma_x^e\right)^\top \tag{14}$$

is shown, where V is a basis matrix of $T = \mathrm{Ker}\ g_x^\top$ and $\vartheta = v_y^\top g_{yy}^{-1} v_y$. Hence, the restrictions of H_d and H_e to T differ by a matrix of rank one at most, which proves the assertion.

Part (iii). Let V and ϑ be defined as in the proof of part (ii) (the definitions of A_d and A_e as well as the formulas (41) and (43) are contained in the appendix). The determinant property of Schur complements yields

$$\frac{\det \begin{pmatrix} H_d & -g_x \\ -g_x^\top & 0 \end{pmatrix}}{\det \begin{pmatrix} H_e & -g_x \\ -g_x^\top & 0 \end{pmatrix}} \;=\; -\vartheta \cdot \frac{\det A_d}{\det A_e}. \tag{15}$$

Thus, we have

$$\frac{\det H_d|_T}{\det H_e|_T} = \frac{\det V^\top H_d V}{\det V^\top H_e V} \overset{\text{Lemma 3.4}}{=} \frac{\det\begin{pmatrix} H_d & -g_x \\ -g_x^\top & 0 \end{pmatrix}}{\det\begin{pmatrix} H_e & -g_x \\ -g_x^\top & 0 \end{pmatrix}} \overset{(15),(41)}{=} -\vartheta \cdot \frac{\delta}{\alpha}.$$

Since ϑ is positive by (43), it follows that the value $\mathrm{QI}(H_d|_T) - \mathrm{QI}(H_e|_T)$ is odd if and only if $\alpha \cdot \delta$ is positive. Thus, the assertion follows from Corollary 3.3 and part (ii).

Part (iv). Since one of the branches consists of local minimizers we have $\bar{\mu} > 0$ by condition (6.3) and by part (i). Hence, the assertion follows from (14), Lemma 3.5, and part (iii). □

For examples showing that both situations in Theorem 3.6(ii) occur, compare [6]. (Note that in [6] a class of optimization problems broader than $SIP(t)$ is under investigation, but nevertheless, the cited example fits in our context.)

3.2. Points of type 7. At g.c. points of type 7, it is the violation of LICQ in the lower level problem $Q(\bar{x}, \bar{t})$ that causes the degeneracy of the minimum $\bar{y}^s$. Yet, the total number of active constraints at $\bar{y}^s$ (equalities and inequalities) does not exceed the lower level dimension m.

Definition 3.7 (type 7). *A point $\bar{z} = (\bar{x}, \bar{t}) \in \Sigma$ is of* type 7 *if the following conditions* (7.1)–(7.3) *are fulfilled.*

(7.1) *There exists an $s \in \mathbb{N}_0$ with $Y_0(\bar{x}, \bar{t}) = \{\bar{y}^j,\ j \in J\}$, $J = \{1, \dots, s\}$, and the points $\bar{y}^1, \dots, \bar{y}^{s-1}$ are nondegenerate global minima for $Q(\bar{x}, \bar{t})$.*

Define the marginal functions $\varphi^j(x,t) = g(x,t,y^j(x,t))$, $j \in \{1, \dots, s-1\}$, as well as the function $\varphi^s(x,t) = g(x,t,\bar{y}^s)$ and the one-parametric finite optimization problem

$$P(t) \qquad \textit{minimize} \;\; f(\cdot, t) \;\; \textit{on the feasible set } \mathcal{M}(t),$$

where

$$\mathcal{M}(t) \;=\; \{x \in \mathbb{R}^n \;\big|\; h^i(x,t) = 0\ ,\ i \in I,\ \varphi^j(x,t) \geq 0\ ,\ j \in J\}.$$

Note that, as in problem $P^{(\bar{x},\bar{t})}$ (cf. section 2.2*), we have $J_0(\bar{x}, \bar{t}) = J$ by definition of the functions φ^j.*

(7.2) *$\bar{x}$ is a nondegenerate critical point for $P(\bar{t})$.*

(7.3) *The point $(\bar{t}, \bar{y}^s)$ satisfies conditions* (4.2)–(4.7) *(cf. Definition* 2.12*) for $Q(\bar{x}, t)$.*

After renumbering we may assume that $L_0(\bar{x}, \bar{t}, \bar{y}^s) = \{1, \dots, p\}$, $p \geq 0$. Moreover, let K possess a fixed order.

(7.3.1) $|K| + p > 0$.

(7.3.2) $|K| + p - 1 < m$.

(7.3.3) *The matrix $M = (u_y^k, k \in K, v_y^1, \dots, v_y^p)|_{(\bar{t},\bar{y})}$ has rank $|K| + p - 1$.*

Let the one-dimensional space $\mathrm{Ker}(M)$ *be generated by $(\bar{\beta}, \bar{\gamma})^\top$ and define $\mathcal{L}(t,y)$, T, W, and A as in Definition* 2.12.

(7.3.4) *A is nonsingular.*

Define $\delta \;=\; g_y^\top(\bar{x}, \bar{t}, \bar{y}^s) W A^{-1} W^\top g_y(\bar{x}, \bar{t}, \bar{y}^s)$.

(7.3.5) $\delta \neq 0$.

In the case $p \geq 1$ we additionally require

(7.3.6) $\bar{\gamma}_l \neq 0,\ l \in \{1, \dots, p\}$

and we normalize the $\bar{\gamma}_l$'s by setting $\bar{\gamma}_p = 1$.

The characteristic number for a g.c. point of type 7 *is sign*(δ)*.*

Before we perform an analysis of the local structure of Σ in a neighborhood of $\bar{z}$, we reduce the problem locally as in section 3.1. In the upper level problem, we treat the equality constraint functions h^i, $i \in I$, and the marginal functions φ^j, $j = 1, \dots, s-1$, as new coordinates. In the lower level problem we have to consider two cases.

Case 1. $p = 0$.
By condition (7.3.3) the set of vectors $\{u^k_y(\bar{t}, \bar{y}),\ k \in K \setminus \{k_0\}\}$ is linearly independent for some $k_0 \in K$. Hence, we introduce the equality constraint functions u^k, $k \in K \setminus \{k_0\}$ as new coordinates.

Case 2. $p \geq 1$.
The set of vectors $\{u^k_y,\ k \in K,\ v^l_y,\ l \in \{1, \dots, p\} \setminus \{l_0\}\}$ is linearly independent for any $l_0 \in \{1, \dots, p\}$ by conditions (7.3.3) and (7.3.6). Furthermore, by condition (7.3.6) the corresponding constraints are binding near the point $\bar{y}^s$. Thus, we choose $l_0 = p$ and treat the functions u^k, $k \in K$, and v^l, $l \in \{1, \dots, p-1\}$, as new coordinates.

The preceding observations show that, by deleting all functions which define new coordinates, we may proceed locally with the following simplified system $SIP(t)^*$ (in new coordinates and dimensions), satisfying $I = \emptyset$ and $|K| + |L| = 1$:

$SIP(t)^*$ minimize $f(\cdot, t)$ on the feasible set $M(t)$,

where

$$\begin{aligned} M(t) &= \{x \in \mathbb{R}^n \mid g(x,t,y) \geq 0\ ,\ y \in Y(t)\} \\ Y(t) &= \{y \in \mathbb{R}^m \mid w(t,y)\ \rho\ 0\ \}, \end{aligned}$$

and $Y_0(\bar{x}, \bar{t}) = \{\bar{y}\}$. In Case 1, ρ means "="; otherwise it means "$\geq$".

By condition (7.2) we have $g_x(\bar{z}, \bar{y}) \neq 0$ and $f_x(\bar{z}) = \bar{\mu} g_x(\bar{z}, \bar{y})$ with a Lagrange multiplier $\bar{\mu} \neq 0$. Moreover, the matrix

$$(16) \qquad \begin{pmatrix} f_{xx}(\bar{z}) - \bar{\mu} g_{xx}(\bar{z}, \bar{y}) & -g_x(\bar{z}, \bar{y}) \\ -g_x^\top(\bar{z}, \bar{y}) & 0 \end{pmatrix} \text{ is nonsingular.}$$

Conditions (7.3.3), (7.3.4), and (7.3.5) imply

$$(17) \qquad w_y \ = \ 0,$$

$$(18) \qquad w_t \cdot w_{yy} \ \text{ is nonsingular,}$$

$$(19) \qquad \delta \ = \ w_t^{-1} g_y^\top w_{yy}^{-1} g_y \ \neq \ 0,$$

all partial derivatives being evaluated at $(\bar{x}, \bar{t}, \bar{y})$.

THEOREM 3.8. *Let $\bar{z} = (\bar{x}, \bar{t})$ be a point of type* 7*. Then, Σ locally consists of the branch corresponding to nonnegative α of a one-dimensional C^2-manifold*

$$\Gamma \ = \ \{(x(\alpha), t(\alpha)),\ \alpha \in (-\varepsilon, \varepsilon)\},$$

which has a quadratic turning point at $\bar{z} = (x(0), t(0))$. For increasing t, Σ emanates from (ends at) $\bar{z}$ according to sign$(\delta) = -1$ $(+1)$.

A proof of Theorem 3.8 is given in the appendix. In order to illustrate the structure of Σ in a neighborhood of a g.c. point of type 7, we give the following example.

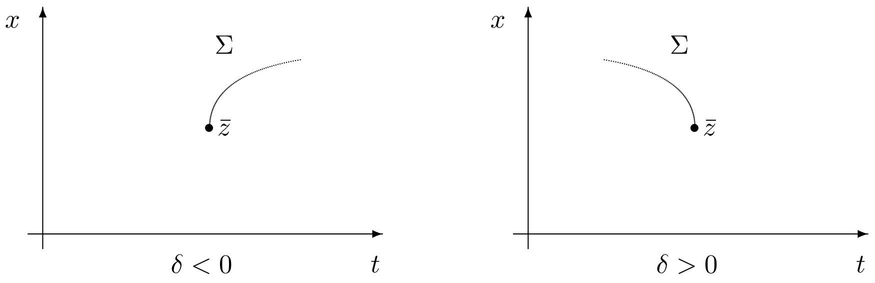


Fig. 3.2. *G.c. points of type* 7.

Example 3.1. *Consider the problem (with $I = K = \emptyset$, $|L| = 1$, $n = m = 1$)*

$$\textit{minimize} \quad f(x,t) = x \quad \textit{on the feasible set} \quad M(t),$$

where

$$M(t) = \{x \in \mathbb{R} \mid 0 \le g(x,t,y) = x + y \ , \ y \in Y(t)\},$$
$$Y(t) = \{y \in \mathbb{R} \mid 0 \le v(t,y) = t - y^2 \ \}.$$

It is easily checked that $(\bar{x}, \bar{t}) = (0,0)$ is a g.c. point of type 7 with $Y_0(0,0) = \{\bar{y}\} = \{0\}$. In particular, we obtain $\delta = -\frac{1}{2}$.

In fact, observing that

$$Y(t) \ = \ \begin{cases} \emptyset, & t < 0, \\ [-\sqrt{t}, \sqrt{t}\ ], & t \ge 0, \end{cases} \quad \textit{and} \quad M(t) \ = \ \begin{cases} \mathbb{R}, & t < 0, \\ [\ \sqrt{t}, \infty), & t \ge 0, \end{cases}$$

there are no g.c. points for $t < 0$, but a unique g.c. point $x = \sqrt{t}$ for $t \ge 0$. This is just the situation sketched in Figure 3.2 for $\delta < 0$.

Note that in this example, the set Σ of g.c. points coincides with the set of global minima.

3.3. Points of type 8. Like at g.c. points of type 7, the violation of LICQ in the lower level problem $Q(\bar{x}, \bar{t})$ causes the degeneracy of the minimum $\bar{y}^s$. However, at g.c. points of type 8 the total number of active constraints at $\bar{y}^s$ (equalities and inequalities) exceeds the lower level dimension m (by one).

Definition 3.9 (type 8). *A point $\bar{z} = (\bar{x}, \bar{t}) \in \Sigma$ is of* type 8 *if the following conditions* (8.1)–(8.3) *are fulfilled.*

(8.1) *There exists an $s \in \mathbb{N}_0$ with $Y_0(\bar{x}, \bar{t}) = \{\bar{y}^j,\ j \in J\}$, $J = \{1, \ldots, s\}$, and the points $\bar{y}^1, \ldots, \bar{y}^{s-1}$ are nondegenerate global minima for $Q(\bar{x}, \bar{t})$.*

(8.2) *$\bar{x}$ is a nondegenerate critical point for $P(\bar{t})$ with $P(t)$ defined as in Definition* 3.7.

(8.3) *The point $(\bar{t}, \bar{y}^s)$ satisfies conditions* (5.2)–(5.5) *(cf. Definition* 2.13*) for $Q(\bar{x}, t)$.*

(8.3.1) $|K| + |L_0(\bar{x}, \bar{t}, \bar{y}^s)| = m + 1$.

After renumbering we may assume that $L_0(\bar{x}, \bar{t}, \bar{y}^s) = \{1, \ldots, p\}$, $p \ge 2$. Moreover, let K possess a fixed order and put $\omega = (y,t)$.

(8.3.2) *The set of vectors $\{u^k_\omega(\bar{t}, \bar{y}^s),\ k \in K,\ v^l_\omega(\bar{t}, \bar{y}^s),\ l \in \{1, \ldots, p\}\}$ is linearly independent.*

Now put $M = \big(u^k_y(\bar t, \bar y^s),\ k \in K,\ v^1_y(\bar t, \bar y^s), \dots, v^p_y(\bar t, \bar y^s)\big)$. *By condition* (8.3.2), M *has rank* m *and thus, there exists a solution* (β, γ) *(nonvanishing and unique up to a common multiple) of*

$$M \begin{pmatrix} \beta \\ \gamma \end{pmatrix} \;=\; 0. \tag{20}$$

(8.3.3) *In* (20) *we have* $\gamma_l \neq 0,\ l = 1, \dots, p$.
By condition (8.3.2), *there is a unique solution* (ζ, η) *of the system*

$$g_\omega(\bar x, \bar t, \bar y^s) \;=\; \big(u^k_\omega(\bar t, \bar y^s),\ k \in K,\ v^1_\omega(\bar t, \bar y^s), \dots, v^p_\omega(\bar t, \bar y^s)\big) \begin{pmatrix} \zeta \\ \eta \end{pmatrix}. \tag{21}$$

Put

$$\Delta_{ij} \;=\; \eta_i \;-\; \eta_j \frac{\gamma_i}{\gamma_j}\ , \quad i, j = 1, \dots, p,$$

and let Δ *be the* $p \times p$ *matrix with* Δ_{ij} *as its* (i,j)*th element.*

(8.3.4) *All off-diagonal elements of* Δ *are unequal to zero.*
Put

$$\mathcal{L}(t, y) \;=\; \sum_{k \in K} \beta_k u^k(t, y) \;+\; \sum_{l=1}^{p} \gamma_l v^l(t, y),$$

where $\beta_k,\ \gamma_l$ *satisfy* (20). *From condition* (8.3.2) *we see that* $\mathcal{L}_t(\bar t, \bar y^s)$ *does not vanish. We define the characteristic numbers*

$$\alpha_l \;=\; sign\,(\gamma_l \cdot \mathcal{L}_t(\bar t, \bar y^s))\ , \quad l = 1, \dots, p.$$

Additionally, we distinguish two situations occurring at points of type 8 *since they give rise to essentially different local structures of* Σ. *A g.c. point* $\bar z$ *of type* 8 *is of* type 8a *if MFCQ is satisfied at* $\bar y^s$ *in the lower level problem* $Q(\bar x, \bar t)$. *Otherwise,* $\bar z$ *is of* type 8b.

In case of a g.c. point of type 8b *we define the additional characteristic number*

$$l^* \;=\; \operatorname{argmin} \left\{ \frac{\eta_l}{\gamma_l} \frac{\alpha_1}{\mathcal{L}_t(\bar t, \bar y^s)}\ ,\ l = 1, \dots, p \right\},$$

(which can be shown to be well defined).

Before we perform an analysis of the local structure of Σ in a neighborhood of $\bar z$, we reduce the problem locally again like in sections 3.1 and 3.2. In the upper level problem, we treat the equality constraint functions $h^i,\ i \in I$, and the marginal functions $\varphi^j,\ j = 1, \dots, s-1$, as new coordinates. Note that in the lower level problem, there are exactly two (one) branches of local minima if $\bar z$ is of type 8a (type 8b) (see Definition 2.13). For any index $q \in \{1, \dots, p\}$ we see from using (20) and (21) that

$$g_y(\bar x, \bar t, \bar y^s) \;=\; \sum_{k \in K} \left(\zeta_k - \frac{\eta_q}{\gamma_q} \beta_k \right) u^k_y(\bar t, \bar y^s) \;+\; \sum_{l \neq q} \Delta_{lq} v^l_y(\bar t, \bar y^s). \tag{22}$$

The right-hand side vectors in (22) are linearly independent by conditions (8.3.2) and (8.3.3), none of Δ_{lq} vanish by condition (8.3.4) and we have $|K| + |\{2, \dots, p\}|$

$= m$ by condition (8.3.1). As new coordinates we choose the constraint functions $u^k,\ k \in K$, and $v^l,\ l \in \{1, \dots, p\} \setminus \{l_1, l_2\}$, where the indices of branches consisting of local minima are contained in the set $\{l_1, l_2\}$. This yields a lower level problem of dimension one.

The preceding observations show that, by deleting all functions which define new coordinates, we may proceed locally with the following simplified system $SIP(t)^*$ (in new coordinates and dimensions), satisfying $I = K = \emptyset$ and $|L| = 2$:

$SIP(t)^*$ minimize $f(\cdot, t)$ on the feasible set $M(t)$,

where

$$\begin{aligned} M(t) &= \{x \in \mathbb{R}^n \mid g(x,t,y) \geq 0\ ,\ y \in Y(t)\}, \\ Y(t) &= \{y \in \mathbb{R}^1 \mid v^1(t,y) \geq 0\ ,\ v^2(t,y) \geq 0\}, \end{aligned}$$

and $Y_0(\bar{x}, \bar{t}) = \{\bar{y}\}$. The corresponding lower level problem is

$Q(x,t)^*$ min $g(x,t,y)$ subject to $v^1(t,y) \geq 0\ , v^2(t,y) \geq 0$.

From condition (8.2) we conclude that $g_x(\bar{z}, \bar{y}) \neq 0$ and $f_x(\bar{z}) = \bar{\mu} g_x(\bar{z}, \bar{y})$ with a Lagrange multiplier $\bar{\mu} \neq 0$. Furthermore, the matrix

$$\text{(23)} \qquad \begin{pmatrix} f_{xx}(\bar{z}) - \bar{\mu} g_{xx}(\bar{z}, \bar{y}) & -g_x(\bar{z}, \bar{y}) \\ -g_x^\top(\bar{z}, \bar{y}) & 0 \end{pmatrix} \text{ is nonsingular.}$$

Conditions (8.3.2), (8.3.3), and (8.3.4) imply that the 2×2 matrix

$$\text{(24)} \qquad \begin{pmatrix} v_y^1 & v_y^2 \\ v_t^1 & v_t^2 \end{pmatrix} \text{ is nonsingular,}$$

$$\text{(25)} \qquad v_y^1 \cdot v_y^2 \neq 0,$$

$$\text{(26)} \qquad g_y \neq 0,$$

all partial derivatives being evaluated at $(\bar{x}, \bar{t}, \bar{y})$. Moreover, a short calculation shows that

$$\text{(27)} \qquad \alpha_1 = \operatorname{sign}\left(v_t^1 \ - \ \frac{v_y^1}{v_y^2}\, v_t^2 \right),$$

$$\text{(28)} \qquad \alpha_2 = \operatorname{sign}\left(v_t^2 \ - \ \frac{v_y^2}{v_y^1}\, v_t^1 \right),$$

and that $\bar{z}$ is of type 8a (of type 8b) for $SIP(t)^*$ according to $\alpha_1 \cdot \alpha_2 = -1\ (+1)$. In case of a point of type 8b we find that

$$\text{(29)} \qquad l^* \in \{1,2\} \text{ is the unique index with } g_y \cdot v_y^{l^*} < 0.$$

From (24)–(26) the following lemma is easily deduced.

LEMMA 3.10. *The point $\bar{y}$ is a nondegenerate critical point both for problem $Q(\bar{x}, \bar{t})_1^*$ and $Q(\bar{x}, \bar{t})_2^*$, where*

$Q(x,t)_i^*$ *min $g(x,t,y)$ subject to $v^i(t,y) \geq 0$, $i \in \{1,2\}$.*

By Lemma 3.10, there exist locally unique C^2-functions $y^i(x,t)$ and $\gamma^i(x,t)$, $i \in \{1,2\}$, satisfying $y^i(\bar z) = \bar y$ and $\gamma^i(\bar z) = \bar\gamma^i$ (where $\bar\gamma^i$ denotes the unique Lagrange multiplier in problem $Q(\bar x, \bar t)^*_i$) and

$$\begin{aligned} g_y(x,t,y^i(x,t)) \;-\; \gamma^i(x,t)\; v^i_y(t,y^i(x,t)) \;&\equiv\; 0, \\ -v^i(t,y^i(x,t)) \;&\equiv\; 0. \end{aligned} \tag{30}$$

By differentiation of (30) with respect to x at $(\bar x, \bar t, \bar y)$ we obtain for $i \in \{1,2\}$

$$\begin{pmatrix} y^i_x \\ \gamma^i_x \end{pmatrix} = -\begin{pmatrix} g_{yy} - \bar\gamma^i v^i_{yy} & -v^i_y \\ -v^i_y & 0 \end{pmatrix}^{-1} \begin{pmatrix} g_{yx} \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{v^i_y} g_{yx} \end{pmatrix}. \tag{31}$$

Now we introduce the problems

$$SIP(t)^*_i \qquad \min\; f(x,t) \;\text{ subject to }\; g(x,t,y^i(x,t)) \geq 0\;, \quad i \in \{1,2\}.$$

Essentially by equation (31) we have the following lemma.

LEMMA 3.11. *The point $\bar x$ is a nondegenerate critical point both for problem $SIP(t)^*_1$ and $SIP(t)^*_2$.*

Proof. By Lemma 3.10 and Lemma 2.5

$$\frac{d}{dx} g(x,t,y^i(x,t)) \;\equiv\; g_x(x,t,y^i(x,t))$$

holds locally for $i \in \{1,2\}$ and hence

$$\frac{d}{dx} g(x,t,y^i(x,t))|_{(\bar x,\bar t)} \;=\; g_x(\bar x,\bar t,\bar y),$$

as well as

$$\begin{aligned} \frac{d^2}{dx^2} g(x,t,y^i(x,t))|_{(\bar x,\bar t)} \;&=\; g_{xx}(\bar x,\bar t,\bar y) \;+\; g_{xy}(\bar x,\bar t,\bar y) \cdot y^i_x(\bar x,\bar t) \\ &\overset{(31)}{=} g_{xx}(\bar x,\bar t,\bar y)\;. \end{aligned}$$

Defining the Lagrangians $L^i(x,t) = f(x,t) - \bar\mu g(x,t,y^i(x,t))$, $i \in \{1,2\}$, we obtain

$$L^i_x(\bar x,\bar t) = f_x(\bar x,\bar t) \;-\; \bar\mu g_x(\bar x,\bar t,\bar y) \;=\; 0, \tag{32}$$

$$T^i = \text{Ker}\left(\frac{d}{dx} g(x,t,y^i(x,t))|_{(\bar x,\bar t)}\right)^\top \;=\; \text{Ker}\; g_x^\top(\bar x,\bar t,\bar y), \tag{33}$$

$$L^i_{xx}(\bar x,\bar t)|_{T^i} = \begin{pmatrix} f_{xx}(\bar x,\bar t) - \bar\mu g_{xx}(\bar x,\bar t,\bar y) & -g_x(\bar x,\bar t,\bar y) \\ -g_x^\top(\bar x,\bar t,\bar y) & 0 \end{pmatrix}. \tag{34}$$

Thus, the assertion follows from condition (8.2). Note that the right-hand side expressions in (32), (33), and (34) do not depend on $i \in \{1,2\}$. □

By Lemma 3.11, there exist locally unique C^2-functions $x^i(t)$ and $\mu^i(t)$, $i \in \{1,2\}$ satisfying $x^i(\bar t) = \bar x$, $\mu^i(\bar t) = \bar\mu$ and

$$\begin{aligned} f_x(x^i(t),t) \;-\; \mu^i(t)\; g_x(x^i(t),t,y^i(x^i(t),t)) \;&\equiv\; 0, \\ -g(x^i(t),t,y^i(x^i(t),t)) \;&\equiv\; 0. \end{aligned} \tag{35}$$

Now, we turn to the local structure of Σ in a neighborhood of a g.c. point of type 8 (cf. Figures 3.3 and 3.4). A proof of the next theorem can be found in the appendix.

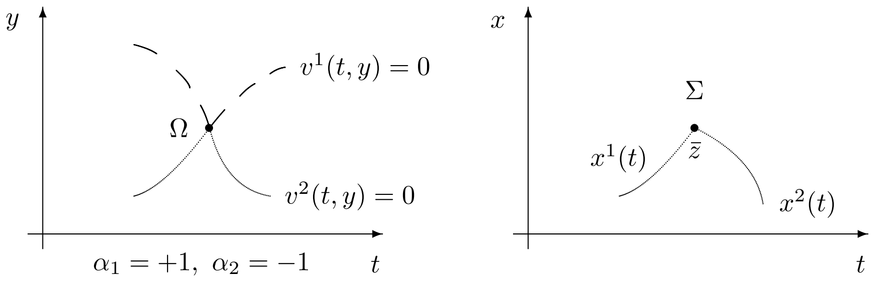


FIG. 3.3. *A g.c. point of type* 8a.

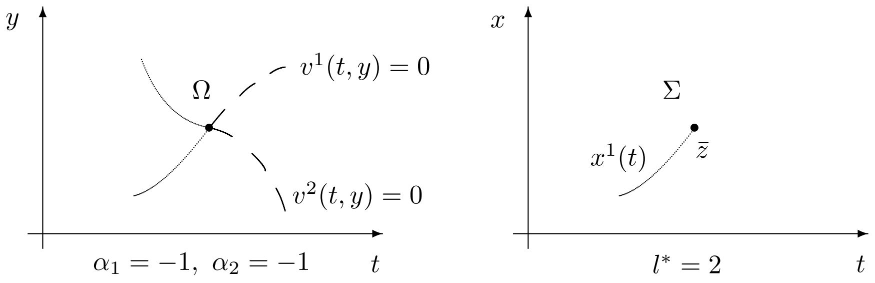


FIG. 3.4. *A g.c. point of type* 8b.

THEOREM 3.12.

(i) *Let $\bar z=(\bar x,\bar t)$ be a g.c. point of type* 8a. *Then, Σ is locally composed by means of the C^2-curves $t\longmapsto(x^1(t),t)$ and $t\longmapsto(x^2(t),t)$, which meet in $\bar z$ under a nonvanishing angle. Exactly one branch of each graph belongs to Σ, and the composition of the branches is such that Σ does not exhibit a turning point at $\bar z$. The index quadruple (LI,LCI,QI,QCI) remains constant when passing the point $\bar z$ along Σ.*

(ii) *Let $\bar z=(\bar x,\bar t)$ be a g.c. point of type* 8b. *Then, Σ locally consists of exactly one branch of the C^2-curve $t\longmapsto(x^i(t),t)$, where i is the index in $\{1,2\}\backslash\{l^*\}$. For increasing t, Σ emanates from (ends at) $\bar z$ according to $\alpha_i=+1\ (-1)$.*

In order to illustrate the structure of Σ in a neighborhood of a g.c. point of type 8, we give the following example.

EXAMPLE 3.2. *Consider the problem (with $I=K=\emptyset,\ |L|=2,\ n=m=1$)*

$$\textit{minimize}\ \ f(x,t)=x\ \ \textit{on the feasible set}\ \ M(t),$$

where

$$M(t)=\{x\in\mathbb{R}\mid 0\le g(x,t,y)=x+y\ ,\ y\in Y(t)\},$$
$$Y(t)=\{y\in\mathbb{R}\mid 0\le v^1(t,y)=y(1-y),\ 0\le v^2(t,y)=\theta(y-t)\},$$

and where θ is an additional parameter taking the values $+1$ and -1. It is easily checked that $(\bar x,\bar t)=(0,0)$ is a g.c. point of type 8a *(type* 8b*) with $Y_0(0,0)=\{\bar y\}=\{0\}$ for $\theta=+1\ (-1)$.*

First, consider the case $\theta = +1$. Observing that

$$Y(t) \;=\; \begin{cases} [0,1]\,, & t<0, \\ [t,1]\,, & 0 \le t \le 1, \\ \emptyset, & 1<t, \end{cases} \qquad \text{and} \;\; M(t) \;=\; \begin{cases} [\,0,\infty), & t<0, \\ [\,-t,\infty), & 0 \le t \le 1, \end{cases}$$

there is the unique g.c. point $x(t) = 0$ for $t < 0$, and the unique g.c. point $x(t) = -t$ for $0 \le t \le 1$. In particular, we compute $\alpha_1 = +1$ and $\alpha_2 = -1$.

In the case $\theta = -1$ we have

$$Y(t) \;=\; \begin{cases} \emptyset, & t<0, \\ [0,t]\,, & 0 \le t \le 1, \\ [0,1]\,, & 1<t, \end{cases} \qquad \text{and} \;\; M(t) \;=\; \begin{cases} \mathbb{R}, & t<0, \\ [\,0,\infty), & t \ge 0. \end{cases}$$

Thus, there are no g.c. points for $t < 0$, but the unique g.c. point $x(t) = 0$ for $t \ge 0$. We compute $\alpha_1 = \alpha_2 = +1$ and $l^* = 2$.

Note that in these examples, the sets Σ of g.c. points coincide with the sets of global minima.

4. On the genericity proof. In this section, we sketch the main ideas of the genericity part in the proof of Theorem 1.5. To this aim, we denote by $\mathcal{F}^* \subset C^3(\mathbb{R}^{n+1},\mathbb{R})^{|I|+1} \times C^3(\mathbb{R}^{n+m+1},\mathbb{R}) \times CUSC$ the set of all function vectors which give rise to g.c. points of types 1 to 8 only, if they are utilized as defining functions for $SIP(t)$. Theorem 1.5 then says that there is a subset $\mathcal{F}$ of $\mathcal{F}^*$ being C^3_s-open and dense in $C^3(\mathbb{R}^{n+1},\mathbb{R})^{|I|+1} \times C^3(\mathbb{R}^{n+m+1},\mathbb{R}) \times CUSC$. A genericity proof for a different class of one-parametric semi-infinite optimization problems has been given by Rupp [21]; however, he considered only problems where LICQ always holds in the lower level problem. It is straightforward to modify and generalize the latter proof, essentially since, by transversality arguments, it can be shown that the points $(\bar{t},\bar{y})$ at which LICQ fails to hold in the lower level problem generically are isolated (cf. also [10]).

The open part of the proof just consists of a continuity argument. Before we treat the dense part by using Thom's jet-transversality theorem, we give a short introduction to transversality theory, as far as we need it for our analysis. For details, cf., e.g., [9], [12].

Two smooth manifolds V, W in $\mathbb{R}^N$ are said to intersect *transversally* (notation: $V \overline{\pitchfork} W$) if at each intersection point $u \in V \cap W$ the tangent spaces T_uV, T_uW together span the embedding space:

$$T_uV \;+\; T_uW \;=\; \mathbb{R}^N. \tag{36}$$

The number $N - \dim V$ is called the *codimension* of V in $\mathbb{R}^N$, shortly $\operatorname{codim} V$, and we have

$$\operatorname{codim} V \;\le\; \dim W \tag{37}$$

whenever $V \overline{\pitchfork} W$ and $V \cap W \ne \emptyset$. For our purpose, the manifold W is induced by the 1-jet extension of a function $F \in C^\infty(\mathbb{R}^N,\mathbb{R}^M)$, i.e., by the mapping

$$j^1F:\; \mathbb{R}^N \longrightarrow J(N,M,1),\;\; z \longmapsto (z, F(z), F_z(z)),$$

where $J(N,M,1) = \mathbb{R}^{N+M+N\cdot M}$ and the partial derivatives are listed according to some order convention (cf. [12]). Choosing W as the graph of j^1F (notation:

$W = j^1F(\mathbb{R}^N))$ it is easily shown that W is a smooth manifold of dimension N in $J(N, M, 1)$. Given another smooth manifold V in $J(N, M, 1)$, we define the set

$$\pitchfork^1 V \;=\; \{F \in C^\infty(\mathbb{R}^N, \mathbb{R}^M) |\; j^1F(\mathbb{R}^N) \pitchfork V\}.$$

Our analysis bases on the following theorem, which is due to Thom.

THEOREM 4.1 (jet transversality). *With respect to the C^∞_s-topology, the set $\pitchfork^1 V$ is generic in $C^\infty(\mathbb{R}^N, \mathbb{R}^M)$.*

In particular, $\pitchfork^1 V$ is C^∞_s-dense in $C^\infty(\mathbb{R}^N, \mathbb{R}^M)$ and hence C^r_s-dense in $C^r(\mathbb{R}^N, \mathbb{R}^M)$ for any $r \in \mathbb{N}_0$ (cf. [9]).

Since jet transversality gives information about certain properties of the functions under investigation only at every *single* point we apply the concept of multijet transversality instead (cf. [12]). Thereby, we are able to study properties that have to be satisfied at all global minima of the lower level problem — i.e., at the points in $Y_0(\bar{x}, \bar{t})$ — at the same time. Let P be a positive integer and define

$$\mathbb{R}^N_P \;=\; \left\{(z^1, \ldots, z^P) \in \textstyle\prod_{k=1}^P \mathbb{R}^N |\; z^i \neq z^j \text{ for } 1 \leq i < j \leq P\right\},$$

as well as the multijet space

$$J_P(N, M, 1) \;=\; \left\{(z^1, u^1, \ldots, z^P, u^P) \in \textstyle\prod_{k=1}^P J(N, M, 1) |\; (z^1, \ldots, z^P) \in \mathbb{R}^N_P\right\}.$$

The multijet extension $j^1_P F :\; \mathbb{R}^N_P \longrightarrow J_P(N, M, 1)$ is the mapping

$$j^1_P F :\; (z^1, \ldots, z^P) \;\longmapsto\; \left(j^1 f(z^1), \ldots, j^1 f(z^P)\right),$$

and for a smooth manifold V in $J_P(N, M, 1)$ we define the set

$$\pitchfork^1_P V \;=\; \{F \in C^\infty(\mathbb{R}^N, \mathbb{R}^M) |\; j^1_P F(\mathbb{R}^N_P) \pitchfork V\}.$$

THEOREM 4.2 (multijet transversality). *With respect to the C^∞_s-topology, the set $\pitchfork^1_P V$ is generic in $C^\infty(\mathbb{R}^N, \mathbb{R}^M)$.*

In order to avoid technicalities, we construct the set $\mathcal{F} \subset \mathcal{F}^*$ only for the case $Y(t) = \{y \in \mathbb{R}^m |\; u(t, y) = 0\}$, i.e., for one equality constraint in the lower level problem, the general case running along the same lines (for details, cf. [23]). Consequently, g.c. points of type 6 and type 8 cannot occur in the $SIP(t)$ corresponding to a function vector $(f, g, u) \in \mathcal{F}$. Consider the sets of matrices

$$\mathbb{R}^{M \times N}_\rho \;=\; \left\{A \in \mathbb{R}^{M \times N} \;\middle|\; \text{rank}(A) = \rho\right\}$$

and, for a (possibly empty) index set $I \subset \{1, \ldots, N\}$,

$$\mathbb{R}^{M \times N}_{\rho, I} \;=\; \left\{A \in \mathbb{R}^{M \times N}_\rho \;\middle|\; A_{(I)} \in \mathbb{R}^{M \times (N - |I|)}_{\rho - |I|}\right\}, \tag{38}$$

where $A_{(I)}$ results from A by deleting the columns with indices in I. Part (i) of the following lemma can be found in [12], part (ii) in [23].

LEMMA 4.3.

(i) $\mathbb{R}^{M \times N}_\rho$ *is a smooth manifold of codimension* $(M - \rho)(N - \rho)$ *in* $\mathbb{R}^{M \times N}$.

(ii) $\mathbb{R}^{M \times N}_{\rho, I}$ *is a smooth manifold of codimension* $(M + |I| - \rho)(N - \rho)$ *in* $\mathbb{R}^{M \times N}$.

Now we construct the crucial manifold for our genericity proof. Let $s\in\mathbb{N}$, $0\le\rho_0\le\min\{n,s+1\}$, $I_0\subset\{1,\dots,s+1\}$, as well as $0\le\rho_j\le\min\{m,2\}$, $I_j\subset\{1,2\}$ for $1\le j\le s$. Then, the set

$$\begin{aligned}
&V_{s,\rho_0,\dots,\rho_s,I_0,\dots,I_s}\\
&=\{(\xi_1,\tau_1,\upsilon_1,0,0,\varphi_1,\gamma_1,\delta_1,\omega_1,\ \dots,\ \xi_s,\tau_s,\upsilon_s,0,0,\varphi_s,\gamma_s,\delta_s,\omega_s),\\
&\ (\xi_j,\tau_j,\upsilon_j,\varphi_j,\gamma_j,\delta_j,\omega_j)\in\mathbb{R}^n\times\mathbb{R}\times\mathbb{R}^m\times\mathbb{R}^n\times\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}^m,\\
&\ 1\le j\le s,\ (\xi_1,\tau_1)=\dots=(\xi_s,\tau_s),\\
&\ (\varphi_1^\top,\gamma_1^\top,\gamma_2^\top,\dots,\gamma_s^\top)\in\mathbb{R}^{n\times(s+1)}_{\rho_0,I_0},\ (\delta_j^\top,\omega_j^\top)\in\mathbb{R}^{m\times 2}_{\rho_j,I_j},\ 1\le j\le s\ \}
\end{aligned}$$

is a smooth manifold of codimension

$$2s+(s-1)(n+1)+(n+|I_0|-\rho_0)(s+1-\rho_0)+\sum_{j=1}^{s}(m+|I_j|-\rho_j)(2-\rho_j)$$

in $J_s(n+m+1,3,1)$. Due to Theorem 4.2 and to the Baire property of $C^\infty(\mathbb{R}^{n+m+1},\mathbb{R}^3)$ with the C^∞_s-topology, the set of functions

$$\tilde{\mathcal{F}}\ =\ \bigcap_{s=1}^{\infty}\ \bigcap_{\substack{\rho_0,\dots,\rho_s\\ I_0,\dots,I_s}}\ \pitchfork^1_s\ V_{s,\rho_0,\dots,\rho_s,I_0,\dots,I_s}$$

is generic, where the inner intersection ranges over all possible choices of $\rho_0,\dots,I_s$. Now we choose a function vector (f,g,u) from $\tilde{\mathcal{F}}$ and a g.c. point $(\bar x,\bar t)$ of the corresponding $SIP(t)$. We focus on the nontrivial case where $Y_0(\bar x,\bar t)\neq\emptyset$, and we choose indices $\bar y^1,\dots,\bar y^s\in Y_0(\bar x,\bar t)$ such that the set $\{\bar f_x,\bar g_x^1,\dots,\bar g_x^s\}$ is linearly dependent (where we let $\bar f_x=f_x(\bar x,\bar t)$ and $\bar g_x^j=g_x(\bar x,\bar t,\bar y^j)$). Obviously, the evaluated (reduced) multijet extension

$$\begin{aligned}
j^1_s(f,g,u)(\bar x,\bar t,\bar y^1,\dots,\bar x,\bar t,\bar y^s)=(&\bar x,\bar t,\bar y^1,\bar g^1,\bar u^1,\bar f_x^\top,(\bar g_x^1)^\top,(\bar g_y^1)^\top,(\bar u_y^1)^\top,\\
&\dots,\bar x,\bar t,\bar y^s,\bar g^s,\bar u^s,\bar f_x^\top,(\bar g_x^s)^\top,(\bar g_y^s)^\top,(\bar u_y^s)^\top\ )
\end{aligned}$$

is contained in some set $V_{s,\bar\rho_0,\dots,\bar\rho_s,I_0,\dots,I_s}$, where $\bar\rho_0=\operatorname{rank}(\bar f_x,\bar g_x^1,\dots,\bar g_x^s)$, $\bar\rho_j=\operatorname{rank}(\bar g_y^j,\bar u_y^j)$, $1\le j\le s$, and the sets $I_0,\dots,I_s$ contain indices of some columns (we identify the columns corresponding to the objective functions f and g^j, respectively, with the index 0) whose deletions diminish the corresponding ranks (cf. (38)). Hence, the intersection of $j^1_s(f,g,u)(\mathbb{R}^{n+m+1}_s)$ with $V_{s,\bar\rho_0,\dots,\bar\rho_s,I_0,\dots,I_s}$ is nonempty and, moreover, this intersection is transverse by the definition of $\tilde{\mathcal{F}}$. Now, application of (36) and (37) yields essentially the desired result.

In the following, we will show the implications of relation (37) only, since the treatment of the tangent spaces in (36) is tedious and would blow up the size of this outline (for details, cf. [23]). However, (37) yields already the main features of our type classification. Noting that $j^1_s(f,g,u)(\mathbb{R}^{n+m+1}_s)$ is a smooth manifold of dimension $s(n+m+1)$ in $J_s(n+m+1,3,1)$, relation (37) gives after a short computation

$$\text{(39)}\qquad \sum_{j=0}^{s}(d_j+|I_j|)+d_0(n-s+d_0+|I_0|)+\sum_{j=1}^{s}d_j(m-1+d_j+|I_j|)\ \le\ 1,$$

where each of the substituted variables $d_0=s-\rho_0$ and $d_j=1-\rho_j$, $1\le j\le s$, is nonnegative. The latter is due to the fact that $(\bar x,\bar t)$ is a g.c. point and that the

$\bar{y}^j$ are minimizers of the corresponding lower level problems. Consequently, the left-hand side of (39) is nonnegative and each of the numbers $d_j,\ |I_j|,\ 0 \le j \le s$ is contained in the set $\{0,1\}$, either none or exactly one of them attaining the value 1. A first implication of this result is that s cannot exceed $n+1$ by the nonnegativity of $(n-s+d_0+|I_0|)$ and hence, $|Y_0(\bar{x},\bar{t})| \le n+1$ in the generic case.

Next, consider the case in which d_0 vanishes or, equivalently, $\operatorname{rank}(\bar{f}_x, \bar{g}^1_x, \ldots, \bar{g}^s_x) = s$, so that $s \le n$ and the corresponding Lagrange multipliers $(\kappa, \mu_1, \ldots, \mu_s)$ are unique up to a common multiple. Omitting the lower level problems for a moment, we have

- LICQ and ND1 (cf. Definition 2.3) hold if and only if no multiplier vanishes, which is easily seen to be equivalent to $I_0 = \emptyset$.
- LICQ is violated if and only if κ vanishes, or equivalently $I_0 = \{0\}$. In this case, none of the μ_j vanish (i.e., "ND1 holds," loosely speaking).
- ND1 is violated if and only if $I_0 = \{j\}$ for a $j \in \{1,\ldots,s\}$. In this case, LICQ holds and exactly one of the μ_j vanishes.

In case $d_0 = 1$, we find $I_0 = \emptyset$ and $0 \le n-s+1 \le 0$, where the second inequality comes from (39). Thus, we have $s = n+1$ and $\bar{\rho}_0 = n$. Completing this analysis with analogous arguments for the lower level problems, we find that LICQ can be violated *at most once* in all occurring problems (upper and lower level), then forcing ND1 to hold in all problems, and vice versa. These observations yield the following preliminary type classification:

$$\begin{array}{rcl} d_0 = \cdots = d_s = 0,\ I_0 = \cdots = I_s = \emptyset & : & \text{type } 1'\text{or type } 3' \\ d_0 = 1 & : & \text{type } 5' \\ I_0 = \{0\} & : & \text{type } 4' \\ I_0 = \{j\},\ j \in \{1,\ldots,s\} & : & \text{type } 2' \\ I_j = \{0\},\ j \in \{1,\ldots,s\} & : & \text{type } 7'. \end{array}$$

Note that in the present setting of one equality constraint in the lower level, the cases $d_j = 1,\ j \in \{1,\ldots,s\}$, do not occur and that the cases $I_j = \{1\},\ j \in \{1,\ldots,s\}$, do not generate singularities. For the complete classification, the tangent space conditions (36) have to be computed explicitly for each of the above cases and, moreover, the manifolds $V_{s,\rho_0,\ldots,\rho_s,I_0,\ldots,I_s}$ have to be further refined, as to take second-order information (i.e., 2-jet extensions) into account. This construction yields the desired set $\mathcal{F}$.

5. Jumps and generalizations.

The analysis of the local structure of Σ around g.c. points of type 6, 7, and 8 in section 3 shows that, in generic one-parametric semi-infinite programming, the set of g.c. points can possess (relative) boundary points, as in contrast to *finite* problems. In fact, only one branch of g.c. points (of type 1) emanates from (or ends at) points of type 7 and type 8b. In particular, if we trace a path of local minimizers along Σ by a continuation method, the minimum is lost at these points. Note that at points of type 8a a path of local minimizers cannot stop (cf. Theorem 3.12(i)), and at points of type 6 it stops if and only if Σ exhibits a turning point there (cf. Theorem 3.6(iii)). At turning points of type 6, as well as at points of type 7 and type 8b, a feasible direction of descent can be constructed so that a jump to another path of local minimizers is possible at each of the "typically semi-infinite" singularities of Σ, provided that the feasible set $M(t)$ is contained in some compact set C for each t (cf. [2] for the finite case). For details, we refer to [16].

In this paper we focussed on the full nonlinear parametric semi-infinite case. Special subcases (such as the linear and the quadratic case, resp.) are of interest, too. However, their study within this paper would blow up the size considerably. On the other hand, these cases can be treated by combining the ideas presented in this paper with the work done in [19] (linear case) and [3] (quadratic case). Moreover, the linear case with regular index sets $Y(t)$ can be found in [21]. In order to apply our results to one-parametric problems with an objective function of maximum type, note that the nondifferentiable problem $\min_x \max_{y\in Y(t)} F(x,t,y)$ is equivalent to a (differentiable) $SIP(t)$, where the additional variable x_{n+1} is minimized, subject to the semi-infinite constraint $F(x,t,y) \le x_{n+1},\ y \in Y(t)$.

Our results extend in an obvious way to one-parametric problems with a finite number of semi-infinite inequality constraints, i.e., the feasible set is given by $\{x \in \mathbb{R}^n \mid h^i(x,t) = 0,\ i \in I,\ g^j(x,t,y) \ge 0,\ y \in Y_j(t),\ j \in J\}$ with $|J| < \infty$. On the other hand, generalized semi-infinite programming problems, where the index set Y depends on the variable x, give rise to a *nontrivial* modification of the generic-type classification for g.c. points. In [6] this classification is given for the case that LICQ always holds in the lower level problem.

Appendix. In this appendix we give the proofs of Theorems 3.2, 3.8, and 3.12, which are concerned with the local structure of the generalized critical point set Σ around points of type 6, 7, and 8, resp.

Proof of Theorem 3.2*, parts* (i), (ii), *and* (iv). We consider the equations

$$\begin{pmatrix} f_x(x,t) \ - \ \mu\, g_x(x,t,y) \\ -\ g(x,t,y) \\ g_y(x,t,y) \end{pmatrix} \ = \ 0, \qquad \begin{pmatrix} f_x(x,t) \ - \ \mu\, g_x(x,t,y) \\ -\ g(x,t,y) \\ g_y(x,t,y) \ - \ \gamma\, v_y(t,y) \\ -\ v(t,y) \end{pmatrix} \ = \ 0,$$

which define locally unique C^2-functions $\left(\tilde{x}^d(t), \tilde{\mu}^d(t), \tilde{y}^d(t)\right)$ around $(\bar{x}, \bar{\mu}, \bar{y})$ and $(\tilde{x}^e(t), \tilde{\mu}^e(t), \tilde{y}^e(t), \tilde{\gamma}^e(t))$ around $(\bar{x}, \bar{\mu}, \bar{y}, 0)$, respectively, since the corresponding Jacobians (with respect to (x,μ,y) and (x,μ,y,γ), resp.)

$$A_d \ = \ \begin{pmatrix} f_{xx} - \bar{\mu} g_{xx} & -g_x & -\bar{\mu} g_{xy} \\ -g_x^\top & 0 & 0 \\ g_{yx} & 0 & g_{yy} \end{pmatrix} \ \text{and} \ A_e \ = \ \left(\begin{array}{ccc|c} & & & 0 \\ & A_d & & 0 \\ & & & -v_y \\ \hline 0 & 0 & -v_y^\top & 0 \end{array}\right)$$

are nonsingular (all partial derivatives being evaluated at $(\bar{x}, \bar{t}, \bar{y})$). The latter fact is due to conditions (6.4.5), (6.6)* and (6.4.4), (6.5)*, resp., and is easily proved by using Schur complements. From the uniqueness of implicit functions we conclude the local identities

$$\begin{pmatrix} \tilde{x}^d(t) \\ \tilde{\mu}^d(t) \\ \tilde{y}^d(t) \end{pmatrix} \equiv \begin{pmatrix} x^d(t) \\ \mu^d(t) \\ y^d(x^d(t),t) \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \tilde{x}^e(t) \\ \tilde{\mu}^e(t) \\ \tilde{y}^e(t) \\ \tilde{\gamma}^e(t) \end{pmatrix} \equiv \begin{pmatrix} x^e(t) \\ \mu^e(t) \\ y^e(x^e(t),t) \\ \gamma^e(x^e(t),t) \end{pmatrix}.$$

Hence, in the sequel we omit the tildes, e.g., we write $y^d(t) = y^d(x^d(t),t)$. In particular, we obtain

$$\begin{aligned} \alpha &= \frac{d}{dt}\, v(t,\ y^d(t))|_{t=\bar{t}} \ = \ v_t(\bar{t},\bar{y}) \ + \ (v_y(\bar{t},\bar{y}))^\top \dot{y}^d(\bar{t})\ , \\ \delta &= \dot{\gamma}^e(\bar{t})\ . \end{aligned}$$

Now, a short calculation yields the equation

$$(40) \qquad A_e \begin{pmatrix} \dot{x}^e - \dot{x}^d \\ \dot{\mu}^e - \dot{\mu}^d \\ \dot{y}^e - \dot{y}^d \\ \delta \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \alpha \end{pmatrix}.$$

Condition (6.7)* and the last row (40) prove part (iv) of the assertion. Applying Cramer's rule to the entry δ in system (40) gives

$$(41) \qquad \delta \; = \; \alpha \cdot \frac{\det A_d}{\det A_e}$$

(where det denotes the determinant). This implies part (ii) in virtue of the nonsingularity of A_d and A_e. Now consider the case $\dot{x}^e - \dot{x}^d = 0$. Then, the third row of (40) yields

$$\dot{y}^e - \dot{y}^d \; = \; \delta \; g_{yy}^{-1} v_y$$

and thus

$$\alpha \; = \; -v_y^\top (\dot{y}^e - \dot{y}^d) \; = \; -\delta\vartheta,$$

where the value $\vartheta = v_y^\top g_{yy}^{-1} v_y$ is positive by (43) (see below). This proves part (i).

Proof of Theorem 3.2, *part* (iii). Now it remains to check whether the points $y^e(t)$ and $y^d(t)$ corresponding to $x^e(t)$ and $x^d(t)$ are actually minima of the lower level problems. Otherwise, $(x^e(t), t)$ and $(x^d(t), t)$ would not belong to Σ. Since $\bar{y}$ is a solution of $Q(\bar{x}, \bar{t})^*$ the second-order necessary condition together with $\bar{\gamma} = 0$ and condition (6.4.4) implies that the restricted Hessian

$$(42) \qquad g_{yy}(\bar{x}, \bar{t}, \bar{y}) \;|_{\text{Ker } v_y^\top(\bar{t}, \bar{y})} \;\text{ is positive definite.}$$

Furthermore, we conclude that the unrestricted Hessian

$$(43) \qquad g_{yy}(\bar{x}, \bar{t}, \bar{y}) \;\text{ is positive definite,}$$

too, because a feasible direction of descent is easily constructed otherwise. In fact, assume that (43) does not hold. Then we can choose a vector d with $v_y^\top d > 0$ and $d^\top g_{yy} d < 0$ by virtue of (42) and condition (6.4.5). But the vector d is a feasible direction of quadratic descent in $\bar{y}$, which contradicts the fact that $\bar{y}$ is a local minimizer for the problem $Q(\bar{x}, \bar{t})^*$. Hence, $\bar{y}$ is a nondegenerate local minimum of $Q(\bar{x}, \bar{t})_d^*$, and the assertion concerning $y^d(t)$ follows immediately.

From $\gamma^e(\bar{t}) = 0$ and $\delta \neq 0$ we know that $\gamma^e(t)$ does not vanish for t in a neighborhood of $\bar{t}$. The second-order sufficiency condition is satisfied because, for t close to $\bar{t}$, $\gamma^e(t)$ is close to zero, and thus, the restricted Hessian

$$g_{yy}(x^e(t), t, y^e(t)) - \gamma^e(t) v_{yy}(t, y^e(t)) \;|_{\text{Ker } v_y^\top(t, y^e(t))}$$

is positive definite by (42). □

Proof of Theorem 3.8. With $\zeta = (x, \mu, y, t)$ we consider the equations of first-order necessary conditions corresponding to problem $SIP(t)^*$ (note that, for the lower level problem, we have a Fritz-John condition):

$$(44) \qquad F(\alpha, \zeta) \; = \; \begin{pmatrix} f_x(x,t) - \mu g_x(x,t,y) \\ -g(x,t,y) \\ \alpha \; g_y(x,t,y) - w_y(t,y) \\ -w(t,y) \end{pmatrix} \; = \; 0.$$

By its block structure and by (16), (18) the Jacobian of (44) at $(0,\bar\zeta)$

$$F_\zeta(0,\bar\zeta) \;=\; \left(\begin{array}{cc|cc} f_{xx}-\bar\mu g_{xx} & -g_x & -\bar\mu g_{xy} & f_{xt}-\bar\mu g_{xt} \\ -g_x^\top & 0 & -g_y^\top & -g_t \\ \hline 0 & 0 & -w_{yy} & -w_{yt} \\ 0 & 0 & 0 & -w_t \end{array}\right)$$

is nonsingular. Hence, there exists a locally unique C^2-function

$$\zeta(\alpha) = (x(\alpha),\mu(\alpha),y(\alpha),t(\alpha))$$

satisfying $\zeta(0)=\bar\zeta$ and

$$F(\alpha,\zeta(\alpha)) \;\equiv\; 0\,. \tag{45}$$

Consequently, in a neighborhood U of $\bar z$ we have (with an $\varepsilon>0$):

$$\Sigma\cap U \;\subset\; \{(x(\alpha),t(\alpha)),\ \alpha\in(-\varepsilon,\varepsilon)\} \;=\; \Gamma.$$

Now we show that Γ is a C^2-manifold with a quadratic turning point at $\bar z$. Differentiation of (45) with respect to α yields $F_\alpha(0,\bar\zeta)+F_\zeta(0,\bar\zeta)\cdot\zeta_\alpha(0)=0$, which implies the equations

$$-\;g_x^\top\,x_\alpha \;-\; g_y^\top\,y_\alpha \;-\; g_t\,t_\alpha \;=\; 0, \tag{46}$$

$$g_y \;-\; w_{yy}\,y_\alpha \;-\; w_{yt}\,t_\alpha \;=\; 0, \tag{47}$$

$$-\; w_t\,t_\alpha \;=\; 0. \tag{48}$$

From (18) and (48) we have

$$t_\alpha \;=\; 0, \tag{49}$$

and (18), (47), and (49) imply

$$y_\alpha \;=\; w_{yy}^{-1}g_y. \tag{50}$$

Now, (46), (49), and (50) yield

$$g_x^\top x_\alpha \;=\; -g_y^\top w_{yy}^{-1}g_y,$$

from which we conclude with (19):

$$x_\alpha \;\neq\; 0. \tag{51}$$

Thus, the set $\Gamma=\{(x(\alpha),t(\alpha)),\ \alpha\in(-\varepsilon,\varepsilon)\}$ is a regularly parametrized C^2-curve and hence, a C^2-manifold of dimension one. Differentiating the identity

$$w(t(\alpha),y(\alpha)) \;\equiv\; 0 \quad \text{(compare (45))}$$

twice with respect to α and using (17), (49), and (50) we get

$$t_{\alpha\alpha} \;=\; -\delta \;\neq\; 0.$$

The assertion about the structure of Γ now follows from the fact that Γ can be regularly reparametrized by some of the variables x_i (recall (51)).

In the remainder of the proof we check for which α the points $y(\alpha)$ are minimizers of the lower level problem. Here, we treat only Case 2; i.e., the lower level problem takes the form

$$Q(x,t)^* \qquad \min\, g(x,t,y) \;\text{ subject to }\; w(t,y) \;\geq\; 0.$$

In Case 1, the proof runs along the same lines, but now second-order conditions for equality constrained problems have to be used. In that case, the constraint function w may be replaced by $-w$ in order to prove the assumption for *nonnegative* α.

Consider the lower level part of system (44)

$$G(\alpha,x,y,t) \;=\; \begin{pmatrix} \alpha\; g_y(x,t,y) - w_y(t,y) \\ -w(t,y) \end{pmatrix} \;=\; 0. \tag{52}$$

By (17) and (18) there are locally unique C^2-functions $\tilde{y}(\alpha,x)$ and $\tilde{t}(\alpha,x)$ with $\tilde{y}(0,\bar{x}) = \bar{y}$, $\tilde{t}(0,\bar{x}) = \bar{t}$ and $G(\alpha,x,\tilde{y}(\alpha,x),\tilde{t}(\alpha,x) \equiv 0$. In particular,

$$G(0,x,\tilde{y}(0,x),\tilde{t}(0,x)) \;\equiv\; 0 \tag{53}$$

holds locally around $\bar{x}$, and differentiating (53) with respect to x gives

$$\tilde{y}_x(0,0) \;=\; \tilde{t}_x(0,0) \;=\; 0. \tag{54}$$

Using (54) as well as (16), it is easily checked that the system

$$H(\alpha,x,\mu) \;=\; \begin{pmatrix} f_x(x,\tilde{t}(x,\alpha)) - \mu g_x(x,\tilde{t}(x,\alpha),\tilde{y}(x,\alpha)) \\ -g(x,\tilde{t}(x,\alpha),\tilde{y}(x,\alpha)) \end{pmatrix} \;=\; 0$$

defines locally unique C^2-functions $\tilde{x}(\alpha)$ and $\tilde{\mu}(\alpha)$ with $\tilde{x}(0) = \bar{x}$, $\tilde{\mu}(0) = \bar{\mu}$ and $H(\alpha,\tilde{x}(\alpha),\tilde{\mu}(\alpha)) \equiv 0$. From the uniqueness of implicit functions we conclude the local identity

$$\begin{pmatrix} \tilde{x}(\alpha) \\ \tilde{\mu}(\alpha) \\ \tilde{y}(\alpha,\tilde{x}(\alpha)) \\ \tilde{t}(\alpha,\tilde{x}(\alpha)) \end{pmatrix} \;\equiv\; \begin{pmatrix} x(\alpha) \\ \mu(\alpha) \\ y(\alpha) \\ t(\alpha) \end{pmatrix}. \tag{55}$$

Now it remains to show that for (α,x) in a sufficiently small neighborhood V of $(0,\bar{x})$, the point $\tilde{y}(\alpha,x)$ is a solution of $Q(x,\tilde{t}(\alpha,x))^*$ if and only if $\alpha > 0$. Then, exactly the points $(x(\alpha),t(\alpha))$ with nonnegative α belong to Σ, because $(x(0),t(0)) = (\bar{x},\bar{t})$ itself is assumed to be a g.c. point, and by (55) there is a local minimizer $y(\alpha)$ of the lower level problem corresponding to $x(\alpha)$ with nonvanishing α if and only if $\alpha > 0$.

Let V be a neighborhood of $(0,\bar{x})$ and fix $(\alpha,x) \in V$ with $\alpha < 0$. From (52) we have

$$g_y(x,\tilde{t}(\alpha,x),\tilde{y}(\alpha,x)) \;=\; \frac{1}{\alpha} w_y(\tilde{t}(\alpha,x),\tilde{y}(\alpha,x)). \tag{56}$$

Because $w_{yy}(\bar{t},\bar{y})$ is regular, $w_y(\tilde{t}(\alpha,x),\tilde{y}(\alpha,x))$ does not vanish and hence, LICQ holds for the problem $Q(x,\tilde{t}(\alpha,x))^*$. This implies that the negative multiplier $\frac{1}{\alpha}$ is uniquely determined and the Karush–Kuhn–Tucker condition is violated. So, $\tilde{y}(\alpha,x)$ is not a solution of $Q(x,\tilde{t}(\alpha,x))^*$.

Now, let $(\alpha, x) \in V$ with $\alpha > 0$. Then, we have (56) again, but with a uniquely determined positive multiplier. From the second-order necessary condition of Fritz-John type (cf. [4]) we conclude

$$\xi^\top w_{yy}(\bar{t}, \bar{y})\ \xi\ <\ 0 \quad \text{for all } \xi \in \text{Ker } g_y^\top(\bar{x}, \bar{t}, \bar{y}) \setminus \{0\},$$

where the strictness of the inequality is due to the regularity of w_{yy}. By (56) we have Ker $g_y^\top(x, \tilde{t}(\alpha, x), \tilde{y}(\alpha, x)) =$ Ker $w_y^\top(\tilde{t}(\alpha, x), \tilde{y}(\alpha, x))$ and thus, for sufficiently small V,

$$\xi^\top w_{yy}(\tilde{t}(\alpha, x), \tilde{y}(\alpha, x))\ \xi\ <\ 0 \quad \text{for all } \xi \in \text{Ker } w_y^\top(\tilde{t}(\alpha, x), \tilde{y}(\alpha, x)) \setminus \{0\}.$$

For α sufficiently close to zero we obtain

$$\begin{gathered}\xi^\top \left(g_{yy}(x, \tilde{t}(\alpha, x), \tilde{y}(\alpha, x)) - \tfrac{1}{\alpha} w_{yy}(\tilde{t}(\alpha, x), \tilde{y}(\alpha, x))\right)\ \xi\ >\ 0 \\ \text{for all } \xi \in \text{Ker } w_y^\top(\tilde{t}(\alpha, x), \tilde{y}(\alpha, x)) \setminus \{0\}\end{gathered}$$

and hence, the second-order sufficiency condition for $\tilde{y}(\alpha, x)$ to be a solution of $Q(x, \tilde{t}(\alpha, x))^*$ is satisfied. We conclude that

$$\Sigma \cap U\ =\ \{(x(\alpha), t(\alpha)),\ \alpha \in [0, \varepsilon)\}\ . \qquad \square$$

Proof of Theorem 3.12. With $\zeta^i = (x, \mu, y, \gamma^i)$ we consider the equations of first-order necessary conditions corresponding to the problems $SIP(t)_i^*$, $i \in \{1, 2\}$:

$$F^i(t, \zeta^i)\ =\ \begin{pmatrix} f_x(x, t) - \mu g_x(x, t, y) \\ -g(x, t, y) \\ g_y(x, t, y) - \gamma^i v_y^i(t, y) \\ -v^i(t, y) \end{pmatrix}\ =\ 0 \tag{57}$$

with the Jacobian with respect to ζ^i at $(\bar{t}, \bar{\zeta}^i)$

$$A_i\ =\ F^i_{\zeta^i}(\bar{t}, \bar{\zeta}^i)\ =\ \begin{pmatrix} f_{xx} - \bar{\mu} g_{xx} & -g_x & -\bar{\mu} g_{xy} & 0 \\ -g_x^\top & 0 & -g_y & 0 \\ g_{yx} & 0 & g_{yy} - \bar{\gamma}^i v_{yy}^i & -v_y^i \\ 0 & 0 & -v_y^i & 0 \end{pmatrix},$$

all partial derivatives being evaluated at $(\bar{x}, \bar{t}, \bar{y})$. Equations (23) and (31) imply that A_i is nonsingular. Hence, there exist locally unique C^2-functions

$$\zeta^i(t)\ =\ (\tilde{x}^i(t), \tilde{\mu}^i(t), \tilde{y}^i(t), \tilde{\gamma}^i(t))$$

satisfying $\zeta^i(\bar{t}) = \bar{\zeta}^i$ and $F^i(t, \zeta^i(t))\ \equiv\ 0$. In particular, we obtain

$$A_i \cdot \dot{\zeta}^i\ =\ \begin{pmatrix} -f_{xt} + \bar{\mu} g_{xt} \\ g_t \\ -g_{yt} + \bar{\gamma}^i v_{yt}^i \\ v_t^i \end{pmatrix}. \tag{58}$$

Furthermore, from the uniqueness of implicit functions we conclude the local identities

$$\begin{pmatrix} \tilde{x}^i(t) \\ \tilde{\mu}^i(t) \\ \tilde{y}^i(t) \\ \tilde{\gamma}^i(t) \end{pmatrix}\ \equiv\ \begin{pmatrix} x^i(t) \\ \mu^i(t) \\ y^i(x^i(t), t) \\ \gamma^i(x^i(t), t) \end{pmatrix}, \quad i \in \{1, 2\}.$$

Omitting the tildes, this yields

$$\frac{d}{dt}\, v^1(t, y^2(t)) \mid_{\bar{t}} \;=\; v^1_t \;+\; v^1_y \cdot \dot{y}^2 \;\overset{(58),(25)}{=}\; v^1_t - v^1_y\, \frac{v^2_t}{v^2_y},$$

as well as

$$\frac{d}{dt}\, v^2(t, y^1(t)) \mid_{\bar{t}} \;=\; v^2_t - v^2_y\, \frac{v^1_t}{v^1_y}\ .$$

Thus, $y^2(t)$ is feasible in the lower level problem for $t \geq 0$ $(t \leq 0)$ according to $\alpha_1 = +1$ (-1), and $y^1(t)$ is feasible for $t \geq 0$ $(t \leq 0)$ according to $\alpha_2 = +1$ (-1). The set Ω of generalized critical points in the lower level problem is locally composed by means of the C^2-curves $t \longmapsto (t, y^1(t))$ and $t \longmapsto (t, y^2(t))$, where exactly the branches with feasible $y^1(t)$ and $y^2(t)$, respectively, belong to Ω. Moreover, Ω exhibits a turning point at $(\bar{t}, \bar{y})$ if and only if $\alpha_1 \cdot \alpha_2 = -1$, i.e., in case that $\bar{z}$ is of type 8b. The branches meet under a nonvanishing angle since the last row of (58) implies

$$\dot{y}^1 - \dot{y}^2 \;=\; \frac{v^2_t v^1_y - v^1_t v^2_y}{v^1_y\ v^2_y} \;\overset{(24)}{\neq}\; 0.$$

Using the second row of (58) we also see that

$$g_x^\top (\dot{x}^1 - \dot{x}^2) \;=\; -g_y \cdot (\dot{y}^1 - \dot{y}^2) \;\overset{(26)}{\neq}\; 0.$$

Hence, the curves $t \longmapsto (x^1(t), t)$ and $t \longmapsto (x^2(t), t)$ meet in $\bar{z}$ under a nonvanishing angle, too. The preceding observations show that at most one branch of each graph belongs to Σ.

In the remainder of the proof we check whether the points $y^i(t)$ corresponding to $x^i(t)$ locally around $\bar{t}$ are minimizers of the lower level problem.

First, let $\bar{z} = (\bar{x}, \bar{t})$ be a g.c. point of type 8a. Since MFCQ is satisfied in the lower level problem $Q(\bar{x}, \bar{t})^*$ we have $v^1_y \cdot v^2_y > 0$. Furthermore,

$$g_y \cdot v^i_y > 0 \tag{59}$$

holds for both $i = 1$ and $i = 2$ because $g_y \cdot v^i_y < 0$ for one $i \in \{1, 2\}$ implies $g_y \cdot v^i_y < 0$ for *both* i, which is easily shown to be a contradiction to the Fritz-John first-order necessary condition for $\bar{y}$ to be a local minimizer of $Q(\bar{x}, \bar{t})^*$. From Lemma 3.10 we already know that $\bar{y}$ is a nondegenerate critical point both for $Q(\bar{x}, \bar{t})^*_1$ and $Q(\bar{x}, \bar{t})^*_2$. By (25) and the fact that we deal with one-dimensional problems, the corresponding tangent spaces are zero spaces, and we obtain that $\bar{y}$ is a nondegenerate local minimum both for $Q(\bar{x}, \bar{t})^*_1$ and $Q(\bar{x}, \bar{t})^*_2$, just by the fact that

$$g_y \;=\; \eta_i v^i_y$$

has a solution $\eta_i > 0$ (compare (59)). Hence, the points $y^i(x, t)$, $i \in \{1, 2\}$ are local minimizers of $Q(x, t)^*_i$ for (x, t) in a neighborhood of $(\bar{x}, \bar{t})$. This implies that Ω is the composition of two branches of minima, not exhibiting a turning point at $(\bar{t}, \bar{y})$, and the assertion concerning the local structure of Σ in part (i) follows immediately. By condition (8.2), the linear indices LI and LCI do not change when passing $\bar{z}$ along Σ. From equations (32), (33), (34) and the remark following thereafter it is easily deduced that the quadratic indices QI and QCI do not change either.

Now, consider a g.c. point $\bar{z} = (\bar{x}, \bar{t})$ of type 8b. The same argument as above shows that there are $i, j \in \{1, 2\},\ i \neq j$, with

$$g_y \cdot v_y^i > 0 \text{ and } g_y \cdot v_y^j < 0.$$

From (29) we know that $j = l^*$. Along the same lines as above we obtain that $\bar{y}$ is a nondegenerate local minimum for $Q(\bar{x}, \bar{t})_i^*$ and a nondegenerate local *maximum* for $Q(\bar{x}, \bar{t})_{l^*}^*$. Thus, Ω is the composition of one branch of minima and one branch of maxima, exhibiting a turning point at $(\bar{t}, \bar{y})$. Since $y^{l^*}(t)$ is a local maximizer, $(x^{l^*}(t), t)$ does not belong to Σ. The remainder of assertion (ii) follows from the definition of α_i. □

Acknowledgment. We would like to thank two anonymous referees for their precise and valuable remarks.

REFERENCES

[1] C. Berge, *Topological Spaces*, Oliver and Boyd, London, 1963.

[2] J. Guddat and D. Nowack, *Parametric optimization: Pathfollowing and jumps in the set of local minimizers and in the critical set*, in Parametric Optimization and Related Topics II, J. Guddat, H. Th. Jongen, B. Kummer, and F. Nožička, eds., Akademie Verlag, Berlin, 1991, pp. 76–111.

[3] M. Henn, P. Jonker, and F. Twilt, *On the critical sets of one-parameter quadratic optimization problems*, in Recent Developments in Optimization, Dijon, 1994, Lecture Notes in Econom. and Math. Systems, 429, Springer-Verlag, Berlin, 1995, pp. 183–197.

[4] R. Hettich and H. Th. Jongen, *On first and second order conditions for local optima for optimization problems in finite dimensions*, Methods Oper. Res., 23 (1977), pp. 82–97.

[5] R. Hettich and H. Th. Jongen, *Semi-infinite programming: Conditions of optimality and applications*, in Optimization Techniques, Part 2, Lecture Notes in Control and Inform. Sci. 7, J. Stoer, ed., Springer-Verlag, Berlin, Heidelberg, New York, 1978, pp. 1–11.

[6] R. Hettich, H. Th. Jongen, and O. Stein, *On Continuous Deformations of Semi-infinite Optimization Problems*, in Approximation and Optimization in the Caribbean II, M. Florenzano, J. Guddat, M. Jimenez, H. Th. Jongen, G. Lopez Lagomasino, and F. Marcellan, eds., Peter Lang Verlag, Frankfurt a. M., 1995, pp. 406–424.

[7] R. Hettich and K. O. Kortanek, *Semi-infinite programming: Theory, methods, and applications*, SIAM Rev., 35 (1993), pp. 380–429.

[8] R. Hettich and P. Zencke, *Numerische Methoden der Approximation und semi-infiniten Optimierung*, Teubner Studienbücher, Stuttgart, 1982.

[9] M. W. Hirsch, *Differential Topology*, Grad. Texts in Math. 33, Springer-Verlag, Berlin, 1976.

[10] H. Th. Jongen, P. Jonker, and F. Twilt, *On one-parameter-families of sets defined by (in)equality constraints*, Nieuw Arch. Wisk. (3), 30 (1982), pp. 307–322.

[11] H. Th. Jongen, P. Jonker, and F. Twilt, *Nonlinear Optimization in* $\mathbb{R}^n$. I. *Morse Theory, Chebyshev Approximation*, Peter Lang Verlag, Frankfurt a. M., Bern, New York, 1983.

[12] H. Th. Jongen, P. Jonker, and F. Twilt, *Nonlinear Optimization in* $\mathbb{R}^n$. II. *Transversality, Flows, Parametric Aspects*, Peter Lang Verlag, Frankfurt a. M., Bern, New York, 1986.

[13] H. Th. Jongen, P. Jonker, and F. Twilt, *One-parameter families of optimization problems: Equality constraints*, J. Optim. Theory Appl., 48 (1986), pp. 141–161.

[14] H. Th. Jongen, P. Jonker, and F. Twilt, *Critical sets in parametric optimization*, Math. Programming, 34 (1986), pp. 333–353.

[15] H. Th. Jongen, J.-J. Rückmann, and G.-W. Weber, *One-parametric semi-infinite optimization: On the stability of the feasible set*, SIAM J. Optim., 4 (1994), pp. 637–648.

[16] H. Th. Jongen and O. Stein, *Parametric semi-infinite programming: Jumps in the set of local minimizers*, in Parametric Optimization and Related Topics IV, J. Guddat, H. Th. Jongen, F. Nožička, G. Still, and F. Twilt, eds., Peter Lang Verlag, Frankfurt a. M., 1997, pp. 161–175.

[17] H. Th. Jongen and G. Zwier, *On the local structure of the feasible set in semi-infinite optimization*, in Parametric Optimization and Approximation, Internat. Ser. Numer. Math. 72, F. Brosowski and F.Deutsch, eds., Birkhäuser-Verlag, Basel, 1984.

[18] M. Kojima and R. Hirabayashi, *Continuous deformations of nonlinear programs*, Math. Programming Study, 21 (1984), pp. 150–198.

[19] D. D. Pateva (Dentcheva), *On the singularities in linear one-parametric optimization problems*, Optimization, 22 (1991), pp. 193–220.

[20] A. B. Poore and C. A. Tiahrt, *Bifurcation problems in nonlinear parametric programming*, Math. Programming, 39 (1987), pp. 189–206.

[21] T. Rupp, *Kontinuitätsmethoden zur Lösung einparametrischer semi-infiniter Optimierungsprobleme*, thesis, University of Trier, Germany, 1988.

[22] T. Rupp, *Kuhn-Tucker curves for one-parametric semiinfinite programming*, Optimization, 20 (1989), pp. 61–77.

[23] O. Stein, *On Parametric Semi-infinite Optimization*, Shaker-Verlag, Aachen, 1997.

[24] C. A. Tiahrt and A. B. Poore, *A bifurcation analysis of the nonlinear parametric programming problem*, Math. Programming, 47 (1990), pp. 117–141.

[25] G. Zwier, *Structural Analysis in Semiinfinite Programming*, thesis, University of Twente, The Netherlands, 1987.

SIAM J. OPTIM.
Vol. 7, No. 4, pp. 1138–1151, November 1997

014

STABILITY THEORY FOR LINEAR INEQUALITY SYSTEMS II: UPPER SEMICONTINUITY OF THE SOLUTION SET MAPPING[*]

M. A. GOBERNA[†], M. A. LÓPEZ[†], AND M. I. TODOROV[‡]

Abstract. This paper deals with the upper semicontinuity of the solution set mapping for linear inequality systems, complementing a previous work on lower semicontinuity and related stability concepts. The main novelty of our approach is that we are not assuming any standard hypothesis about the set indexing the inequalities in the system. This set, possibly infinite, has no topological structure and, therefore, the functional dependence between the linear inequalities and their associated indices has no qualification at all. The space of consistent systems, over a fixed index set, is endowed with the uniform topology derived from the pseudometric of Chebyshev, which turns out to be a natural way to measure the size of the perturbations. In this context, we provide some necessary and some sufficient conditions for the upper semicontinuity of the feasible set map at a given system whose solution set is not necessarily bounded.

Key words. convex analysis, stability theory, linear inequality systems, feasible set mapping, upper semicontinuity, semi-infinite programming

AMS subject classifications. 65F99, 15A39, 49D39, 52A40

PII. S105262349528901X

1. Introduction.

In this paper we consider systems of possibly infinitely many linear inequalities, in the Euclidean space $\Re^n$, of the form $\sigma = \{a_t'x \geq b_t, t \in T\}$, where T is a fixed nonempty arbitrary index set, $a_t \in \Re^n$ and $b_t \in \Re$, for all $t \in T$, a_t' denotes the transpose of a_t, and $a_t'x$ represents the inner product of a_t and x. If we denote by Θ the set of all the systems, in $\Re^n$, whose index set is T, the *solution set mapping,* $\mathcal{F} : \Theta \rightsquigarrow \Re^n$, assigns to each system $\sigma \in \Theta$ its corresponding *solution set* (also called *feasible set* in optimization), which is represented hereafter by F (i.e., $\mathcal{F}(\sigma) = F$). Since these infinite linear systems arise, in a rather natural form, closely connected with problems in functional approximation, numerical analysis, optimal control theory, semi-infinite programming, etc., many authors have approached the stability properties of their solution sets, extending some well-known theories developed in the finite context. In particular, the semi-infinite optimization model provides the principal motivation for studying the topics this paper deals with as far as continuity properties of the feasible set mapping have a strong influence on the stability features of the whole problem (upper semicontinuity of the optimal set mapping, continuity of the optimal value function, Hadamard well-posedness, etc.).

To start with, Robinson, in [10], stated that a system σ is stable under small perturbations if and only if $\mathcal{F}$ is lower semicontinuous (LSC, for short) at σ. This assertion motivates the study of this property. In [6, 7] different characterizations of the lower semicontinuity property are supplied, connecting it with other stability concepts [10, 12].

[*] Received by the editors July 10, 1995; accepted for publication (in revised form) July 23, 1996. This work was partially supported by grants MM-408/94 of the Bulgarian Ministry of Education and Science, PB93-0943 of the Spanish DGICYT, and GV-2219/94 and GV-2294/94 of the Generalitat Valenciana.

http://www.siam.org/journals/siopt/7-4/28901.html

[†] Department of Statistics and Operations Research, Faculty of Sciences, 03071 Alicante, Spain (mgoberna@ua.es, marco.antonio@ua.es).

[‡] Bulgarian Academy of Sciences, Institute of Mathematics, 29 Ph. Macedonsky Str., r. 401, 4002 Plovdiv, Bulgaria (todorovm@bgearn.aead.bg).

The other also very essential part of the continuity analysis concerns the upper semicontinuity. Since the continuity properties always depend on the topologies considered in the parameter and in the range spaces, the various results that can be found in the literature differ from each other. Nevertheless, most of the previous works consider that T is a compact Hausdorff set, $a(t) \equiv a_t \in C(T, \Re^n)$, $b(t) \equiv b_t \in C(T, \Re)$, and, accordingly, the parameter space of all the systems σ satisfying these conditions is a Banach space. In this particular setting, Brosowski [4] gives a necessary and sufficient condition for $\mathcal{F}$ to be upper semicontinuous (USC) at σ. This condition is that F must be either bounded or the whole space $\Re^n$. Helbig [9], in the context of disjunctive optimization, analyzes a more general case, with T being an arbitrary topological space but keeping the continuity of the functions $a(t)$ and $b(t)$. More related to our formulation is the paper of Greenberg and Pierskalla [8], in which no condition is posed on the index set T and on the parameter functions $a(t)$ and $b(t)$. Their approach refers to the use of the sup-function and leads to a sufficient condition for upper semicontinuity, which is also connected with the compactness of the feasible set in a neighborhood of σ.

As in [6, 7, 8], we formulate our system in its most general setting, i.e., T is an arbitrary index set which is required to be neither a finite set nor a topological space, and $a(t)$ and $b(t)$ are arbitrary functions. In order to measure the size of the perturbations of our system $\sigma \in \Theta$, we introduce a pseudometric on the *parameter space* Θ. For any pair of systems, in Θ, $\sigma = \{a_t'x \geq b_t, t \in T\}$ and $\sigma_1 = \{c_t'x \geq d_t, t \in T\}$, we define the *pseudodistance*

$$d(\sigma_1, \sigma) := \sup\nolimits_{t \in T} \left\| \begin{pmatrix} c_t \\ d_t \end{pmatrix} - \begin{pmatrix} a_t \\ b_t \end{pmatrix} \right\|,$$

where $\|.\|$ is the Chebyshev norm (i.e., $\|x\| = \max\{|x_i|, i = 1, ..., p\}$, when $x = (x_1, x_2, ..., x_p)' \in \Re^p$). In this way, (Θ, d) turns out to be a pseudometric space, whose topology is Hausdorff, satisfies the first axiom of countability, and describes the uniform convergence in $\Theta \equiv (\Re^{n+1})^T$.

The main goal of the present paper is to provide some conditions for the upper semicontinuity of $\mathcal{F}$ at a particular consistent ($F \neq \emptyset$) system σ. Recall that $\mathcal{F}$ is USC at $\sigma \in \Theta$ (in the classical Berge sense) if, for each open set W containing F, there exists an open set V, $\sigma \in V \subset \Theta$, such that if $\sigma_1 \in V$ its feasible set, F_1, will be also contained in W.

2. Preliminaries. We shall set out the relevant terminology and some preliminary results as well. The origin or null vector in the Euclidean space $\Re^n$ will be denoted by 0_n and, given a nonempty set X in this space, we denote by aff(X), dim(X), conv(X), cone(X), and X^o the *affine hull* of X, the *dimension* of X (i.e., the dimension of aff(X)), the *convex hull* of X, the *convex cone* spanned by $X \cup \{0_n\}$, and the *dual cone* of X (i.e., $X^o := \{y \in \Re^n \mid y'x \geq 0 \text{ for all } x \in X\}$), respectively.

From the topological side, int(X), rint(X), cl(X), bd(X), and rbd(X) represent the *interior*, the *relative interior*, the *closure*, the *boundary*, and the *relative boundary* of X, respectively. Finally, we shall use B for representing the *open unit ball* in $\Re^n$ for the chosen norm.

We define the *asymptotic cone* of X, denoted by X_∞, as the set of all the limits of the form $\lim_{k \to \infty} \lambda_k x_k$, where $\lambda_k \in \Re_+$, $x_k \in X$, $k = 1, 2, ...$, and $\lambda_k \downarrow 0$.

Next we state the properties of X_∞, which are used throughout the paper.

LEMMA 2.1. *Given a nonempty set X, $X \subset \Re^n$, its asymptotic cone X_∞ has the following properties:*

(i) X_∞ *is a closed cone.*

(ii) *If X is convex, then X_∞ is also convex.*

(iii) *If X is a closed convex set, X_∞ coincides with the set of directions y such that the half-line $\{x + \lambda y \mid \lambda \geq 0\}$, for a certain $x \in X$ (equivalently, for all $x \in X$), is completely contained in X.*

(iv) *If X contains a cone K, then $K \subset X_\infty$.*

(v) *X is bounded if and only if $X_\infty = \{0_n\}$.*

(vi) *$y \in X_\infty$ and $\|y\| = 1$ if and only if there exists a sequence $\{x_k\} \subset X$ such that $\lim_{k\to\infty} \|x_k\| = \infty$ and $\lim_{k\to\infty} \|x_k\|^{-1} x_k = y$.*

(vii) $X_\infty = (X_\infty)_\infty$.

(viii) *If $Y \subset \Re^n$ is a bounded set, then $(X + Y)_\infty = X_\infty$.*

(ix) *Let $X_1, X_2, ..., X_m$ be nonempty closed sets, in $\Re^n$, such that the following condition holds: if $y_1, y_2, ..., y_m$ are vectors satisfying $y_i \in (X_i)_\infty$, $i = 1, 2, ..., m$, and $y_1 + y_2 + \cdots + y_m = 0_n$, then y_i must be zero for $i = 1, 2, ..., m$. Then the set $X_1 + X_2 + \cdots + X_m$ will be closed.*

Proof. The proofs of statements (iii) and (ix) can be found in [11, Theorem 8.2] and [3, Corollary 2.41], respectively. Statement (vii) is a consequence of (i) and (iv). The proofs of the remaining propositions are left to the reader. □

As a consequence of (iv) and of the definition of an asymptotic cone, if X is a cone (not necessarily convex), we have $X \subset X_\infty \subset \mathrm{cl}(X)$.

When X is convex, X_∞ is the well-known *recession cone* (see [11] for the concepts related to convex analysis).

We associate with $\sigma = \{a_t'x \geq b_t, t \in T\} \in \Theta$ the so-called *moment cone* $M := \mathrm{cone}(A)$ and the cone $P := \mathrm{conv}(A_\infty)$, where $A := \{a_t, t \in T\}$. Most of the results presented in this paper come through the relationship between both cones, which is illustrated in the following lemma. Hereafter, when various systems are simultaneously considered, they and their associated sets are distinguished by means of subindices ($\sigma_i \to F_i, M_i, P_i$, etc.).

Lemma 2.2. *Let us consider the system $\sigma = \{a_t'x \geq b_t, t \in T\} \in \Theta$ .Then the following propositions hold:*

(i) $P \subset \mathrm{cl}(M)$.

(ii) $P = \{0_n\}$ *if and only if A is bounded.*

(iii) *If $\mathrm{cl}(M)$ is pointed, i.e., it does not contain a complete line, then P is closed.*

(iv) *If $d(\sigma_1, \sigma)$ is finite, then $P_1 = P$.*

(v) *If σ_1 and σ are consistent, $d(\sigma_1, \sigma)$ is finite, and F_∞ is strictly contained in $(F_1)_\infty$, then one has $P \neq \mathrm{cl}(M)$.*

Proof. (i) comes from the definition of asymptotic cone, and (ii) is a straightforward consequence of Lemma 2.1(v).

(iii) Carathéodory's theorem leads us to

$$P = A_\infty + A_\infty + \cdots^{(m+1)} \cdots + A_\infty,$$

where $m := \dim(A_\infty)$.

Next we prove that if $y_1, y_2, ..., y_{m+1}$ are points in $(A_\infty)_\infty = A_\infty$ (according to Lemma 2.1(vii)) such that $y_1 + y_2 + \cdots + y_{m+1} = 0_n$, then y_i must be zero, $i = 1, 2, ..., m + 1$. Then Lemma 2.1(i, ix) will be applied to conclude that P is closed. Actually, if we assume without loss of generality that $y_1 \neq 0_n$, we shall get

$$y_0 := y_2 + y_3 + \cdots + y_{m+1} = -y_1 \neq 0_n.$$

Since $y_1 \in A_\infty \subset \mathrm{cl}(M)$, $y_0 \in A_\infty + \cdots^{(m)} \cdots + A_\infty \subset \mathrm{cl}(M)$, and $y_0 + y_1 = 0_n$, we obtain a contradiction with the pointedness property assumed for cl(M).

(iv) If $d(\sigma_1, \sigma) = \alpha$, we have $A_1 \subset A + \alpha\mathrm{cl}(B)$ and $A \subset A_1 + \alpha\mathrm{cl}(B)$, and Lemma 2.1(viii) yields

$$(A_1)_\infty \subset \{A + \alpha\mathrm{cl}(B)\}_\infty = A_\infty \text{ and } (A)_\infty \subset \{A_1 + \alpha\mathrm{cl}(B)\}_\infty = (A_1)_\infty.$$

We have obtained $(A_1)_\infty = A_\infty$ and, hence, $P_1 = P$.

(v) It is well known that $F_\infty = M^o$ and $(F_1)_\infty = (M_1)^o$. Then the hypothesis is equivalent to asserting that M^o is strictly contained in $(M_1)^o$. Thus, $\mathrm{cl}(M) = M^{oo}$ contains strictly $\mathrm{cl}(M_1) = (M_1)^{oo}$, and the propositions (i) and (iv) above make $P = \mathrm{cl}(M)$ impossible. □

The following example shows that P does not need to be closed.

Example 2.3. Let us consider the system, in $\Re^3$, for which A is the algebraic product of [0,∞) by the set

$$\left\{ \begin{pmatrix} 1 \\ \cos t \\ 1 + \sin t \end{pmatrix}, t \in [0, 2\pi]; \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} \right\}.$$

It is evident that $A_\infty = A$ and $P = \{a \subset \Re^3 \mid a_3 > 0 \text{ or } a_2 = a_3 = 0\}$.

Let us introduce a new system, related to σ, which plays a crucial role in our approach. According to Lemma 2.1(vi), if $a \in A_\infty$ and $\|a\| = 1$, there will exist a sequence $\{t_k\} \subset T$ such that $\lim_{k\to\infty} \|a_{t_k}\| = \infty$ and $\lim_{k\to\infty} \|a_{t_k}\|^{-1} a_{t_k} = a$. If, additionally, $b := \mathrm{limsup}_{k\to\infty} \|a_{t_k}\|^{-1} b_{t_k}$ is finite, $a'x \geq b$ is said to be an *implicit fixed constraint* for σ. We call *asymptotic system,* associated with the consistent system σ, the one formed by all the implicit fixed constraints. The asymptotic system will be represented by σ^a, and F^a will be its solution set. Obviously, $F \subset F^a$ and, if A is bounded, σ^a will be an empty system, in which case we define $F^a = \Re^n$. Moreover, if $\{b_t, t \in T\}$ is bounded, we obtain

$$\sigma^a = \{a'x \geq 0,\ a \in A_\infty \cap \mathrm{bd}(B)\}.$$

In general, if $a'x \geq b$ is an implicit fixed constraint, one has $a \in A_\infty \cap \mathrm{bd}(B)$ and

$$(F^a)_\infty \supset \{A_\infty \cap \mathrm{bd}(B)\}^o = P^o.$$

LEMMA 2.4. *If σ_1 and σ are two systems such that $d(\sigma_1, \sigma)$ is finite, then one has $(\sigma_1)^a = \sigma^a$; i.e., the inequalities in σ^a become implicit fixed constraints for any system obtained by finite-sized perturbation.*

Proof. We shall write $\sigma = \{a_t'x \geq b_t, t \in T\}$ and $\sigma_1 = \{(a_t + u_t)'x \geq b_t + v_t, t \in T\}$, with $u_t \in \Re^n$, $v_t \in \Re$, and $\sup_{t\in T}\max\{\|u_t\|, |v_t|\} = d(\sigma_1, \sigma) < \infty$. It can be easily checked that $a'x \geq b$ is an implicit fixed constraint for σ if and only if

$$\|a\| = 1 \text{ and } \begin{pmatrix} a \\ b \end{pmatrix} \in \left\{ \begin{pmatrix} a_t \\ b_t \end{pmatrix}, t \in T \right\}_\infty.$$

Since

$$\left\{ \begin{pmatrix} a_t \\ b_t \end{pmatrix}, t \in T \right\}_\infty = \left\{ \begin{pmatrix} a_t + u_t \\ b_t + v_t \end{pmatrix}, t \in T \right\}_\infty$$

(as in the proof of Lemma 2.2(iv)), the conclusion follows trivially. □

The following example illustrates this concept of asymptotic systems and shows how much it depends on the chosen representation.

Example 2.5. Let us consider the closed convex set F obtained by the intersection of the upper half-planes associated with the tangent lines to the circle $x = (x_1, x_2)' = (2 + \cos t, 2 + \sin t)'$ at those points corresponding to $t \in [\pi, 3\pi/2]$. Three different representations of F are considered.

(1) $\sigma_1 := \{-(\cos t)x_1 - (\sin t)x_2 \geq -2(\cos t + \sin t) - 1,\ t \in [\pi, 3\pi/2]\}$.
Now $(\sigma_1)^a$ is empty (or $(F_1)^a = \Re^n$).
(2) $\sigma_2 := \sigma_1 \cup \{sx_1 + sx_2 \geq s(4 - \sqrt{2}), s \in [1, \infty)\}$.
In this case $(\sigma_2)^a = \{x_1 + x_2 \geq 4 - \sqrt{2}\}$.
(3) $\sigma_3 := \sigma_1 \cup \{sx_1 \geq s - 1, s \in \Re_+\} \cup \{ux_2 \geq u - 1, u \in \Re_+\}$.
For this third representation $(\sigma_3)^a = \{x_1 \geq 1, x_2 \geq 1\}$.

3. Conditions based upon perturbations of the solution set. The following characterization of upper semicontinuity requires, in the case of an unbounded initial solution set, that the perturbed solution sets differ from the original one in some *uniformly bounded manner.* The result comes through the standard Dolecki's characterization of upper semicontinuity for mappings between metric spaces (see, for instance, [2, Lemma 2.2.2]). We shall only approach the consistent case since, otherwise, $\mathcal{F}$ is USC at the inconsistent system σ ($F = \emptyset$) if and only if there exists a neighborhood of σ containing exclusively inconsistent systems. On the opposite side, when $F = \Re^n$ (which implies $A = \{0_n\}$), $\mathcal{F}$ is trivially USC at σ.

THEOREM 3.1. *Given a consistent system σ, $\mathcal{F}$ is USC at σ if and only if there exist two positive scalars, ε and ρ, such that*

$$F_1 \setminus \rho\text{cl}(B) \subset F \setminus \rho\text{cl}(B)$$

for every $\sigma_1 \in \Theta$ such that $d(\sigma_1, \sigma) < \varepsilon$.

Proof. First, observe that (Θ, d'), with $d'(\sigma_1, \sigma) := \min\{1, d(\sigma_1, \sigma)\}$, is a complete metric space, which is locally equivalent to (Θ, d) (providing also the topology of the uniform convergence on Θ).

Let us suppose that $\mathcal{F}$ is USC at σ. If there are no such scalars, ε and ρ, we shall take $\varepsilon = 1/k$ and $\rho = k$, $k = 1, 2, ...$, to conclude the existence of two sequences, $\{\sigma_k\} \subset \Theta$ and $\{z_k\} \subset \Re^n$, such that $d(\sigma_k, \sigma) < 1/k$ (and, consequently, $\lim_{k\to\infty}\sigma_k = \sigma$), $\|z_k\| > k$, and $z_k \in F_k \setminus F$. Hence, the sequence $\{z_k\}$ has no accumulation point, and this precludes the fulfillment of the Dolecki condition.

Now we proceed by proving the converse statement. Let W be an open set containing F. Since $\mathcal{F}$ is closed, the *cut-set valued map* $\mathcal{F}_\rho(.) := \mathcal{F}(.) \cap \rho\text{cl}(B)$ is USC at σ (according to [1, Corollary 1.4.10]), so that there exists a $\varepsilon_1 > 0$, $\varepsilon_1 \leq \varepsilon$ such that $F_1 \cap \rho\text{cl}(B) \subset W$ for all σ_1 satisfying $d(\sigma_1, \sigma) < \varepsilon_1$ (we can suppose, without loss of generality, that all the involved epsilons are smaller than 1). Finally, we write, for this ε_1,

$$F_1 = \{F_1 \cap \rho\text{cl}(B)\} \cup \{F_1 \setminus \rho\text{cl}(B)\} \subset W,$$

which completes the proof. □

Our characterization of upper semicontinuity, given in Theorem 3.1, is valid for every closed mapping $\mathcal{F} : \Lambda \rightsquigarrow \Re^n$ with Λ metrizable, and it is weaker than Dolecki's condition in the sense that it fails to be sufficient for upper semicontinuity when the range space is infinite dimensional. Actually, if $\Lambda = [0, 1]$ and X is the space of finitely nonzero sequences (i.e., $X = \{x = (\xi_1, \xi_2, ..., \xi_i, ...) \mid \xi_i \in \Re,\ i = 1, 2, ...,$ and only a finite number of ξ_i are nonzero$\}$), with the supremum norm (i.e., $\|x\| = \max|\xi_i|$), the mapping $\mathcal{F} : \Lambda \rightsquigarrow X$ such that $\mathcal{F}(\lambda) := \{x \in X \mid \|x\| = \lambda\}$ satisfies trivially our condition in Theorem 3.1 but fails to be USC at $\lambda = 1$ (if we take $\lambda_k = (k-1)/k$, $k = 1, 2, ...$, it is evident that the element $u_k \in X$, which has, as the unique nonzero component, $\xi_k = (k-1)/k$, satisfies $u_k \in \mathcal{F}(\lambda_k) \setminus \mathcal{F}(1)$, but the sequence $\{u_k\}$ has no accumulation point and, so, Dolecki's condition fails).

Corollary 3.2. *If the solution set of the consistent system σ is bounded, then $\mathcal{F}$ is USC at σ.*

Proof. First we shall prove that $\mathcal{F}$ is uniformly bounded in some neighborhood of σ; i.e., there exist positive scalars, ε and ρ, such that $F_1 \subset \rho\operatorname{cl}(B)$, provided that $d(\sigma_1, \sigma) < \varepsilon$. If this property does not hold, taking again $\varepsilon = 1/k$ and $\rho = k$, $k = 1, 2, ...$, we build the corresponding couple of sequences, $\{\sigma_k\}$ and $\{z_k\}$, as they were created in the proof of Theorem 3.1. It can be assumed, without loss of generality, the existence of $z = \lim_{k\to\infty} \|z_k\|^{-1} z_k$, and it is easily checked that $a_t' z \geq 0$ for all $t \in T$, so that $z \in F_\infty \setminus \{0_n\}$ and F will be unbounded.

We have concluded that $d(\sigma_1, \sigma) < \varepsilon$ entails $F_1 \subset \rho\operatorname{cl}(B)$. Hence, $F_1 \setminus \rho\operatorname{cl}(B) = F \setminus \rho\operatorname{cl}(B) = \emptyset$, and Theorem 3.1 can be applied. □

The last result can also be derived from the classical Berge theory (e.g., via the supremum function).

Example 3.3. If σ is a consistent system in $\Re$ $(n = 1)$, then $\mathcal{F}$ is always USC at σ.

The solution set F will be a bounded interval, an unbounded interval, or the whole space $\Re$. The third case is trivial and in the first case we apply Corollary 3.2. Then we analyze the only remaining case, for instance $F = [\alpha, \infty)$. Now the cut-set valued map $\mathcal{F}_{|\alpha|+1}(.) := \mathcal{F}(.) \cap [-|\alpha| - 1, |\alpha| + 1]$ is USC at σ, and for any open neighborhood $(\alpha - \delta, \infty)$ of F, the set $\mathcal{F}_{|\alpha|+1}(\sigma_1)$ is included in it, provided that σ_1 is close enough to σ. Convexity of F_1 implies that this set must be also contained in $(\alpha - \delta, \infty)$. Since, for every open set W containing F, there must exist $\delta > 0$ for which $W \supset (\alpha - \delta, \infty) \supset F$, the statement follows.

The following partial characterization of the upper semicontinuity property was established in [5, Theorem 3.1] in a more particular setting.

Theorem 3.4. *Let $\sigma = \{a_t' x \geq b_t, t \in T\}$ be a consistent system in $\Re^n$, with $n \geq 2$ and such that A is bounded and different from $\{0_n\}$. Then $\mathcal{F}$ is USC at σ if and only if F is bounded.*

Proof. Let $\mu > 0$ be such that $\|a_t\| < \mu$ for all $t \in T$. In order to prove the direct statement, let us consider two arbitrary positive numbers ε and ρ. If F were unbounded, a suitable perturbation of a point $z \in \operatorname{bd}(F) \setminus \rho\operatorname{cl}(B)$ would provide a point $y \notin F$ such that $\|y\| > \rho$ and $\|y - z\| < (n\mu)^{-1}\varepsilon$ (this requires $n \geq 2$). Then y would be a feasible solution of $\sigma_1 = \{a_t' x \geq b_t + a_t'(y - z), t \in T\}$ with $d(\sigma_1, \sigma) < \varepsilon$, so that $F_1 \setminus \rho\operatorname{cl}(B)$ would fail to be included in $F \setminus \rho\operatorname{cl}(B)$. The nontrivial part of the proof is then a consequence of Theorem 3.1. □

According to this result, $\mathcal{F}$ is not USC at the system σ_1 considered in Example 2.5.

When A is unbounded, upper semicontinuity of $\mathcal{F}$ does not imply boundedness of the solution set, relying heavily on the representation.

Example 3.5. We take two different representations of $F := \{x \in \Re^2 \mid -1 \leq x_1 \leq 1\}$ with the same index set $T = (-\infty, -1] \cup [1, \infty)$:

$$\begin{aligned} &\sigma_1 = \{t x_1 + 0 x_2 \geq -t^2, t \in T\} \\ \text{and} \qquad &\sigma_2 = \{t x_1 + 0 x_2 \geq -\mid t \mid, t \in T\}. \end{aligned}$$

It can be easily established that $\mathcal{F}$ is USC only at σ_2 (see [6, Example 2.1] for additional details).

The major difficulty in our analysis is the unboundedness of A. One can think about replacing σ by the following equivalent inequality system:

$$\sigma_s := \left\{ \left(\frac{a_t}{\max\{1, \|a_t\|\}} \right)' x \geq \frac{b_t}{\max\{1, \|a_t\|\}},\ t \in T \right\}.$$

If we use the uniform convergence pseudometric, d, restricted to the space of these *scaled* parameters, we are confining ourselves to the scenario delimited by Theorem 3.4, but this scaling procedure would permit arbitrarily large perturbations (in those constraints that originally have $\|a_t\|$ large), and this phenomenon changes drastically the original sense of the perturbation analysis.

Example 3.6. Given the consistent system $\sigma = \{a_t'x \geq b_t, t \in T\}$, with T infinite, the system $\sigma_1 = \{ka_t'x \geq kb_t, (t,k) \in T \times \mathcal{N}\}$, where $\mathcal{N}$ represents the natural numbers set, is equivalent to σ (same solution set), its index set has the same cardinality, and $\mathcal{F}$ is USC at σ_1.

In order to prove the last assertion, we perturb σ_1 in the usual way to get

$$\sigma_2 = \{(ka_t + u_{(t,k)})'x \geq kb_t + v_{(t,k)}, (t,k) \in T \times \mathcal{N}\}.$$

If $\left\|u_{(t,k)}\right\| < \varepsilon$ and $\left|v_{(t,k)}\right| < \varepsilon$, for every $(t,k) \in T \times \mathcal{N}$, we have $d(\sigma_2, \sigma_1) \leq \varepsilon$. If x_0 is a solution of σ_2, multiplying by k^{-1} each inequality associated with a particular $k \in \mathcal{N}$, we obtain

$$(a_t + k^{-1}u_{(t,k)})'x_0 \geq b_t + k^{-1}v_{(t,k)}.$$

For each fixed $t \in T$, we take limits in both sides for $k \longrightarrow \infty$ giving rise to $a_t'x_0 \geq b_t$; i.e., x_0 is also a solution of σ_1 and, obviously, $\mathcal{F}$ is USC at σ_1. □

The only drawback of the conditions given in Theorems 3.1 and 3.4 is that they can hardly be checked in practice, as they are not explicitly related to the coefficients of σ. Thus, the next sections will be devoted to deriving conditions involving these coefficients and their associated elements (the cones M and P and the asymptotic system σ^a).

4. Necessary conditions for upper semicontinuity. The following result exploits the subtle relationship between M and P, which was described in Lemma 2.2.

THEOREM 4.1. *Let $\sigma = \{a_t'x \geq b_t, t \in T\}$ be a consistent system such that sufficiently small perturbations preserve consistency. If $\mathcal{F}$ is USC at σ, then there cannot exist $y \in \operatorname{bd}(M) \setminus P$ such that $\{\lambda y \mid \lambda \geq 0\}$ is an exposed ray of* $\operatorname{cl}(M)$.

Proof. Let us assume the contrary; i.e., there exists $y \in \operatorname{bd}(M) \setminus P$ such that $D := \{\lambda y \mid \lambda \geq 0\}$ is an exposed ray of $\operatorname{cl}(M)$.

We start by considering a nontrivial supporting hyperplane $H := \{z \mid c'z = 0\}$, $c \neq 0_n$, such that $H \cap \operatorname{cl}(M) = D$. We shall assume that $c'z \geq 0$ for each $z \in \operatorname{cl}(M)$ and, therefore, $c'z > 0$ for every $z \in P \setminus \{0_n\}$ (recall Lemma 2.2(i)).

Let us take any $u \in M \setminus D$, $\|u\| = 1$. It must hold $c'u > 0$ and, for an arbitrary $\varepsilon > 0$, we define the system $\sigma_1 = \{(a_t + \varepsilon u)'x \geq b_t, t \in T\}$. It is obvious that $d(\sigma_1, \sigma) = \varepsilon$, $M_1 \subset M$, and, then, $\operatorname{cl}(M_1) \subset \operatorname{cl}(M)$. Moreover, we can take ε small enough to have σ_1 consistent.

Now we consider a vector e such that $e'y < 0$. For all $z \in D \setminus \{0_n\}$ and $\mu > 0$ we have $(c + \mu e)'z = c'z + \mu e'z = \mu e'z < 0$. Our immediate aim is to prove the existence of $\mu_0 > 0$ such that $(c+\mu_0 e)'(a_t+\varepsilon u) \geq 0$, whichever $t \in T$ we take. This would imply that the exposed D and $\mathrm{cl}(M_1)$ can be properly separated, concluding that $\mathrm{cl}(M_1)$ is, in fact, strictly contained in $\mathrm{cl}(M)$.

If such a μ_0 does not exist, for each $k \in \mathcal{N}$ we will find $t_k \in T$ such that

$$(c + k^{-1}e)'(a_{t_k} + \varepsilon u) < 0.$$

Two possibilities have to be considered.

(i) If the sequence $\{a_{t_k}\}$ is bounded, there must exist a convergent subsequence. Without loss of generality, we write $\lim_{k\to\infty} a_{t_k} = a \in \mathrm{cl}(M)$ and, taking limits in the last inequality, we get $c'(a + \varepsilon u) \leq 0$, but this is impossible because $c'a \geq 0$ and $c'u > 0$.

(ii) If $\{a_{t_k}\}$ is unbounded, a subsequence $\{a_{t_{k_r}}\}$ exists such that $\left\|a_{t_{k_r}}\right\| \geq r$, $r = 1, 2, \ldots$. Now the sequence $\{\left\|a_{t_{k_r}}\right\|^{-1} a_{t_{k_r}}\}$ will contain a convergent subsequence and, for the sake of brevity, we write $\lim_{r\to\infty} \left\|a_{t_{k_r}}\right\|^{-1} a_{t_{k_r}} = b$, with $b \in A_\infty \setminus \{0_n\} \subset P \setminus \{0_n\}$. Taking limits again over the inequalities corresponding to the indices t_{k_r}, we obtain

$$\lim_{r\to\infty}(c + k_r^{-1}e)'\{\left\|a_{t_{k_r}}\right\|^{-1}(a_{t_{k_r}} + \varepsilon u)\} = c'b \leq 0,$$

but one must have $c'b > 0$ since $b \in P \setminus \{0_n\}$.

We have concluded that, for an arbitrarily small ε, a consistent system σ_1 can be built such that $d(\sigma_1, \sigma) = \varepsilon$ and with $\mathrm{cl}(M_1)$ strictly contained in $\mathrm{cl}(M)$. Therefore, F_∞ is strictly contained in $(F_1)_\infty$. Now if we take, for any $\alpha > 0$, the open set $W = F + \alpha B$, we have $F \subset W$ and $W_\infty = F_\infty$ (Lemma 2.1(viii)), leading to $(F_1)_\infty \backslash W_\infty \neq \emptyset$. This prevents the inclusion $F_1 \subset W$ since taking any $x \in F_1$ and $y \in (F_1)_\infty \setminus W_\infty$, the half-line, in F_1, $\{x + \lambda y \mid \lambda \geq 0\}$ will leave W, for λ large enough. Otherwise, $y = \lim_{k\to\infty} k^{-1}(x + ky)$ will belong to W_∞, giving rise to a contradiction. □

If F is bounded, $\mathrm{cl}(M) = (F_\infty)^o = \{0_n\}^o = \Re^n$ and $\mathrm{bd}(M) = \emptyset$. Hence, the necessary condition for upper semicontinuity given in Theorem 4.1 holds trivially in this case as well as in the case $n = 1$.

It has been established (see [7, Theorem 3.1]) that $\mathcal{F}$ is LSC at σ if and only if σ lies in the topological interior of the consistent systems set in Θ. Therefore, Theorem 4.1 provides a necessary condition for the continuity of $\mathcal{F}$ at σ.

COROLLARY 4.2. *Let $\sigma = \{a_t'x \geq b_t, t \in T\}$ be a consistent system in $\Re^n$, with $n \geq 2$, such that sufficiently small perturbations provide consistent systems and for which* $\mathrm{cl}(M)$ *is pointed. If $\mathcal{F}$ is USC at σ, then $P = \mathrm{cl}(M)$.*

Proof. Let S be any set of vectors in $\mathrm{cl}(M)$ such that each exposed ray of $\mathrm{cl}(M)$ is generated by some $a \in S$. Theorem 4.1 enables us to write $S \subset P$, whereas Corollary 18.7.1 in [11] yields $\mathrm{cl}(M) = \mathrm{cl\ cone}(S) \subset P$, because P is closed (Lemma 2.2 (iii)). Hence, $\mathrm{cl}(M) = P$ (Lemma 2.2(i)). □

Unfortunately, the example below shows that the necessary conditions given here are not sufficient.

Example 4.3. Let us consider the system, in $\Re^2$, $\sigma = \{-2tx_1 + x_2 \geq -t^2,\ t \in \Re\}$, which provides a linear representation of $F = \{x \in \Re^2 \mid x_2 \geq (x_1)^2\}$.

We observe that $M = \{a \in \Re^2 \mid a_2 > 0\} \cup \{0_2\}$ and $P = \{a \in \Re^2 \mid a_2 = 0\}$. Thus, $\mathrm{bd}(M) \setminus P = \emptyset$ and the conditions in Theorem 4.1 and Corollary 4.2 are trivially satisfied. Nevertheless, $\mathcal{F}$ is not USC at σ. In fact, for any $\varepsilon > 0$, we consider the system $\sigma_1 = \{-2tx_1 + x_2 \geq -t^2 - \varepsilon,\ t \in \Re\}$, whose feasible set is $F_1 = F - (0, \varepsilon)'$. Obviously, $d(\sigma_1, \sigma) = \varepsilon$, but there is no positive ρ such that $F_1 \backslash \rho\mathrm{cl}(B) \subset F \setminus \rho\mathrm{cl}(B)$. □

Next we give another necessary condition involving the asymptotic system σ^a.

THEOREM 4.4. *Let $\sigma = \{a_t'x \geq b_t, t \in T\}$ be a system in $\Re^n$, with $n \geq 2$, such that the solution set F and $A = \{a_t, t \in T\}$ are both unbounded. If $\mathcal{F}$ is USC at σ, then there will exist $\rho > 0$ such that, for every $z \in \operatorname{bd}(F) \setminus \rho\operatorname{cl}(B)$, one can find an implicit fixed constraint which is active at z.*

Proof. If $\mathcal{F}$ is USC at σ, we know, from Theorem 3.1, the existence of a couple of positive scalars, ε and ρ, such that

$$F_1 \setminus \rho\operatorname{cl}(B) \subset F \setminus \rho\operatorname{cl}(B)$$

for every $\sigma_1 \in \Theta$ such that $d(\sigma_1, \sigma) < \varepsilon$. If we take $z \in \operatorname{bd}(F) \setminus \rho\operatorname{cl}(B)$, a sequence $\{z_k\}$ exists such that $z_k \notin F$, $\|z_k\| > \rho$, and $\lim_{k\to\infty} z_k = z$. We define a perturbed system associated with each $k \in \mathcal{N}$:

$$\sigma_k := \left\{ \begin{array}{cc} a_t'x \geq b_t + a_t'(z_k - z), & t \in T_k := \{t \in T \mid a_t'z_k < b_t\} \\ a_t'x \geq b_t, & t \in T \setminus T_k \end{array} \right\}.$$

It is evident that $z_k \in F_k$ and this implies $d(\sigma_k, \sigma) \geq \varepsilon$. Thus, there will exist $t_k \in T_k$ such that $\left|a_{t_k}'(z_k - z)\right| \geq \varepsilon/2$, which leads to $\lim_{k\to\infty} \|a_{t_k}\| = \infty$, and we can suppose, without loss of generality, that $\lim_{k\to\infty} \|a_{t_k}\|^{-1} a_{t_k} = a \in A_\infty$. On the other hand, $a_{t_k}'z_k < b_{t_k}$ and $a_{t_k}'z \geq b_{t_k}$ yield together the existence of $y_k \in [z, z_k[$ satisfying $a_{t_k}'y_k = b_{t_k}$. Since $\lim_{k\to\infty} y_k = z$, we get

$$a'z = \lim_{k\to\infty} \|a_{t_k}\|^{-1} a_{t_k}' y_k = \lim_{k\to\infty} \|a_{t_k}\|^{-1} b_{t_k},$$

and $a'x \geq a'z$ belongs, obviously, to σ^a. □

According to this result, $\mathcal{F}$ cannot be USC at the system σ_2 introduced in Example 2.5.

The condition given in Theorem 4.4 is not sufficient for the upper semicontinuity property, as the following example shows.

Example 4.5. Let us consider the following system, in $\Re^2$,

$$\sigma := \left\{ \begin{array}{c} x_2 \geq 0, \\ -kx_2 \geq 0, \ k = 1, 2, \ldots \end{array} \right\}.$$

Straightforward calculations yield the following.

(i) $F = \{x \in \Re^2 \mid x_2 = 0\}$, $\sigma^a = \{-x_2 \geq 0\}$, and the condition in Theorem 4.4 is fulfilled.

(ii) The system σ_ε obtained from σ replacing the first inequality by $x_2 \geq -\varepsilon$ satisfies $d(\sigma_\varepsilon, \varepsilon) = \varepsilon$, but the condition in Theorem 3.1 is never accomplished.

We shall finish this section dealing with a particular case where the USC property is characterized through Theorem 4.4. If $\dim(M) = 1$, we take $a \in \Re^n$, $\|a\| = 1$, such that $a_t = \alpha_t a$, $\alpha_t \in \Re$, for every $t \in T$. Then one can define

$$\varphi_1 := \left\{ \begin{array}{c} \sup\{b_t/\alpha_t : \ t \in T \text{ and } \alpha_t > 0\}, \\ -\infty \text{ if } \alpha_t \leq 0 \text{ for all t} \in T, \end{array} \right.$$

$$\varphi_2 := \left\{ \begin{array}{c} \inf\{b_t/\alpha_t : \ t \in T \text{ and } \alpha_t < 0\}, \\ \infty \text{ if } \alpha_t \geq 0 \text{ for all t} \in T. \end{array} \right.$$

If both values are finite, the solution set will be the *slice* $F = \{x \in \Re^n \mid \varphi_1 \leq a'x \leq \varphi_2\}$. If one of these values is infinite (for instance, $\varphi_2 = \infty$), F will be a half-space

($F = \{x \in \Re^n \mid \varphi_1 \leq a'x\}$, accordingly). The possibility $F = \Re^n$ is excluded here because it would imply $\alpha_t = 0$ for all $t \in T$, and then $\dim(M) = 0$.

COROLLARY 4.6. *Let $\sigma = \{a_t'x \geq b_t, t \in T\}$ be a consistent system for which* $\dim(M) = 1$ *and $n \geq 2$. Then $\mathcal{F}$ is USC at σ if and only if the following condition holds:*

♢ *If φ_i is finite for $i = 1$ or $i = 2$, then there must exist a sequence $\{\alpha_{t_k^i}\} \subset (-1)^{i+1}\Re_+$ such that $\lim_{k\to\infty}|\alpha_{t_k^i}| = \infty$ and $\lim_{k\to\infty}(b_{t_k^i}/\alpha_{t_k^i}) = \varphi_i$.*

Proof. We focus on the most involved case, in which both values φ_i are finite. If the condition ♢ holds, we can write

$$\begin{pmatrix} a \\ \varphi_1 \end{pmatrix} = \lim_{k\to\infty}(\alpha_{t_k^1})^{-1}\begin{pmatrix} \alpha_{t_k^1} a \\ b_{t_k^1} \end{pmatrix} = \lim_{k\to\infty}\left\|a_{t_k^1}\right\|^{-1}\begin{pmatrix} a_{t_k^1} \\ b_{t_k^1} \end{pmatrix},$$

and

$$\begin{pmatrix} -a \\ -\varphi_2 \end{pmatrix} = \lim_{k\to\infty}\left|\alpha_{t_k^2}\right|^{-1}\begin{pmatrix} \alpha_{t_k^2} a \\ b_{t_k^2} \end{pmatrix} = \lim_{k\to\infty}\left\|a_{t_k^2}\right\|^{-1}\begin{pmatrix} a_{t_k^2} \\ b_{t_k^2} \end{pmatrix}.$$

This means that the inequalities $a'x \geq \varphi_1$ and $a'x \leq \varphi_2$ belong to σ^a, and Lemma 2.4 applies, yielding $F_1 \subset (F_1)^a = F^a = F$, for any system σ_1 such that $d(\sigma_1, \sigma)$ is finite. Now the upper semicontinuity of $\mathcal{F}$ at σ becomes evident.

In order to prove the converse, it is enough to apply Theorem 4.4. Again, we assume that φ_1 and φ_2 are finite, and let us consider first $\dim(F) = n$; i.e., $\varphi_1 \neq \varphi_2$. Now, if $z \in \operatorname{bd}(F)\backslash\rho\operatorname{cl}(B)$ it must be, for instance, $a'z = \varphi_1$, and Theorem 4.4 ensures that the constraint $a'x \geq a'z$ belongs to σ^a (it defines the only possible supporting half-space to F at z, and $\|a\| = 1$). This fact requires the existence of a sequence $\{t_k^1\} \subset T$ such that $\lim_{k\to\infty}\|a_{t_k^1}\| = \infty$ and

$$\begin{aligned}\begin{pmatrix} a \\ a'z \end{pmatrix} = \begin{pmatrix} a \\ \varphi_1 \end{pmatrix} &= \lim_{k\to\infty}\left\|a_{t_k^1}\right\|^{-1}\begin{pmatrix} a_{t_k^1} \\ b_{t_k^1} \end{pmatrix} \\ &= \lim_{k\to\infty}\left|\alpha_{t_k^1}\right|^{-1}\begin{pmatrix} \alpha_{t_k^1} a \\ b_{t_k^1} \end{pmatrix},\end{aligned}$$

and this expression entails the corresponding part in the condition ♢. The discussion of the possibility $a'z = \varphi_2$ follows a similar reasoning, yielding the other part.

If $\varphi_1 = \varphi_2$ (i.e., $\dim(F) = n - 1$), taking two sequences $\{z_k := z + k^{-1}a\}$ and $\{y_k := y - k^{-1}a\}$, where $z \in F \setminus \rho\operatorname{cl}(B)$, and following the same steps as those in the proof of Theorem 4.4, we obtain a couple of constraints in σ^a, namely $a_1'x \geq a_1'z$ and $a_2'x \geq a_2'z$, such that $a_1'a \leq 0$ and $a_2'a \geq 0$. Since $\{a_1, a_2\} \subset \{-a, +a\}$, if we had $a_1 = a_2$ we would get $a_1'a = a_2'a = 0$, and this is obviously impossible. □

Observe that condition ♢, in Corollary 4.6, never holds for a finite system ($|T| < \infty$); i.e., upper semicontinuity is excluded in this case (this conclusion also follows from Theorem 3.4).

Let us illustrate the scope of the last result by revisiting Example 3.5. Concerning σ_1, observe that $\lim_{t\to\infty}(-t^2)/t = -\infty \neq -1 = \varphi_1$, so that $\mathcal{F}$ is not USC at σ_1. However, $\{k\}$ and $\{-k\}$ are sequences satisfying condition ♢ for σ_2, and $\mathcal{F}$ is USC at σ_2.

5. Sufficient conditions for upper semicontinuity.

The first result in this section refers to the asymptotic system σ^a and its solution set F^a.

THEOREM 5.1. *If σ is a consistent system for which $F^a \setminus F$ is bounded, then $\mathcal{F}$ is USC at σ.*

Proof. Since $F^a \setminus F$ is bounded, there will exist $\rho > 0$ such that $F^a \setminus F \subset \rho\mathrm{cl}(B)$, and $F \setminus \rho\mathrm{cl}(B) = F^a \setminus \rho\mathrm{cl}(B)$. If $d(\sigma_1, \sigma)$ is finite, Lemma 2.4 allows us to write

$$F_1 \setminus \rho\mathrm{cl}(B) \subset F^a \setminus \rho\mathrm{cl}(B) = F \setminus \rho\mathrm{cl}(B),$$

and Theorem 3.1 applies (with $\varepsilon = \infty$). □

Theorem 5.1 provides the upper semicontinuity of $\mathcal{F}$ at the system σ_3 introduced in Example 2.5.

The following example shows that the condition given in Theorem 5.1 fails to be necessary, even in the favorable case in which F is full dimensional.

Example 5.2. Let us consider the system, in $\Re^2$,

$$\sigma := \{tx_1 + x_2 \geq -|t|\,,\ t \in \Re\}.$$

We shall prove that $\dim(F) = 2$ and that $\mathcal{F}$ is USC at σ, despite the unboundedness of $F^a \setminus F$.

It can be easily checked that

$$F = \{x \in \Re^2 \mid x_2 \geq 0 \text{ and } x_1 \in [-1, +1]\},$$

whereas

$$\sigma^a = \{-x_1 \geq -1,\ x_1 \geq -1\},$$

and $F^a \setminus F$ is unbounded.

Next we shall prove that $\mathcal{F}$ is USC at σ, starting from a perturbed system σ_1 such that $d(\sigma_1, \sigma) < \varepsilon < 1$

$$\sigma_1 := \{(t + u_t)x_1 + (1 + v_t)x_2 \geq -|t| + w_t,\ t \in \Re\},$$

where $|u_t| < \varepsilon,\ |v_t| < \varepsilon,\ |w_t| < \varepsilon$ for all $t \in \Re$. According to Lemma 2.4, $F_1 \subset F^a$ and if $x = (x_1, x_2)' \in F_1$ one has $x_1 \in [-1, +1]$. We proceed by showing that x_2 is also bounded from below in F_1. For $t = 1$ we get

$$(1 + v_1)x_2 \geq -1 + w_1 - (1 + u_1)x_1 \geq -1 + w_1 - 1 - u_1,$$

and, since $1 + v_1 > 0$, we write

$$x_2 \geq \frac{-2 + w_1 - u_1}{1 + v_1} > \frac{-2 - 2\varepsilon}{1 + v_1} > \frac{-2(1 + \varepsilon)}{1 - \varepsilon}.$$

Then, if we define $\rho := 2(1 + \varepsilon)/(1 - \varepsilon)$, it is obvious that $F_1 \setminus \rho\mathrm{cl}(B) \subset F \setminus \rho\mathrm{cl}(B)$, provided that $d(\sigma_1, \sigma) < \varepsilon$.

The system studied in the following example shows that the upper semicontinuity of $\mathcal{F}$ at σ does not guarantee $\dim(F) = \dim(F^a)$. However, if $\dim(F) = n$, the above dimensional equality is trivial, and if $\dim(F) = n - 1$ the equality is still valid under the upper semicontinuity property (it can be argued as in the final paragraph of the proof given for Corollary 4.6, approaching any point $z \in \mathrm{rint}(F) \setminus \rho\mathrm{cl}(B)$ by means of the same couple of sequences $\{z_k := z + k^{-1}a\}$ and $\{y_k := y - k^{-1}a\}$, where $\mathrm{aff}(F) = \{x \in \Re^n \mid a'x = b\}$).

Example 5.3. We analyze the system in $\Re^2$

$$\sigma = \left\{ \begin{array}{l} tx_1 + x_2 \geq 0,\ t \in \Re \\ sx_1 - x_2 \geq 0,\ s \in \Re \end{array} \right\}.$$

It is obvious that $F = \{0_2\}$ and, so, $\mathcal{F}$ is USC at σ (Corollary 3.2), whereas $\sigma^a = \{x_1 \geq 0,\ -x_1 \geq 0\}$, giving rise to $F^a = \{x \in \Re^2 \mid x_1 = 0\}$. Thus, $\dim(F) < \dim(F^a)$.

Before we reach the following sufficient condition, we need a technical lemma.

LEMMA 5.4. *Let $c \in \Re^n$, $h \in \Re^n \setminus \{0_n\}$ and $\rho > 0$ such that $\|c + \lambda h\| > \rho$ for all $\lambda \geq 0$. Then, for every $d \in \Re^n$, there exists $\lambda_d > 0$ such that, for each $\lambda \geq \lambda_d$ and all $\alpha \in [0, 1]$, the following inequality holds:*

$$\|\alpha(d + \lambda h) + (1 - \alpha)c\| > \rho.$$

Proof. First we shall prove that there exists $\varepsilon > 0$ such that, for each $u \in \varepsilon B$ and for all $\lambda \geq 0$, we have $\|c + \lambda(h + u)\| > \rho$. If this is not the case, there will exist sequences $\{u_k\}$ and $\{\lambda_k\}$ such that $\lim_{k\to\infty} u_k = 0_n,\ \lambda_k > 0$, and $\|c + \lambda_k(h + u_k)\| \leq \rho,\ k = 1, 2, \ldots$.

Two possibilities arise. If $\{\lambda_k\}$ is bounded, there must exist $\lambda_0 \geq 0$ for which $\|c + \lambda_0 h\| \leq \rho$ is fulfilled, and this represents a contradiction. If, alternatively, $\{\lambda_k\}$ is unbounded, we get $\left\|\lambda_k^{-1} c + h + u_k\right\| \leq \rho/\lambda_k,\ k = 1, 2, \ldots$, giving rise to $h = 0_n$, which constitutes another contradiction.

Now we are ready to finish the proof:

$$\|\alpha(d + \lambda h) + (1 - \alpha)c\| = \left\|c + \alpha\lambda[h + \lambda^{-1}(d - c)]\right\| > \rho,$$

if λ is large enough to guarantee that $\lambda^{-1}(d - c) < \varepsilon$. □

THEOREM 5.5. *Let $\sigma = \{a_t' x \geq b_t, t \in T\}$ be a system, in $\Re^n$, such that the solution set F and $A = \{a_t, t \in T\}$ are both unbounded. Suppose that, additionally, σ satisfies the following conditions:*

(a) $\dim(F) = \dim(F^a)$.

(b) *There exists $\rho > 0$ such that $F \cap\ \rho\mathrm{cl}(B) \neq \emptyset$, and*

(b1) *for all $z \in F^a \setminus \rho\mathrm{cl}(B)$ there exists $h \in F_\infty \setminus \{0_n\}$ such that $\|z + \lambda h\| > \rho$ for every $\lambda \geq 0$;*

(b2) *for all $z \in \mathrm{rbd}(F) \setminus \rho\mathrm{cl}(B)$ there exists an inequality in σ^a, $a'x \geq a'z$, which is properly active at z (i.e., F is not completely contained in the hyperplane $\{x \in \Re^n \mid a'x = a'z\}$).*

Then $\mathcal{F}$ is USC at σ.

Proof. The first step in the proof establishes

$$\mathrm{rbd}(F) \setminus \rho\mathrm{cl}(B) \subset \mathrm{rbd}(F^a) \setminus \rho\mathrm{cl}(B).$$

This a straightforward consequence of (b2) and [11, Theorem 11.6].

The second step consists of proving

$$F \setminus \rho\mathrm{cl}(B) \subset F^a \setminus \rho\mathrm{cl}(B),$$

and then applying Theorem 5.1. Otherwise, we take $z \in F^a \setminus F,\ \|z\| > \rho$, and by condition (b1) we can consider $h \in F_\infty \setminus \{0_n\}$ such that $\|z + \lambda h\| > \rho$ for every $\lambda \geq 0$. If we pick $y \in \mathrm{rint}(F)$, Lemma 5.4 yields $\|\alpha(y + \lambda h) + (1 - \alpha)z\| > \rho$ if $\alpha \in [0, 1]$ and λ is large enough, for instance $\lambda \geq \lambda_0$.

Corollary 8.3.1 in [11] establishes $(\mathrm{rint}(F))_\infty = (F)_\infty$ and, for all $\lambda \geq 0$, $y + \lambda h \in \mathrm{rint}(F) \subset \mathrm{rint}(F^a)$ (we have used condition (a)). Since it has been assumed that $z \notin F$, if $\lambda \geq \lambda_0$ we can find in the segment $]y + \lambda h, z[$ a point v such that $v \in \mathrm{rbd}(F) \setminus \rho\mathrm{cl}(B)$ at the same time that the accessibility lemma [11, Theorem 6.1] yields $v \in \mathrm{rint}(F^a) \setminus \rho\mathrm{cl}(B)$. This last statement contradicts the inclusion relation, already established, between the relative boundaries. □

In Example 5.2, conditions (a) and (b2), for $\rho \geq 1$, are satisfied, but (b1) fails. This fact allows us to point out that the conditions in Theorem 5.5 are not conjointly necessary for upper semicontinuity. The corollary that follows provides a vast class of systems for which condition (b1) is fulfilled.

COROLLARY 5.6. *Let* $\sigma = \{a_t' x \geq b_t, t \in T\}$ *be a consistent system, in* $\Re^n$, *which satisfies the following conditions:*

(i) $\mathrm{cl}(M)$ *is pointed.*

(ii) $\mathrm{bd}(M) \setminus P = \emptyset$.

(iii) $\{b_t,\ t \in T\}$ *is bounded.*

Then conditions (a) *and* (b1) *in Theorem* 5.5 *are fulfilled.*

Proof. Corollary 14.6.1 in [11] leads to

$$\dim(F_\infty) = \dim\{\mathrm{cl}(M)\}^o = n - \mathrm{lineality}\{\mathrm{cl}(M)\},$$

where lineality$\{\mathrm{cl}(M)\}$ denotes the dimension of the largest subspace contained in $\mathrm{cl}(M)$. Since $\mathrm{cl}(M)$ is pointed, we get $\dim(F_\infty) = n$, and both F and F^a are full dimensional. Hence, (a) is obviously satisfied.

By Lemma 2.2(iii), P is closed. Moreover, in the proof of Corollary 4.2 we have established $\mathrm{cl}(M) = \mathrm{cl\ cone}(S)$, where $S \subset \mathrm{bd}(M)$. Condition (ii) implies $\mathrm{bd}(M) \subset P$ and, therefore,

$$\mathrm{cl}(M) = \mathrm{cl\ cone}(S) \subset \mathrm{cl\ conv\ bd}(M) \subset P.$$

This inclusion actually means $\mathrm{cl}(M) = P$ and $F_\infty = \{\mathrm{cl}(M)\}^o = P^o = F^a$ as a consequence of (iii). Now, if $z \in F^a \setminus \rho\mathrm{cl}(B)$, we can take $h = z$ and so $\|z + \lambda h\| = (1 + \lambda)\|z\| > \rho$ for every $\lambda \geq 0$. Therefore, condition (b1) holds. □

Acknowledgment. The authors are indebted to the referees for their valuable comments and suggestions.

REFERENCES

[1] J. -P. AUBIN AND H. FRANKOWSKA, *Set-Valued Analysis,* Birkhäuser Boston, Cambridge, MA, 1990.

[2] B. BANK, J. GUDDAT, D. KLATTE, B. KUMMER, AND K. TAMMER, *Non-linear Parametric Optimization,* Birkhäuser-Verlag, Basel, Switzerland, 1983.

[3] K. BORDER, *Fixed Point Theorems with Applications to Economics and Game Theory,* Cambridge University Press, Cambridge, 1985.

[4] B. BROSOWSKI, *Parametric Semi-Infinite Optimization,* Verlag Peter D. Lang, Frankfurt a. M. and Bern, 1982.

[5] T. FISCHER, *Contributions to semi-infinite linear optimization,* Methoden und Verfahren der mathematischen Physik, Band 27, Verlag Peter D. Lang, Berlin, 1987, pp. 175–199.

[6] M. A. GOBERNA AND M. A. LOPEZ, *A note on topological stability of linear semi-infinite inequality systems,* J. Optim. Theory Appl., 89 (1996), pp. 227–236.

[7] 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.

[8] H. J. GREENBERG AND W. P. PIERSKALLA, *Stability theorems for infinitely constrained mathematical programs,* J. Optim. Theory Appl., 16 (1975), pp. 409–428.

[9] S. Helbig, *Stability in disjunctive linear optimization* I*: Continuity of the feasible set*, Optimization, 21 (1990), pp. 855–869.

[10] S. M. Robinson, *Stability theory for systems of inequalities. Part* I*: Linear systems*, SIAM J. Numer. Anal., 12 (1975), pp. 754–769.

[11] R. T. Rockafellar, *Convex Analysis,* Princeton University Press, Princeton, NJ, 1970.

[12] H. Tuy, *Stability property of a system of inequalities,* Math. Oper. Statist. Ser. Opt., 8 (1977), pp. 27–39.

SIAM J. OPTIM.
Vol. 7, No. 4, pp. 1152–1159, November 1997

015

INTEGER ISOTONE OPTIMIZATION*

MING-HONG LIU† AND VASANT A. UBHAYA‡

Abstract. Consider the following integer isotone optimization problem. Given an n-vector x find an n-vector y with integer components so as to minimize $\max\{w_j|x_j - y_j| : 1 \le j \le n\}$ subject to $y_1 \le y_2 \le \cdots \le y_n$, where each weight $w_j > 0$. In this article, the dual of this problem is defined, a strong duality theorem is established, and the set of all optimal solutions is shown to be all monotonic integer vectors lying in a vector interval. In addition, algorithms are obtained for computation of optimal solutions having the worst-case time complexity $O(n^2)$, when w_j are arbitrary, and $O(n)$, when $w_j = 1$ for all j. The problem considered is of isotonic regression type and has practical applications, for example, to estimation and curve fitting. It is also of independent mathematical interest. The problem and the results can be easily extended to a partially ordered set.

Key words. isotonic regression, isotone optimization, duality, uniform norm, optimal solutions, min–max and max–min, algorithms, complexity

AMS subject classifications. 41A30, 90C10, 26A48

PII. S1052623494272302

1. Introduction.

Consider the following integer isotone optimization problem. Given $x = (x_1, x_2, \ldots, x_n) \in R^n$ find $y = (y_1, y_2, \ldots, y_n) \in R^n$ with integer y_j, $1 \le j \le n$, so as to minimize

$$d(x, y) := \max\{w_j|x_j - y_j| : 1 \le j \le n\} \tag{1.1}$$

subject to the isotonicity or monotonicity constraint $y_1 \le y_2 \le \cdots \le y_n$, where each weight $w_j > 0$, $1 \le j \le n$. Such problems without the integer constraint on y_j fall into the general class of problems called isotonic regression. For example, if $d_p(x, y) := \sum_{j=1}^{n} w_j|x_j - y_j|^p$, $1 \le p < \infty$, and we minimize $d_2(x, y)$ and $d_1(x, y)$, instead of $d(x, y)$ of (1.1), the problem is called the isotonic regression [3, 16] and isotonic median regression [5, 15], respectively. When $d(x, y)$ is minimized without the integer constraint on y_j, the problem is called isotone optimization [18, 19]. In the classical approach to regression and other optimization problems, the least squares objective, $d_2(x, y)$, has been extensively used. This method gained its popularity due to its applications to linear regression—the differentiability properties of the objective leading to explicit mathematical expressions. For nonlinear problems, one is generally forced to use either an iterative procedure justified by a mathematical algorithm or a highly inefficient exhaustive search [13]. With the advances in optimization and computational applications, for some time now, both $d_1(x, y)$ (mean absolute deviation) and $d(x, y)$ (maximum absolute deviation) objective functions are being used to obtain best fits and estimators. See, for example, MINMAD and MINMADAX regression in [2] and the least absolute value (LAV) or L_1-norm estimation and L_∞-norm estimation in [10]. See also [4]. Note that $\|x\|_p := d_p(x, 0)^{1/p}$, $1 \le p < \infty$, and $\|x\|_\infty := d(x, 0)$ are, respectively, the L_p and L_∞ norms on R^n. Both d_1 and d_∞ objectives have the strong advantage that their form allows transformation of the

*Received by the editors August 3, 1994; accepted for publication (in revised form) July 16, 1996. http://www.siam.org/journals/siopt/7-4/27230.html

†Information Advantage Inc., 7905 Golden Triangle Drive, Eden Prairie, MN 55344-7227.

‡Department of Computer Science and Operations Research, 258 IACC Building, North Dakota State University, Fargo, ND 58105 (ubhaya@plains.nodak.edu).

problem to a linear program facilitating computation of its solution [2]. Different objective functions, such as the ones considered above, yield different computational complexities and give best fits which have different properties and which are, therefore, appropriate in different situations. The nature of the problem essentially determines the choice of the objective function. The example that follows illustrates this point.

Goldstein and Kruskal [9] first introduced the integer constraint on y_j in isotonic regression and cited practical applications of this problem. They showed that the optimal solution of this problem can be obtained by rounding the unique optimal solution of the isotonic regression problem. Motivated by their work, we analyze the integer isotone optimization problem. The following example is taken from their article and modified for the purpose of illustrating an application of our problem. A magazine has different local versions published in distinct localities. Each advertiser purchases space in one or more versions, is assigned a "class number," and is charged according to that number. The class number is an integer which is approximately proportional to the combined circulation of the versions selected by the advertiser. As a result of several "special case" decisions, circulation shifts, historical developments, etc., these numbers have become inconsistent. It is desired to assign new class numbers to advertisers according to their combined circulation. The class numbers must be fair; i.e., for any two advertisers X and Y, X must not be assigned a lower class number than Y if X has a higher circulation than Y. Furthermore, the maximum absolute change in each class number must be kept as small as possible to minimize the "sense of disturbance" each advertiser experiences. Suppose that the advertisers are numbered $1, 2, \ldots, n$ according to the increasing order of their circulation. For the jth advertiser, $1 \leq j \leq n$, let x_j and y_j denote, respectively, the old class number and the new one to be computed. The sense of disturbance to be minimized is quantified by $\max\{|x_j - y_j| : 1 \leq j \leq n\}$. We certainly have $y_1 \leq y_2 \leq \cdots \leq y_n$ with integer y_j. This is problem (1.1) with $w_j = 1$. Note that we could not simultaneously minimize the absolute change in each class number; minimizing $d(x, y)$, which bounds each change, is the best we could do. Note also that d_2 or d_1 would not be as appropriate for this problem as d.

It will be seen from the results summarized below that the problem has a rather rich and interesting mathematical structure. We establish a dual problem and weak duality (section 2). We obtain a strong duality result and a closed form representation for the set of optimal solutions—a vector interval whose endpoints are the minimal and maximal solutions (section 3). We also obtain max–min and min–max forms of optimal solutions. We devise algorithms of worst-case time complexity $O(n^2)$ for computation of solutions. When $w_j = 1$ for all j, we show that a solution to our problem is obtained by rounding a special solution of the problem without the integer constraint. This gives an $O(n)$ algorithm for computing the solution (section 4). The problem of isotone optimization (without the integer constraint) was considered in [18, 19]. Clearly, the set of feasible solutions to this problem, all monotone vectors y, is a convex cone. However, such an inference is not possible with the integer constraint. Nevertheless, strong results as stated above hold. For a further comparison of the two versions of the problem see section 3.

Finally, we note that our problem has the following integer programming formulation. Given x find y with integer y_j so as to minimize t subject to $w_j y_j + t \geq w_j x_j$, $1 \leq j \leq n$, $-w_j y_j + t \geq -w_j x_j$, $1 \leq j \leq n$, and $y_1 \leq y_2 \leq \cdots \leq y_n$.

2. The dual problem and weak duality.

The problem defined in section 1 is the primal problem. In this section, we develop a dual problem and obtain a weak duality result. We strengthen the latter to strong duality in section 3.

Let M be the set of all monotone integer vectors; i.e., all $z \in R^n$ such that $z_1 \leq z_2 \leq \cdots \leq z_n$ and each z_j is an integer. Our primal problem then may be restated as follows. Given $x \in R^n$ find $y \in M$ so that

$$\Delta := \min\{d(x,z) : z \in M\} = d(x,y).$$

The notations $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ are used to denote, respectively, the floor and ceiling functions of the reals. Similarly, $[\cdot]$ denotes the rounding function defined as follows. For any real c, we let $[c] = \lfloor c\rfloor$ if $\lfloor c\rfloor \leq c < \lfloor c\rfloor + 1/2$ and $[c] = \lceil c\rceil$ if $\lceil c\rceil - 1/2 < c \leq \lceil c\rceil$. For $c = \lfloor c\rfloor + 1/2 = \lceil c\rceil - 1/2$, we set $[c]$ equal to either $\lfloor c\rfloor$ or $\lceil c\rceil$ uniformly throughout the computations. Throughout this article, we use the square brackets $[\cdot]$ to denote the rounding function only.

We first obtain lower bounds on Δ. These give us motivation to define a dual problem and lead us naturally to a weak duality result. Indeed, let

$$\delta_1 = \max\{|x_j - [x_j]| : \quad 1 \leq j \leq n\}.$$

If $y \in M$, then, since y_j is an integer, we have $w_j|x_j - y_j| \geq w_j|x_j - [x_j]|$ for all j. This gives $d(x,y) \geq \delta_1$, and, thus, $\Delta \geq \delta_1$. Note that if $x_1 \leq x_2 \leq \cdots \leq x_n$, then we immediately obtain an optimal solution y of the problem by setting $y_j = [x_j]$ for all j, and in this case indeed $\Delta = \delta_1$. Thus, the lower bound δ_1 is attained by Δ. Now consider the complementary case $x_j > x_k$ for some $j < k$. Hopefully, we could obtain a different bound on Δ. To this end, let T and T_0 be the following sets of ordered pairs:

$$T = \{(j,k): \quad 1 \leq j \leq k \leq n\},$$
$$T_0 = \{(j,k) \in T: \quad j < k, \quad x_j > x_k\}.$$

Let also, $m_{jk} = (w_j x_j + w_k x_k)/(w_j + w_k)$, $(j,k) \in T$. Note that m_{jk} may be viewed as the average of x_j and x_k with weights $w_j/(w_j + w_k)$ and $w_k/(w_j + w_k)$, which add up to 1. Then, clearly, $m_{jj} = x_j$ and, by the property of averages, we observe (for use in section 3) that

$$\min\{x_j, x_k\} \leq m_{jk} \leq \max\{x_j, x_k\}. \tag{2.1}$$

By relaxing the constraints of the original problem, for each $(j,k) \in T_0$, we define the following subproblem of two variables. Find integer y_j and y_k so as to minimize

$$\max\{w_j|x_j - y_j|, \quad w_k|x_k - y_k|\}$$

subject to $y_j \leq y_k$. Because the subproblem is a relaxation of the original problem, the optimal objective value of the subproblem gives a lower bound on Δ for each $(j,k) \in T_0$. To solve the subproblem, then, let us first drop the integer restriction on y_j, y_k. Then, clearly, since $x_j > x_k$, the unique optimal solution (y_j, y_k) must satisfy $y_j = y_k$ and $w_j(x_j - y_j) = w_k(y_k - x_k)$. This gives $y_j = y_k = m_{jk}$ with the optimal objective value equal to $w_j(y_j - m_{jk}) = (w_j w_k)/(w_j + w_k)(x_j - x_k)$.

Now introduce the integer constraint on y_j, y_k. The subproblem becomes more complicated. Its optimal solution is not unique in general. Furthermore, it is not true that $y_j = y_k$ in its every solution. For example, suppose $w_j = w_k = 1$, $x_j = 5/2$, $x_k = 1/2$. Then $y_j = 1$, $y_k = 2$ is optimal. Note, however, that $y_j = y_k = 1$ or 2 are also optimal. Thus, we may surmise the following.

Lemma 2.1. *The subproblem always has at least one optimal solution* (y_j, y_k) *with* $y_j = y_k$.

Proof. Assume that $y_j < y_k$ at optimality. We produce another optimal solution (y'_j, y'_k) with $y'_j = y'_k$. If $y_j \geq x_k$, let $y'_j = y_j$ and $y'_k = y_j$. Then $y'_j = y'_k$. Since, $x_j - y'_j = x_j - y_j$ and $0 \leq y'_k - x_k < y_k - x_k$, the new solution (y'_j, y'_k) is at least as good as (y_j, y_k), and, hence, optimal. If $y_k \leq x_j$, then, by a symmetric argument, the lemma holds. Now suppose that $y_j < x_k$ and $y_k > x_j$. If $w_j \geq w_k$, then let $y'_j = y_j$ and $y'_k = y_j$. Then $y'_j = y'_k$ and $x_j - y'_j = x_j - y_j$. Since $x_j > x_k$, we have $x_j - y_j > x_k - y_j = x_k - y'_k \geq 0$. This with the inequality $w_j \geq w_k$ gives $w_j|x_j - y'_j| = w_j|x_j - y_j| \geq w_k|x_k - y'_k|$. Consequently,

$$\max\{w_j|x_j - y'_j|, w_k|x_k - y'_k|\} = w_j|x_j - y_j| \leq \max\{w_j|x_j - y_j|, w_k|x_k - y_k|\}.$$

Thus (y'_j, y'_k) is optimal. If $w_j < w_k$, a symmetric argument completes the proof. □

Having established that y_j, y_k have a common value in this special optimal solution, we ask if we could reasonably expect this value to equal $\lfloor m_{jk} \rfloor$ or $\lceil m_{jk} \rceil$. If so, what are the conditions which let us make the choice among the two? To answer these questions, define an integer c_{jk} for each $(j, k) \in T$ as follows.

$$\begin{aligned} c_{jk} &= \lfloor m_{jk} \rfloor, \text{if } w_j|m_{jk} - \lfloor m_{jk} \rfloor| \leq w_k|m_{jk} - \lceil m_{jk} \rceil|, \\ &= \lceil m_{jk} \rceil \text{ otherwise.} \end{aligned}$$

It is easy to verify that $c_{jj} = [m_{jj}] = [x_j]$. We show below that c_{jk} solves the subproblem.

Lemma 2.2. $y_j = y_k = c_{jk}$ *gives an optimal solution to the subproblem, and its minimum objective function value is given by*

$$\begin{aligned} \max\{w_j|x_j - c_{jk}|, w_k|x_k - c_{jk}|\} &= r_{jk} + w_j|m_{jk} - \lfloor m_{jk} \rfloor|, \\ &\qquad \textit{if } w_j|m_{jk} - \lfloor m_{jk} \rfloor| \leq w_k|m_{jk} - \lceil m_{jk} \rceil|, \\ &= r_{jk} + w_k|m_{jk} - \lceil m_{jk} \rceil| \textit{ otherwise}, \end{aligned} \tag{2.2}$$

where $r_{jk} = (w_j w_k)/(w_j + w_k)(x_j - x_k)$.

Proof. Using Lemma 2.1, we let $y_j = y_k = c$ in the objective function of the subproblem and find the value of c which minimizes the objective. To this end, we verify that $w_j(x_j - m_{jk}) = w_k(m_{jk} - x_k) = r_{jk}$, by substituting $(w_j x_j + w_k x_k)/(w_j + w_k)$ for m_{jk}. Since $x_j - c = (x_j - m_{jk}) + (m_{jk} - c)$, we obtain

$$w_j(x_j - c) = w_j(x_j - m_{jk}) + w_j(m_{jk} - c) = r_{jk} + w_j(m_{jk} - c).$$

Similarly, $w_k(c - x_k) = r_{jk} + w_k(c - m_{jk})$. Consequently, we have

$$\begin{aligned} \max\{w_j|x_j - c|, w_k|x_k - c|\} &= r_{jk} + w_j(m_{jk} - c), \quad c \leq m_{jk}, \\ &= r_{jk} + w_k(c - m_{jk}) \text{ otherwise.} \end{aligned} \tag{2.3}$$

Thus, for $c \leq m_{jk}$, (2.3) is minimized by $c = \lfloor m_{jk} \rfloor$ giving the minimum value $r_{jk} + w_j(m_{jk} - \lfloor m_{jk} \rfloor) = V_1$, say. Similarly, for $c > m_{jk}$, the minimizer of (2.3) is $c = \lceil m_{jk} \rceil$ with the minimum $r_{jk} + w_k(\lceil m_{jk} \rceil - m_{jk}) = V_2$, say. If $V_1 \leq V_2$, or, equivalently, $w_j|m_{jk} - \lfloor m_{jk} \rfloor| \leq w_k|m_{jk} - \lceil m_{jk} \rceil|$, then $\lfloor m_{jk} \rfloor$ is the minimizer of the objective of the subproblem, otherwise $\lceil m_{jk} \rceil$ is the minimizer. Now, the definition of c_{jk} shows that it minimizes the objective with the minimum value as in the lemma. □

As was observed before, the optimal objective value of the subproblem gives a lower bound on Δ for each $(j,k) \in T_0$. Hence, if we let

$$\delta_2 = \max\{\max\{w_j|x_j - c_{jk}|, w_k|x_k - c_{jk}|\}\colon (j,k) \in T_0\},$$

then $\Delta \ge \delta_2$, which is the new bound we wanted. (The use of the right side of (2.2) in the computation of δ_2 is discussed in section 4.) Now, if

$$\delta = \max\{\delta_1, \delta_2\},$$

then $\Delta \ge \delta$. This leads to our dual problem which is to determine δ. Note that δ depends only on the given vector x. Observe also that δ has two components: δ_1, which is formed by considering each variable y_j separately, and δ_2, which is obtained by considering variables pairwise. We state below the weak duality formally.

PROPOSITION 2.3 (weak duality) $\Delta \ge \delta$.

We obtain in Theorem 3.3 the strong duality $\Delta = \delta$. Thus the singular and pairwise values, δ_1 and δ_2, turn out to be adequate to generate the overall optimality, a rather remarkable occurrence.

3. Strong duality and characterization of optimal solutions. In this section, we establish a strong duality result and identify optimal solutions to the primal. We start with the following definitions:

$$\underline{y}_j = \lceil \max\{x_i - \delta/w_i\colon \quad i \le j\} \lceil, \qquad 1 \le j \le n, \tag{3.1}$$

$$\bar{y}_j = \lfloor \min\{x_k + \delta/w_k\colon \quad k \ge j\} \rfloor, \qquad 1 \le j \le n. \tag{3.2}$$

Clearly, $\underline{y}, \bar{y} \in M$. In Theorem 3.3 below, we establish the strong duality $\Delta = \delta$ and characterize the set of all optimal solutions to the problem as a "vector interval" $[\underline{y}, \bar{y}] \cap Z$ of R^n, where $\underline{y} \le \bar{y}$; $\underline{y}$ and $\bar{y}$ are, respectively, the minimal and maximal optimal solutions. We need the following preliminary results.

LEMMA 3.1. *For all $(j,k) \in T$, the following two conditions are equivalent:*

(a) $w_j(m_{jk} - \lfloor m_{jk} \rfloor) \le w_k(\lceil m_{jk} \rceil - m_{jk})$,

(b) $w_j(x_j - \lfloor m_{jk} \rfloor) \le w_k(\lceil m_{jk} \rceil - x_k)$.

Proof. It is easy to verify that (a) may be written as $(w_j + w_k)m_{jk} \le w_j\lfloor m_{jk} \rfloor + w_k\lceil m_{jk} \rceil$. Since $(w_j + w_k)m_{jk} = w_jx_j + w_kx_k$, the previous inequality is equivalent to $w_jx_j + w_kx_k \le w_j\lfloor m_{jk} \rfloor + w_k\lceil m_{jk} \rceil$, which may be written in the form of (b). □

LEMMA 3.2. *If $(j,k) \in T$, then $x_j - \delta/w_j \le c_{jk} \le x_k + \delta/w_k$.*

Proof. If $(j,k) \in T_0$, then, by the definition of δ_2, we have $w_j|x_j - c_{jk}| \le \delta_2 \le \delta$ and $w_k|x_k - c_{jk}| \le \delta_2 \le \delta$. This gives $w_j(x_j - c_{jk}) \le \delta$ and $w_k(c_{jk} - x_k) \le \delta$. Rearranging these two inequalities we obtain $x_j - \delta/w_j \le c_{jk} \le x_k + \delta/w_k$, which is the required result. Now suppose that $(j,k) \in T \backslash T_0$. Then, by the definition of T and T_0, we have $x_j \le x_k$, which gives $[x_j] \le [x_k]$. Also, by (2.1), $x_j \le m_{jk} \le x_k$. Consequently, $[x_j] \le [m_{jk}] \le [x_k]$. Again, by the definition of δ_1, we have $w_j|x_j - [x_j]| \le \delta_1 \le \delta$ and $w_k|x_k - [x_k]| \le \delta_1 \le \delta$. This gives $w_j(x_j - [x_j]) \le \delta$ and $w_k([x_k] - x_k) \le \delta$. Rearranging these inequalities we have

$$x_j - \delta/w_j \le [x_j] \le [x_k] \le x_k + \delta/w_k. \tag{3.3}$$

Suppose now that $w_j(m_{jk} - \lfloor m_{jk} \rfloor) \leq w_k(\lceil m_{jk} \rceil - m_{jk})$. In this case $c_{jk} = \lfloor m_{jk} \rfloor$, and condition (b) of Lemma 3.1 holds. If $w_j(x_j - \lfloor m_{jk} \rfloor) \leq 0$ in that condition, then $x_j \leq \lfloor m_{jk} \rfloor$. Consequently, $[x_j] \leq \lfloor m_{jk} \rfloor \leq [m_{jk}] \leq [x_k]$. By (3.3), the required result holds for $c_{jk} = \lfloor m_{jk} \rfloor$. Now suppose that $w_j(x_j - \lfloor m_{jk} \rfloor) > 0$ in condition (b) of Lemma 3.1. Then, by that condition, $w_k(\lceil m_{jk} \rceil - x_k) > 0$, which gives $\lfloor m_{jk} \rfloor < x_j \leq x_k < \lceil m_{jk} \rceil$. This implies that $\lceil m_{jk} \rceil - \lfloor m_{jk} \rfloor = 1$ and $\lfloor m_{jk} \rfloor \leq [x_j] \leq [x_k] \leq \lceil m_{jk} \rceil$. Clearly, either $[x_j] = \lfloor m_{jk} \rfloor$ or $[x_j] = \lceil m_{jk} \rceil$. In the former case, by (3.3), the required result holds for $c_{jk} = \lfloor m_{jk} \rfloor$. In the latter case, $[x_j] = [x_k] = \lceil m_{jk} \rceil$. Now condition (b) of Lemma 3.1 with $\lceil m_{jk} \rceil = [x_k]$ gives $w_j(x_j - \lfloor m_{jk} \rfloor) \leq w_k([x_k] - x_k)$. By the definition of δ_1, we have $w_k([x_k] - x_k) \leq \delta_1 \leq \delta$. Consequently, $w_j(x_j - \lfloor m_{jk} \rfloor) \leq \delta$. This gives $x_j - \delta/w_j \leq \lfloor m_{jk} \rfloor \leq \lceil m_{jk} \rceil = [x_k]$. Then the required result follows again for $c_{jk} = \lfloor m_{jk} \rfloor$ by (3.1). The case $w_j(m_{jk} - \lfloor m_{jk} \rfloor) > w_k(\lceil m_{jk} \rceil - m_{jk})$ may be established similarly. □

THEOREM 3.3. (a) (*Strong duality*) $\Delta = \delta$.

(b) (*Characterization of optimal solutions*). *Both $\underline{y}$ and $\bar{y}$ are optimal solutions with $\underline{y} \leq \bar{y}$; thus $d(x, \underline{y}) = d(x, \bar{y}) = \delta$. Furthermore, $y \in M$ is an optimal solution if and only if $\underline{y} \leq y \leq \bar{y}$.*

Proof. We prove both parts simultaneously. Clearly, $\underline{y}, \bar{y} \in M$. We now show that $\underline{y} \leq \bar{y}$. Let an index j be fixed and $i \leq j \leq k$. Then, by Lemma 3.2, there exists an integer c such that $x_i - \delta/w_i \leq c \leq x_k + \delta/w_k$. Consequently, $\lceil x_i - \delta/w_i \rceil \leq \lfloor x_k + \delta/w_k \rfloor$ for all i and k with $i \leq j \leq k$. It follows at once from (3.1) and (3.2) that $\underline{y}_j \leq \bar{y}_j$; i.e., $\underline{y} \leq \bar{y}$. Again, by (3.1) and (3.2), we have $x_j - \delta/w_j \leq \underline{y}_j \leq \bar{y}_j \leq x_j + \delta/w_j$, $1 \leq j \leq n$. Hence, $-\delta \leq w_j(\underline{y}_j - x_j) \leq w_j(\bar{y}_j - x_j) \leq \delta$, which gives $d(x, \underline{y}) = d(x, \bar{y}) \leq \delta$. Since $\underline{y}$ and $\bar{y} \in M$, we have shown that $\Delta = \delta$, and $\underline{y}$ and $\bar{y}$ are optimal solutions.

Now we characterize an optimal solution. Suppose $y \in M$ is optimal. Fix an index j. Then for all $i \leq j$, we have $y_i \leq y_j$. But $w_i|x_i - y_i| \leq \delta$, which gives $x_i - \delta/w_i \leq y_i$. Thus $x_i - \delta/w_i \leq y_j$ for all $i \leq j$, which implies $\underline{y}_j \leq y_j$. Similarly, $y_j \leq \bar{y}_j$. Hence $\underline{y} \leq y_j \leq \bar{y}_j$. Conversely, suppose that $y \in M$ and $\underline{y} \leq y \leq \bar{y}$. Since $\underline{y}_j \leq y_j \leq \bar{y}_j$, it is easy to verify that $|x_j - y_j| \leq \max\{|x_j - \underline{y}_j|, |x_j - \bar{y}_j|\}$. This at once gives $d(x, y) \leq \max\{d(x, \underline{y}), d(x, \bar{y})\}$. Now, both $\underline{y}$ and $\bar{y}$ are optimal; hence, $d(x, \underline{y}) = d(x, \bar{y}) = \delta$. It follows that $d(x, y) \leq \delta$, which shows the optimality of y. □

We now make several remarks. The results of the above theorem are similar to those for the problem without the integer constraint [18, 19]; however, the definitions of quantities are much different and the proofs are more involved. Special duality results exists for general approximation problems such as the one we are considering but without the integer constraint [6, 7, 20, 22]. However, when the integer constraint is introduced, the problem assumes a different structure. A comprehensive treatment of the duality for such problems does not exist at this time. A number of approximation problems are known to have extremal (i.e., minimal and/or maximal) optimal solutions [11, 12, 21, 22]. The above theorem shows that this is true for our problem. A general theory of extremal solutions for such problems with integer constraints does not currently exist.

Both the problems of isotonic regression and isotonic medium regression can be solved by the well-known pool adjacent violators (PAV) algorithms. It is shown in [17] that the PAV algorithms can be applied to a broad class of problems including the two above. However, the PAV algorithms cannot be applied to our problem because it has a different structure than isotonic regression.

Let Δ' and M' be quantities for the isotone optimization problem without the integer constraint corresponding, respectively, to Δ and M of our problem with the integer restriction. If $d(x,u) = \Delta'$ for some $u \in M'$ then, clearly, $[u] := ([u_1], [u_2], \dots, [u_n]) \in M$. This observation shows that $\Delta \le \Delta' + \max\{w_j : 1 \le j \le n\}/2$. On the other hand, Δ/Δ' can be arbitrarily large. For example, let $n = 2$, $w_j = 1$ for all j, and $x = (1/2 + 2\varepsilon, 1/2)$, where $\varepsilon > 0$. It is easy to see that $y = (1,1)$ and $u = (1/2 + \varepsilon, 1/2 + \varepsilon)$, respectively, are solutions to the two problems with and without the integer constraint. Consequently, $\Delta = 1/2$ and $\Delta' = \varepsilon$ giving $\Delta/\Delta' = 1/(2\varepsilon)$.

The results of this section can be easily extended to a problem on a set $S = \{s_1, s_2, \dots, s_n\}$ with partial order $\le$. For any function x on S, let $x_j = x(s_j)$. A function y on S is called isotone [18, 19] if $y_j \le y_k$ whenever $s_j \le s_k$. Given x, the problem is to find an isotone y which minimizes $d(x,y)$. All the previous formulae and results hold for this problem when $\le$ is interpreted as the partial order and $<$ is interpreted as $\le$ but $\ne$.

4. Minimax forms of optimal solutions and algorithms. In this section we derive the min–max and max–min forms of optimal solutions and obtain algorithms for computing the solutions. It is known that the (unique) optimum of the isotonic regression problem can be expressed in max–min and min–max forms [16]. Van Eeden [8] showed that if the objective function of the isotonic problem is separable and each component function is strictly unimodal, then such representations for the optimal solution can be obtained. Clearly, $d_p(x,y)$, $1 < p < \infty$, is a special case of such a separable function, and in particular so is $d_2(x,y)$, the isotonic regression objective. Note that $d(x,y)$ is not separable or unimodal. It was shown in [19] that, in spite of this, one solution of the isotonic optimization problem (without the integer constraint) has such a representation and certain special properties. The max–min and min–max representations also hold for our integer restricted problem as shown below, and they lead us to a linear time algorithm for computation of an optimal solution when $w_j = 1$.

For each $(j,k) \in T$, we defined integers c_{jk} in section 2. We define two vectors $\underline{z}$ and $\bar{z}$ by

$$\underline{z}_j = \max_{i \le j} \min_{k \ge j} c_{ik}, \qquad 1 \le j \le n,$$
$$\bar{z}_j = \min_{k \ge j} \max_{i \le j} c_{ik}, \qquad 1 \le j \le n.$$

Theorem 4.1. *Both $\underline{z}$ and $\bar{z}$ are optimal solutions with $\underline{z} \le \bar{z}$. If $w_j = 1$ for all j, then $\underline{z}_j = \bar{z}_j = [(\max_{i \le j} x_i + \min_{i \ge j} x_i)/2]$.*

Proof. By an elementary result in minimax theory, we have max min $\le$ min max [14]. This gives $\underline{z}_j \le \bar{z}_j$. By Lemma 3.2, we have $x_i - \delta/w_i \le c_{ik} \le x_k + \delta/w_{kj}$ for all $i \le k$. Taking the relevant minima and maxima, we at once obtain by (3.1) and (3.2) that $\underline{y}_j \le \underline{z}_j \le \bar{z}_j \le \bar{y}_j$. Now optimality of $\underline{z}$ and $\bar{z}$ follows from part (b) of Theorem 3.3. If $w_j = 1$ for all j, we have $c_{ik} = [m_{ik}] = [(x_i + x_k)/2]$ for all $(i,k) \in T$ as may be easily verified. The last statement of the theorem follows from this observation. □

We note that if $w_j = 1$ for all j, then u defined by $u_j = (\max_{i \le j} x_i + \min_{i \ge j} x_i)/2$ is an optimal solution of the isotone optimization problem [19]. Since $\underline{z}_j = \bar{z}_j = [u_j]$ when $w_j = 1$, we see that the rounding of a solution of the isotone problem gives a solution of the problem with the integer constraint. As was observed earlier, a similar result holds for the integer isotonic regression problem [9].

The algorithms for computing the solutions $\underline{y}$, $\bar{y}$ and $\underline{z}$, $\bar{z}$ may be based directly on their representations. Clearly, the sets T and T_0, at worst, contain $O(n^2)$ elements.

First suppose that $w_j > 0$ are arbitrary. It is easy to see that the computations of c_{jk}, $(j,k) \in T$, and of δ_2 and, hence, δ require $O(n^2)$ time. It is easy to verify that δ_2 can be determined with less computations using the right side of (2.2) than its left side once $w_j(m_{jk} - \lfloor m_{jk} \rfloor)$ and $w_k(\lceil m_{jk} \rceil - m_{jk})$ are obtained (compute r_{jk} using $r_{jk} = w_j(x_j - m_{jk})$). We may then compute $\underline{y}$ and $\bar{y}$ in $O(n)$ time using their obvious recursive definitions. For example, $\underline{y}_1 = x_1 - \delta/w_1$, $\underline{y}_{j+1} = \max\{\underline{y}_j, x_j - \delta/w_j\}$, $1 \le j \le n-1$. Hence, the overall worst-case time complexity [1] of the algorithm for computing $\underline{y}$, $\bar{y}$ is $O(n^2)$. Again, clearly, the computation of $\underline{z}$ and $\bar{z}$ is of $O(n^2)$ worst-case time complexity. Now suppose that $w_j = 1$ for all j. Then $\underline{y}$ and $\bar{y}$ still take $O(n^2)$ overall time, but we may compute $\underline{z}$ and $\bar{z}$ in $O(n)$ time using, again, their obvious recursive definitions. Thus, the assumption $w_j = 1$ only reduces the complexity of computation of $\underline{z}$ and $\bar{z}$ but not of $\underline{y}$ and $\bar{y}$.

REFERENCES

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison–Wesley, Reading, MA, 1974.

[2] T. S. Arthanari and Y. Dodge, *Mathematical Programming in Statistics*, John Wiley and Sons, New York, 1981.

[3] M. J. Best and N. Chakravarti, *Active set algorithms for isotonic regression; a unifying framework*, Math. Programming, 47 (1990), pp. 425–439.

[4] D. Birkes and Y. Dodge, *Alternative Methods of Regression*, John Wiley and Sons, New York, 1993.

[5] N. Chakravarti, *Isotonic median regression; a linear programming approach*, Math. Oper. Res., 14 (1989), pp. 303–308.

[6] F. R. Deutsch, *Some Applications of Functional Analysis to Approximation Theory*, Doctoral Dissertation, Division of Applied Mathematics, Brown University, Providence, RI, 1965.

[7] F. R. Deutsch and F. H. Maserick, *Applications of the Hahn–Banach theorem in approximation theory*, SIAM Rev., 9 (1967), pp. 516–530.

[8] C. Van Eeden, *Testing and Estimating Ordered Parameters of Probability Distributions*, Doctoral Dissertation, Studentendrukkerij Poortpers, Amsterdam, 1958.

[9] A. J. Goldstein and J. B. Kruskal, *Least-squares fitting by monotonic functions having integer values*, J. Amer. Statist. Assoc., 71 (1976), pp. 370–373.

[10] R. Gonin and A. H. Money, *Nonlinear L_p-norm Estimation*, Marcel Dekker, New York, 1989.

[11] D. Landers and L. Rogge, *Best approximants in L_Φ-spaces*, Z. Wahrsch. Verw. Gebiete, 51 (1980), pp. 215–237.

[12] D. Landers and L. Rogge, *Natural choice of L_1-approximants*, J. Approx. Theory, 33 (1981), pp. 268–280.

[13] D. A. Ratkowsky, *Handbook of Nonlinear Regression Models*, Marcel Dekker, New York, 1990.

[14] J. Ponstein, *Approaches to the Theory of Optimization*, Cambridge University Press, Cambridge, United Kingdom, 1980.

[15] T. Robertson and F. T. Wright, *Algorithms in order restricted statistical inference and the Cauchy mean value property*, Ann. Statist., 8 (1980), pp. 645–651.

[16] T. Robertson, F. T. Wright, and R. L. Dykstra, *Order Restricted Statistical Inference*, John Wiley & Sons, New York, 1988.

[17] U. Stromberg, *An algorithm for isotonic regression with arbitrary convex distance function*, Comput. Statist. Data Anal., 11 (1991), pp. 205–219.

[18] V. A. Ubhaya, *Isotone optimization*, I, J. Approx. Theory, 12 (1974), pp. 146–159.

[19] V. A. Ubhaya, *Isotone optimization*, II, J. Approx. Theory, 12 (1974), pp. 315–331.

[20] V. A. Ubhaya, *Duality in approximation and conjugate cones in normed linear spaces*, J. Math. Anal. Appl., 58 (1977), pp. 419–436.

[21] V. A. Ubhaya, *Lipschitzian selections in best approximation by continuous functions*, J. Approx. Theory, 61 (1990), pp. 40–52.

[22] V. A. Ubhaya, *Duality and Lipschitzian selections in best approximation from nonconvex cones*, J. Approx. Theory, 64 (1991), pp. 315–342.